Министерство образования и науки Республики Казахстан

Карагандинский государственный университет имени академика

Е.А. Букетова

Факультет Математики и информационных технологий

Кафедра Прикладной математики и информатики

**Спирина Елена Александровна**

**Курс лекций**

**по дисциплине «Проектирование интернет-приложений»**

образовательная программа «6В06103 – Информационные системы»

Караганда 2020

**Лекция 1. Интернет-технологии**

План:

1 Основные понятия Интернета.

2 Эволюция

3 Структура Интернета

4 Провайдеры

5 Адресация в Интернете

6. Система доменных имен Интернета (DNS)

7. Протокол TCP / IP

8. Универсальный локатор ресурсов (URL)

9. Способы подключения к интернету

10. Интернет-Услуги

11. World Wide Web

# Интернет

Интернет — это объединение взаимосвязанных сетей в мировом масштабе.

Интернет - всемирная система интегрированных компьютерных сетей для хранения и передачи информации.

Интернет можно рассматривать как «сеть сетей», каждая из которых управляется независимым оператором - провайдером услуг Интернета. Серверы интернета поддерживают работу сети интернет и пользователей. К сети интернет могут подключаться локальные и корпоративные сети, мобильные устройства и компьютеры. Интернет не принадлежит какому-либо лицу или группе людей.

*Вопросами регулирования структуры и стандартизации протоколов и процессов Интернета занимаются специальные организации.*

Эти организации включают в себя Инженерную группу по развитию Интернета (Internet Engineering Task Force, IETF), Корпорацию Интернета по распределению адресов и номеров (International Corporation for Assigned Names and Numbers, ICANN) и Совет по архитектуре сети Интернет (Internet Architecture Board, IAB), а также многие другие.

Интернет - это единая сеть способна передавать информацию из любой точки мира в любую другую точку.

Интернет определяется как информационная супермагистраль чтобы получить доступ к информации. С точки зрения пользователей, Интернет представляет собой набор информационных ресурсов, распределенных по различным сетям, включая сети интернет-провайдеров, корпоративные сети, сети и отдельные компьютеры домашних пользователей.

Каждый отдельный компьютер в данной сети называется **хостом**

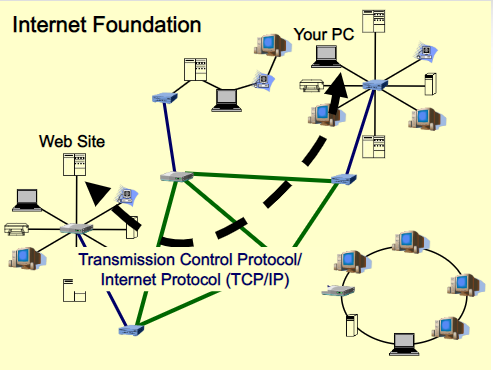
**Сеть интернет имеет следующие особенности:**

• Каждый компьютер в Интернете идентифицируется уникальным IP-адресом

• Интернет использует костюмные протоколы - TCP / IP

• Система доменных имен (DNS) используется для пользователей, чтобы найти определенный компьютер

- Интернет доступен каждому пользователю во всем мире

****

**Эволюция**

Концепция Интернет возникла в 1969 году и претерпел ряд технологических и Инфраструктурных изменений, как описано ниже:

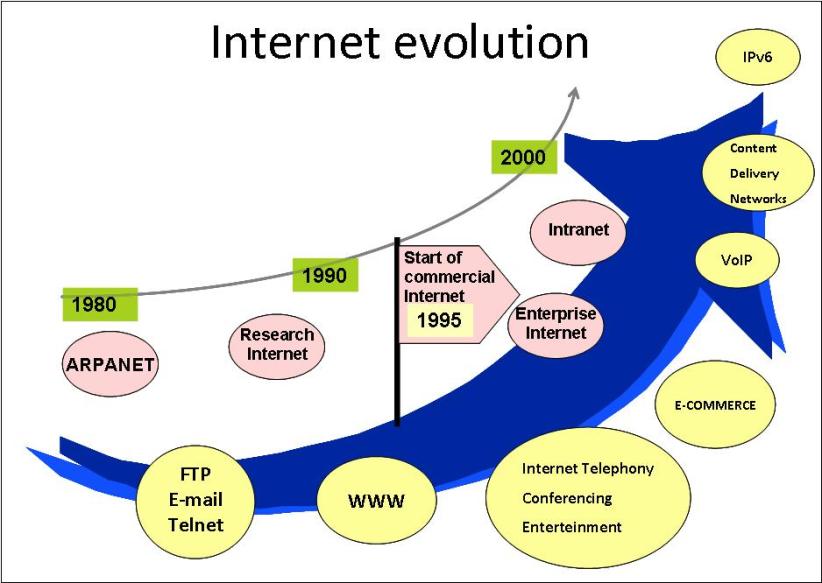
• Происхождение Интернета разработаны от концепции Advanced ResearchProjectAgencyNetwork (ARPANET).

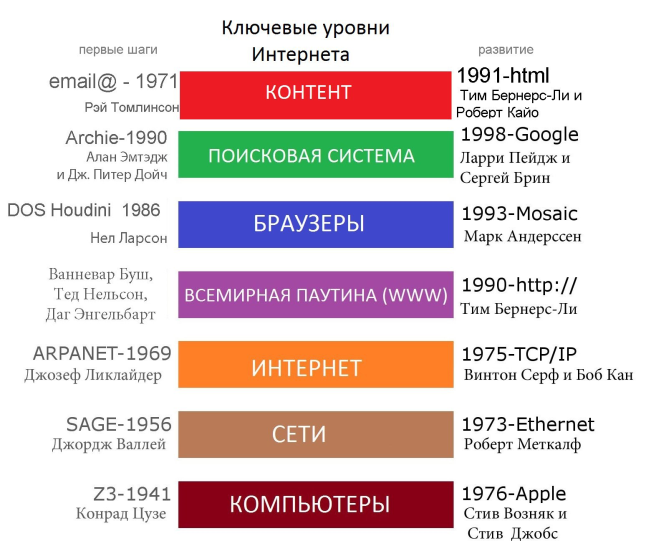
• ARPANET была разработана Департаментом обороны Соединенных Штатов.

• Основная цель ARPANET заключалась в обеспечении связи между различными органами власти.

• Первоначально было только четыре узла, формально называемые **Хосты**.

• с изобретением новых технологий, таких как TCP / IP протоколы, DNS, WWW, браузеров, языков сценариев и т.д., Интернет предоставлен носитель для публикации и доступа к информации через Интернет.



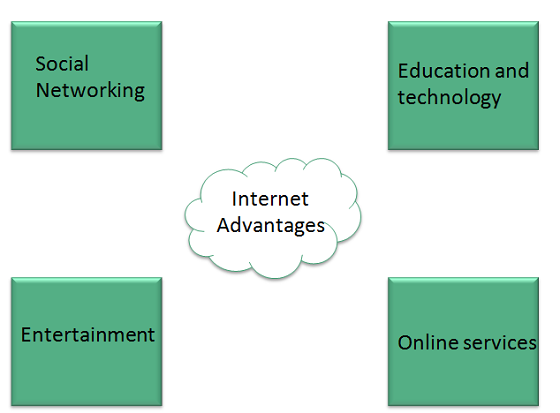


Как вы видите на слайде, в 1975 году был разработан протокол TCP / IP.

- В 1990 году был разработан WWW

**Преимущества**

Интернет охватывает практически все аспекты жизни. Здесь мы рассмотрим некоторые из преимуществ Интернета:



• Интернет позволяет нам общаться с людьми, сидя в удаленных местах. Существуют различные приложения, доступные на ср, который использует Интернет в качестве среды для общения. Можно найти различные сайты социальных сетей, таких как:

 Facebook

 Twitter

 Yahoo

 Google+

 Flickr

• Можно путешествовать для любого вида информации через Интернет. Информация о различных такие темы, как технологии, здоровье и науки, социальных исследований, географической информации, информационных технологий, продуктов и т.д. может быть перемещался с помощью поисковой системы.

• Помимо коммуникации и источника информации, Интернет также служит средой для развлечений. Ниже приведены различные режимы для развлечений в сети Интернет.

 Интернет Телевидение

 онлайн игры

 Песни

 Видео

 социальные приложения сети

Интернет позволяет использовать множество услуг, таких как:

 Интернет-банкинг

 Интернет-магазин

 Интернет Заказ билетов

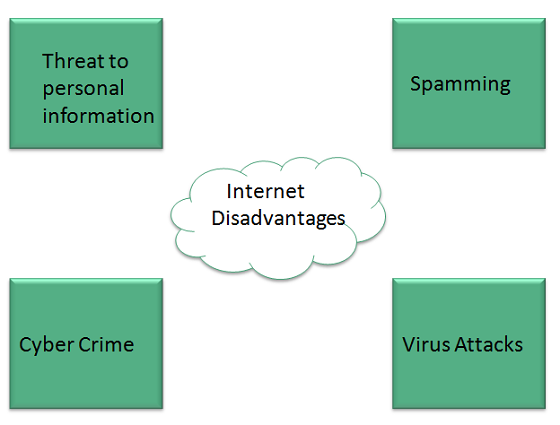
 Интернет Оплата счетов

 Обмен данными

 Электронная почта

**Недостатки**

Тем не менее, Интернет стал мощным источником информации практически во всех областях, но существует много недостатков:



1. **Угрозы**

Есть всегда шансы потерять личную информацию, такую как имя, адрес, номер кредитной карты.

**2) Spamming**

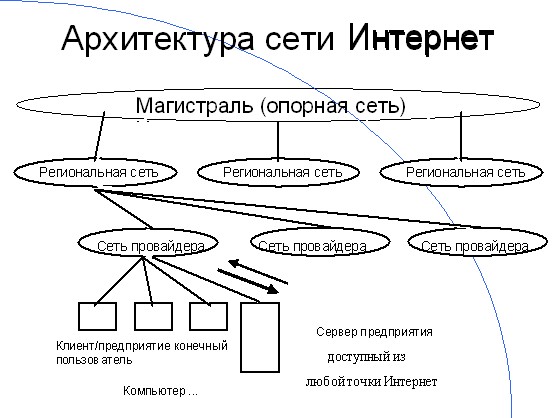
Другим недостатком является Spamming. Spamming соответствует нежелательных сообщений электронной почты навалом.

**3)Вирус** легко может распространиться на компьютерах, подключенных к Интернету. Такие вирусные атаки могут привести вашу систему к краху или ваши важные данные могут быть удалены.

**4) Cybercrime**

Также самая большая угроза в Интернете является порнография.

## Существуют различные сайты, которые не обеспечивают проверку подлинности информации. Это приводит к заблуждению среди многих людей.



Internet представляет собой Всемирную сеть, информация в которой хранится на серверах.

Серверы имеют свои адреса и управляются специализированными программами. Они позволяет пересылать почту и файлы, производить поиск в базах данных и т. п. бмен информацией между серверами сети выполняется по высокоскоростным каналам связи.

Доступ отдельных пользователей к информационным ресурсам Internet обычно осуществляется по телефонной сети через провайдера или корпоративную сеть. В качестве **провайдера** выступает некоторая организация, имеющая модемный пул для соединения с клиентами и выхода во Всемирную сеть.

**Опорная сеть Интернета** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Internet backbone*) — главные магистрали передачи данных между огромными, стратегически взаимосвязанными [сетями](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) и [основными маршрутизаторами](https://ru.wikipedia.org/w/index.php?title=Core_router&action=edit&redlink=1) в [Интернете](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82).

Эти магистрали передачи данных контролируются коммерческими, государственными, научными и другими высокопроизводительными центрами, [точками обмена трафиком](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D1%87%D0%BA%D0%B0_%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%D0%B0_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%BE%D0%BC) и [точками доступа к сети](https://ru.wikipedia.org/w/index.php?title=Network_access_point&action=edit&redlink=1), которые обмениваются интернет-трафиком между странами и континентами.

[Интернет-провайдеры](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BF%D1%80%D0%BE%D0%B2%D0%B0%D0%B9%D0%B4%D0%B5%D1%80) (часто [Tier-1-операторы](https://ru.wikipedia.org/wiki/Tier-1-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D1%8B)) участвуют в обмене трафиком опорной сети Интернета с помощью частным образом заключённых соглашений о соединениях сетей, главным образом по принципу [пиринга](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%80%D0%B8%D0%BD%D0%B3) (соглашение интернет-операторов об обмене трафиком между своими сетями, а также техническое взаимодействие).

**В качестве высокоскоростной магистрали передачи данных** используются выделенные телефонные линии, оптоволоконные и спутниковые каналы связи.

Любая организация для подключения к Internet использует специальный компьютер, который называется шлюзом (gateway). На нем устанавливается программное обеспечение, осуществляющее обработку всех сообщений, проходящих через шлюз. Каждый шлюз имеет свой IP-адрес. Если поступает сообщение, адресованное локальной сети, к которой подключен шлюз, то оно передается в эту локальную сеть. Если сообщение предназначено для другой сети, то оно передается следующему шлюзу

Каждый шлюз имеет информацию обо всех остальных шлюзах и сетях. Когда сообщение посылается из локальной сети через шлюз в Internet, то при этом выбирается самый «быстрый» путь. Шлюзы обмениваются друг с другом информацией о маршрутизации и состоянии сети, используя специальный шлюзовый протокол.

**Провайдер**

Некоторые компании могут выступать в качестве провайдера. Провайдер имеет свой шлюз в Internet и позволяет другим компаниям и отдельным пользователям подключаться к Сети через этот шлюз. Кроме информации о маршрутизации сообщений, шлюзу необходимы данные о параметрах подсетей, подключенных к более крупной сети, для корректировки маршрутов передачи сообщений в случае сбоев в отдельных частях сети.

**Шлюзы бывают двух типов: внутренние и внешние.** Внутренними называют шлюзы, расположенные в небольшой подсети и обеспечивающие связь с более крупной корпоративной сетью.

Такие шлюзы поддерживают связь между собой с помощью внутреннего шлюзового протокола IGP (Internal Gateway Protocol).

Внешние шлюзы применяются в больших сетях, подобных Internet, настройки их постоянно меняются из-за изменений в мелких подсетях.

Связь между внешними шлюзами осуществляется через внешний шлюзовый протокол EGP (Exterior Gateway Protocol).

Существуют различные типы интернет-провайдеров:

Просто поставщик Интернет-услуг выполняет функцию транспорта для конечных пользователей - передает свой трафик в сети других поставщиков Интернет-услуг;

Интернет-провайдер имеет свои информационные и справочные ресурсы, предоставляя свой контент в виде веб-сайтов;

Хостинг-провайдер предоставляет свои помещения, каналы связи и серверы для размещения внешнего контента;

Провайдер - поставщик услуг доставки контента доставляет контент только нескольким точкам доступа, чтобы повысить скорость доступа пользователей к информации;

Провайдер поставщик услуг поддержки приложений предоставляет клиентам доступ к крупным универсальным программным продуктам, таким как SAP R3;

Провайдер поставщик биллинговых услуг обеспечивает оплату счетов в Интернете.

**Адресация в Интернет**

Три типа адресов используются для идентификации сетевых интерфейсов:

 аппаратные адреса (или MAC-адреса); MAC адрес или физический адрес используется для уникальной идентификации устройств в локальной сети. Он записывается на заводе-производителе в постоянную (энергонезависимую) память устройства, например сетевой карты или маршрутизатора.

 сетевые адреса (IP-адреса);

 символьные (доменные) имена.

**IP-адреса**

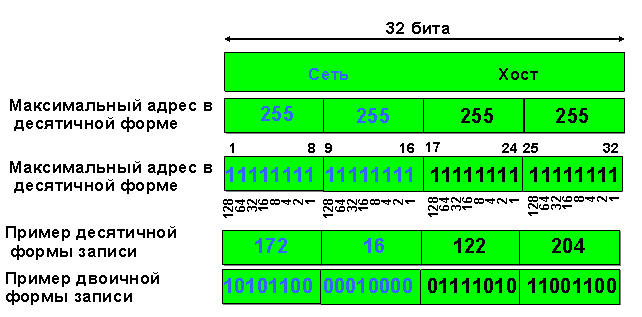
Система ІР-адресации учитывает структуру Интернет, т. е. то что Интернет является сетью сетей, а не объединением отдельных компьютеров.

ІР–адрес состоит из двух частей, одна из которых является адресом сети, а другая – адресом компьютера в данной сети.

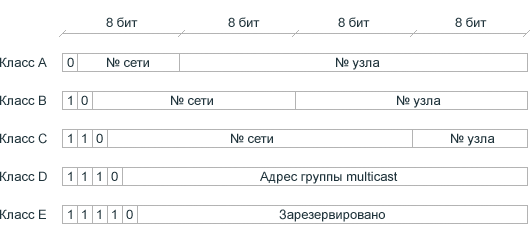
Адреса разделяются на три класса – А, В, С в зависимости от количества сетей данного класса и количества компьютеров в сети. Первые биты адреса отводятся для идентификации класса, а остальные разделяются на адрес сети и адрес компьютера.

В четвёртой версии Internet-протокола IP-адрес состоял из четырёх цифр, разделённых точками:

123.123.34.56



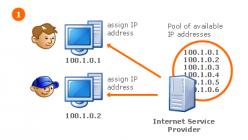


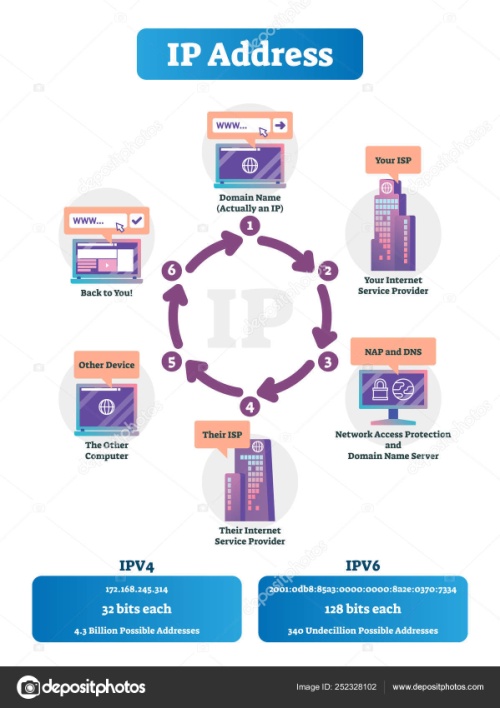


В шестой версии Internet-протокола IP-адрес состоит из 32 шестнадцатеричных цифр, разделённых двоеточиями:

2091:0db8:0000:0000:0d00:ff50:0092:8378







То есть конечный узел может входить в несколько IP-сетей

**Доменное имя** — это символьное обозначение, зарегистрированное для сетевой адресации, в которой используется система доменных имён (DNS).

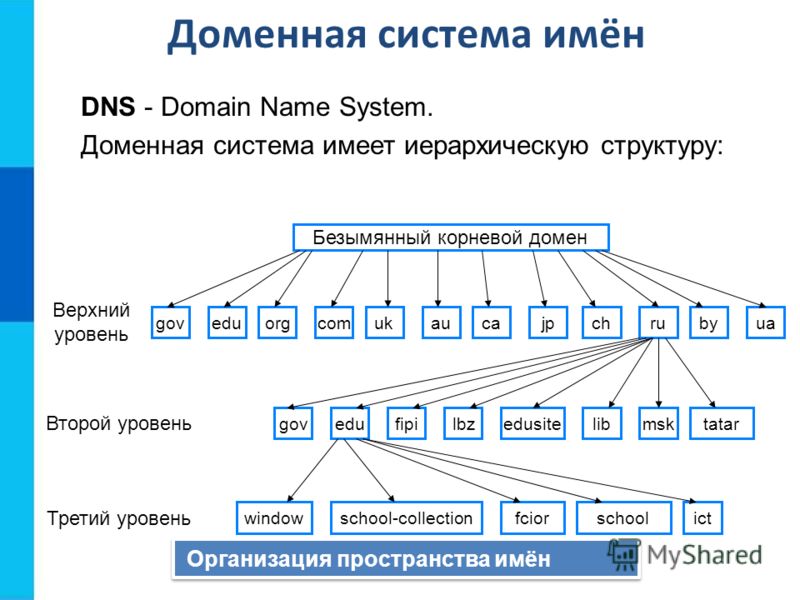
**Доме́нное имя** — символьное имя, служащее для идентификации областей, которые являются единицами административной автономии в сети Интернет, в составе вышестоящей по иерархии такой области. Каждая из таких областей называется доме́ном

**6. Система доменных имен в Интернете (DNS)**

Система доменных имен (DNS) представляет собой иерархическую децентрализованную систему присвоения имен компьютерам, подключенным к Интернету.

Основой DNS является представление об иерархической структуре доменного имени и зонах.

Система DNS (***Domain Name System)  - обеспечивает соответствие между IP-адресами компьютеров и символьными доменными именами***

****

**Domain Name System Architecture**

**Доменные имена**

DomainName является символической строки, связанной с IP-адресом.

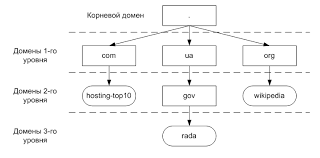
Доменные имена имеют **уровневую структуру**

**Система доменных имен** включает доменных имен, пространства доменных имен, имя сервера.

Домены верхнего уровня могут быть 2 типов:

- общие (административные)

- географические



Есть несколько доменных имен, доступных; некоторые из них носят общий характер, такие, как ком, Edu, гов, сеть и т.д., в то время как некоторые доменные имена на уровне страны, такие как KZ, RU, AU, в З.А., нас и т.д.

В следующей таблице приведены имена Административного домена верхнего уровня:

Значение доменных имен

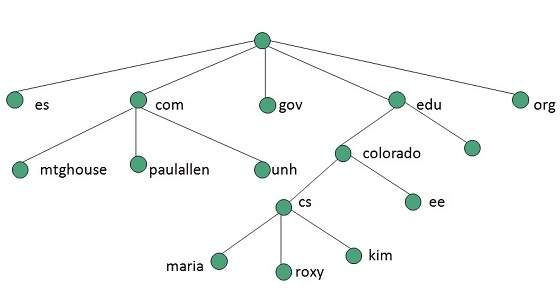
|  |  |
| --- | --- |
| **Domain Name** | **Meaning** |
| Com | Commercial business |
| Edu | Education |
| Gov | U.S. government agency |
| Int | International entity |
| Mil | U.S. military |
| Net | Networking organization |
| Org | Non profit organization |

В следующей таблице приведены Географические имена доменов верхнего уровня Страна:

|  |  |
| --- | --- |
| **Domain Name** | **Meaning** |
| kz | Kazakhstan |
| ru | Russia |
| in | India |
| fr | France |
| us | United States |
| uk | United Kingdom |
| jp | Japan |
| es | Spain |
| de | Germany |
| ca | Canada |
| hk | Hong Kong |
| etc |  |

**Пространства доменных имен**

Доменное имя пространство означает иерархию в структуре интернет-имен. Эта иерархия имеет несколько уровней (от 0 до 127), с корнем наверху. На следующей диаграмме показана иерархия пространства доменных имен:



В приведенной диаграмме каждое поддерево представляет собой домен. Каждый домен может быть разделена на субдомены, и они могут быть дополнительно разделены и так далее.

**Имя сервера**

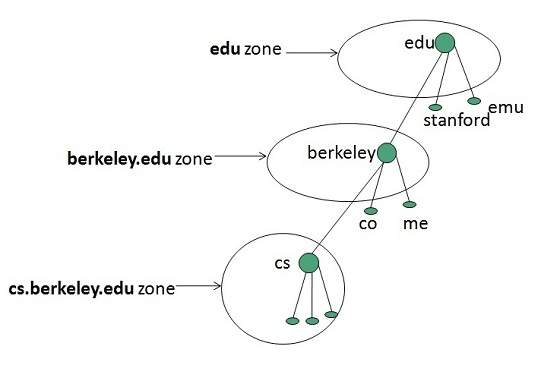
Сервер имен содержит базу данных DNS. Эта база данных включает в себя различные имена и соответствующие им IP-адреса. Так как это не возможно для одного сервера, чтобы сохранить всю базу данных DNS, поэтому, информация распространяется среди многих DNS-серверов.

• Иерархия сервера такой же, как иерархии имен.

• Полное имя пространство разделено на зоны

**зоны**

Зона является сбор узлов (поддоменов) под основной домен. Сервер поддерживает базу данных с именем файла зоны для каждой зоны.



Если домен далее не разделен на поддомены, то домен и зона относятся к одному и тому же предмету.

Однако информация об узлах в поддомене хранится на серверах нижних уровней; исходный сервер сохраняет ссылки на эти нижние уровни серверов.

**7. Протокол ТСР/ІР**

Сеть интернет, являющаяся сетью сетей и объединяющая громадное количество различных локальных, региональных и корпоративных сетей, функционирует и развивается благодаря использованию единого протокола передачи данных ТСР/ІР.

ТСР/ІР включает два протокола:

• Transmission Control Protocol (TCP) – транспортный протокол;

• Internet Protocol (IP) – протокол маршрутизации.

Протокол ТСР/ІР обеспечивает передачу информации между компьютерами сети

Internet Protocol (ІР) - протокол маршрутизации обеспечивает маршрутизацию ІР-пакетов, т.е. доставку информации от компьютера-отправителя к компьютеру-получателю.

Transmission Control Protocol (TCP), т.е. транспортный протокол обеспечивает разбиение файлов на ІР-пакеты в процессе передачи и сборку файлов в процессе получения.

**8. Uniform Resource Locator (URL)**

Для обозначения адресов ресурсов Интернета применяется **Uniform Resource Identifier** (**URI**) и Uniform Resource Locator URL

Сейчас URL позиционируется как часть более общей системы идентификации ресурсов URI, сам термин URL постепенно уступает место более широкому термину URI. Стандарт URL регулируется организацией IETF и её подразделениями.

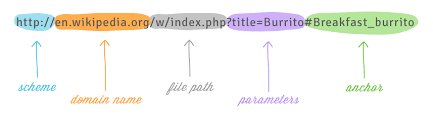
Унифицированный идентификатор ресурса (URI) - это строка символов, используемых для идентификации ресурса.

URL-адрес *подразумевает* способ доступа к указанному ресурсу и обозначается протоколом или механизмом доступа, что неверно для каждого URI.

Таким образом, http://www.example.com - это URL-адрес, а www.example.com - **URI**.

 Большинство веб-браузеров отображают URL-адрес веб-страницы над страницей в адресной строке.

Типичный URL-адрес может иметь форму http://www.example.com/index.html, которая указывает протокол (схему) (http), имя узла (доменное имя) (www.example.com) и имя файла (Index.html).



Например, www.tutorialspoint.com/internet\_technology/ index.html является URL на index.html, который хранится на веб-сервере tutorialspoint под директории internet\_technology.

Например*, http: // www.tutorialspoint.com / internet\_technology /index.htm*. где:

• HTTP является протоколом (схема).

• tutorialspoint.com это имя сервера.

• index.htm это имя файла.

Часть протокола сообщает веб-браузеру, как обрабатывать файл. Точно так же у нас есть некоторые другие протоколы, которые могут также быть использованы для создания URL являются:

* FTP
* https
* Gopher
* mailto
* news

Такие локаторы URL являются комбинацией URI и системы DNS.

Имя домена (или IP-адрес) является частью URL-адреса, указывающего компьютер (его сетевой интерфейс), на котором работает программа веб-сервера.

На клиентском компьютере для просмотра информации, получаемой с веб-сервера, используется специальная программа - веб-браузер. Основной функцией веб-браузера является отображение гипертекстовых страниц (веб-страниц). Для создания гипертекстовых страниц в Интернете изначально использовался HTML. Многие веб-страницы образуют веб-сайт.

**9. Способы подключения к Интернет**

**Подключение к Интернет для дома и небольшого офиса**

На рисунке показаны стандартные варианты подключения малых и домашних офисов.



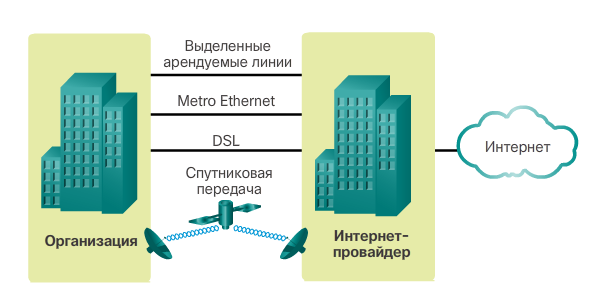
* **Кабельное подключение**— обычно предлагают поставщики услуг кабельного телевидения. Данные передаются по тому же кабелю, который используется для передачи сигналов кабельного телевидения. Этот способ обеспечивает подключения к Интернету с высокой пропускной способностью и постоянным доступом к сети.
* **DSL**— цифровая абонентская линия обеспечивает подключение к Интернету с высокой пропускной способностью и постоянным доступом к сети. DSL использует телефонные линии связи. Обычно небольшие и домашние офисы используют асимметричные линии DSL (ADSL), в которых данные пользователю передаются с большей скоростью, чем от пользователя.
* **Сотовая связь**— для доступа в Интернет используется мобильная телефонная сеть. В любой точке, где доступен сигнал сотовой сети, можно получить доступ в Интернет. Производительность будет ограничена возможностями телефона и базовой станции, к которой он подключен.
* **Спутниковая связь**— спутниковые интернет-каналы можно использовать в районах, где нет других способов подключения. Для использования спутниковых антенн необходимо, чтобы спутник находился в зоне прямой видимости.
* **Телефонный коммутируемый доступ**— это экономичный вариант подключения с использованием любой телефонной линии и модема. Низкая пропускная способность коммутируемой линии обычно недостаточна для передачи большого объема данных. Однако такая линия может быть полезна для мобильного доступа в пути.

Для подключения домов и небольших офисов все чаще используются оптоволоконные кабели. Это позволяет интернет-провайдерам обеспечивать более высокие скорости передачи данных, а также предоставлять больше услуг, например Интернет, телефон и телевидение.

Способ подключения зависит от географического местоположения пользователей и наличия в регионе оператора связи.

**Подключение к Интернет для предприятий**

Корпоративные варианты подключения отличаются от вариантов для домашнего пользователя. Компании может требоваться более высокая пропускная способность, выделенная линия и управляемые услуги. Доступные варианты подключения зависят от технологий, которые используют операторы связи, находящиеся рядом.



На рисунке показаны стандартные варианты подключения для предприятий.

* **Выделенная арендованная линия**— арендованные линии представляют собой зарезервированные каналы в сети оператора связи, обеспечивающие связь между географически удаленными офисами для передачи голоса и данных в частной сети. Плата за аренду таких каналов связи обычно взимается ежемесячно или ежегодно. Они могут быть дорогими.
* **Глобальная сеть Ethernet**— глобальные сети Ethernet позволяют расширить сети LAN до WAN. О технологии локальных сетей Ethernet вы узнаете из следующих глав. Преимущества технологии Ethernet теперь доступны и в глобальных сетях.
* **DSL**— корпоративное DSL-подключение доступно в различных форматах. Популярностью пользуются симметричные цифровые абонентские линии (Symmetric Digital Subscriber Lines, SDSL), аналогичные абонентской версии DSL, но обеспечивающие одинаковую скорость при получении и отправке данных.
* **Спутниковая связь**— как и в случае небольших и домашних офисов, спутниковые услуги обеспечивают подключение там, где проводная связь недоступна.

Способ подключения зависит от географического местоположения пользователей и наличия в регионе оператора связи.

# 10. Интернет сервисы и услуги

# В настоящее время в интернете существует достаточно большое количество сервисов.

# Наиболее известными среди них являются:

# электронная почта (E-mail), обеспечивающая возможность обмена сообщениями одного человека с одним или несколькими абонентами;

# телеконференции, или группы новостей (Usenet), Предлагает форум для людей, чтобы обсудить темы, представляющие общие интересы; форумы

# сервис Telnet, предназначенный для управления удалёнными компьютерами в терминальном режиме;

# сервис IRC, Позволяет людям со всего мира общаться в режиме реального времени (chat);

# Коммуникационные сервисы

# Почта работает на основе протоколов. Протоколы электронной почты - это набор правил, которые помогают клиенту правильно передавать информацию на или с почтового сервера.

# SMTP расшифровывается как Simple Mail Transfer Protocol. Он был впервые предложен в 1982 году. Это стандартный протокол, используемый для эффективной и надежной отправки электронной почты через Интернет.

# IMAP - это протокол доступа к Интернет-почте.

# POP - это протокол почтового отделения. Обычно используется для поддержки одного клиента.

**Информационно-поисковые услуги**

# сервис FTP — система файловых архивов, обеспечивающая хранение и пересылку файлов различных типов;

# Потоковое мультимедиа.

# World Wide Web (WWW, W3, «Всемирная паутина») — гипертекстовая (гипермедиа) система, предназначенная для интеграции различных сетевых ресурсов в единое информационное пространство;

Перечисленные выше сервисы относятся к стандартным. Существуют и нестандартные сервисы, представляющие собой оригинальную разработку той или иной компании. В качестве примера можно привести различные системы типа InstantMessenger (своеобразные интернет-пейджеры — ICQ, AOL и т. п.), системы интернет-телефонии, трансляции радио и видео и т. д.

**World Wide Web (WWW)**

WWW также известен как W3. Он предоставляет возможность доступа к документам растекаться по нескольким серверам через Интернет. Эти документы могут содержать тексты, графику, аудио, видео, гиперссылок. Гиперссылки позволяют пользователям перемещаться между документами.

Техническое определение WWW:

**WorldWideWeb –** это ресурсы и пользователи Интернета, которые используют протокол передачи гипертекста (HTTP).

Всемирная паутина предоставляет способ обмена информацией между компьютерами в Интернете, связывая их вместе в обширную коллекцию интерактивных мультимедийных ресурсов.

Интернет и веб-это не то же самое: Web использует Интернет, чтобы передать информацию. WWW работает на клиент-серверном подходе.

Веб-браузер - это программное приложение, которое позволяет просматривать и исследовать информацию в Интернете. Пользователь может запросить любой адрес веб-страницы, просто введя URL в адресную строку.

Ранние веб-браузеры были текстовые. Ниже приведены наиболее распространенные веб-браузеры:, GoogleChrome, InternetExplorer , MozillaFirefox, NetscapeNavigator, Opera, Safari, SeaMonkey

**Поисковые системы**

Поисковые системы работают на основе:

- Индексирования. Индексная поисковая система (web indexing) собирает, анализирует и сохраняет данные, чтобы облегчить быстрый и точный поиск информации по темам.

- Web crawling . Иногда называемый пауком, является интернет-ботом, который систематически просматривает World Wide Web, как правило, для целей веб-индексации (веб-пауки).

**Веб-ресурсы**

**Веб-страница** - это документ, доступный в World Wide Web. Веб-страницы хранятся на веб-сервере и могут быть просмотрены с помощью веб-браузера.

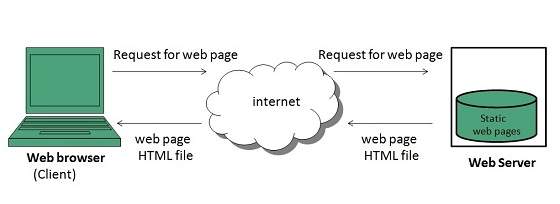
Веб-страница может содержать огромную информацию, включая текст, графику, аудио, видео и гиперссылки. Эти гиперссылки являются ссылкой на другие веб-страницы.

Коллекция связанных веб-страниц на веб-сервере называется **веб-сайтом**. С каждой веб-страницей связан уникальный унифицированный указатель ресурса (URL).

**Статическая веб-страница**

Статические веб-страницы также известны как плоские или стационарные веб-страницы. Они загружаются в браузер клиента точно так же, как они хранятся на веб-сервере. Такие веб-страницы содержат только статическую информацию. Пользователь может только читать информацию, но не может изменять или взаимодействовать с информацией.

Статические веб-страницы создаются с использованием только HTML. Статические веб-страницы используются только тогда, когда информация больше не требуется для изменения.



**Динамическая веб-страница**

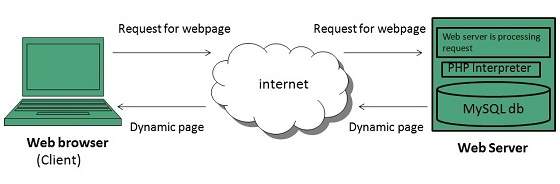
Динамическая веб-страница показывает различную информацию в разное время. Можно изменять части веб-страницы без загрузки всей веб-страницы. Это стало возможным благодаря технологии Ajax.

*Динамическая веб-страница на стороне сервера*

Он создается с помощью сценариев на стороне сервера. Существуют параметры сценариев на стороне сервера, которые определяют, как собрать новую веб-страницу, которая также включает настройку дополнительной обработки на стороне клиента.

*Динамическая веб-страница на стороне клиента*

Он обрабатывается с использованием клиентских скриптов, таких как JavaScript. И затем передается в Document Object Model (DOM).



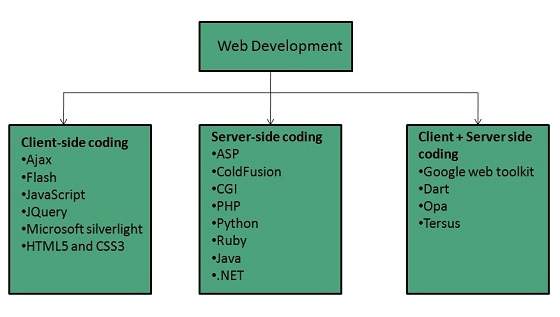
Веб-сервер - это компьютер, на котором хранится веб-контент. В основном веб-сервер используется для размещения веб-сайтов, но существуют и другие веб-серверы, такие как игры, хранилище, FTP, электронная почта и т. д.

**Разработка веб-сайта**

Веб-разработка относится к созданию веб-сайта и развертыванию в Интернете. Для веб-разработки требуется использование языков сценариев, как на стороне сервера, так и на стороне клиента.

Инструмент веб-разработки помогает разработчику тестировать и отлаживать веб-сайты. Теперь инструменты веб-разработки поставляются с веб-браузерами в качестве надстроек. Для этой цели у всех веб-браузеров есть встроенные инструменты.

Эти инструменты позволяют веб-разработчику использовать HTML, CSS и JavaScript и т. д. Доступ к ним осуществляется при наведении курсора на элемент на веб-странице и выборе «Inspect Element» из контекстного меню.



Прежде чем приступить к разработке веб-сайта, необходимо учесть несколько аспектов, например:

- Что ставить на веб-сайт?

- Как сделать его интерактивным?

- Как это кодировать?

- Будет ли дизайн веб-сайта хорошо отображаться в разных браузерах?

- Удобны ли навигационные меню?

- Как легко посетители смогут найти важные детали, относящиеся к веб-сайту?

- Насколько эффективны таблицы стилей на ваших веб-сайтах?

**Веб-разработка**

Процесс веб-разработки включает в себя все шаги, которые можно предпринять, чтобы создать привлекательный, эффективный и быстро реагирующий веб-сайт. Эти шаги:

- Определить цель

- Планирование

- Проектирование

- Сбор контента

- Разработка

- Тестирование

- Загрузка на сервер

- Маркетинг и обслуживание

***Основная литература***

1. Дронов В.А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше Х.Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 2. Структура Web технологий**

План

# 1. Основные понятия Всемирной паутины

# 2. Основные компоненты технологии WWW

# 2.1 HTML - язык разметки гипертекста.

# 2.2. URL - унифицированный указатель ресурса

# 2.3 HTTP протокол обмена гипертекстовой информацией

# 2.4 Универсальный интерфейс шлюзов CGI

# 3. Клиент-серверные веб-технологии

# 4. Протокол передачи гипертекста - HTTP

# 5. Защита передачи данных HTTP

# 6. Механизм Cookie

**1. Основные понятия Всемирной паутины (World Wide Web (WWW))**

WWW также известен как W3. Он предоставляет возможность доступа к документам растекаться по нескольким серверам через Интернет. Эти документы могут содержать тексты, графику, аудио, видео, гиперссылок. Гиперссылки позволяют пользователям перемещаться между документами.

*Техническое определение:* ***WorldWideWeb – это все ресурсы и пользователи Интернета, которые используют протокол передачи гипертекста (HTTP).***

Всемирная паутина предоставляет способ обмена информацией между компьютерами в Интернете, связывая их вместе в обширную коллекцию интерактивных мультимедийных ресурсов.

Интернет и WWW -это не то же самое: Web использует Интернет, чтобы передавать и размещать информацию

**2 Основные компоненты технологии WWW**

**2.1) HTML** - **язык гипертекстовой разметки HTML** (HyperText Markup Language).

В состав HTML входит набор команд, используемых для описания структуры доку­мента. С помощью HTML документ разбивается на соответствующие логические компо­ненты: абзацы, заголовки, списки и т.д.

Конкретные атрибуты форматирования доку­мента (основного текста и выделенных компонентов) при его просмотре определяются используемым броузером.

Примеры броузеров являются:

• Mosaic для Windows (1993)

• Microsoft Internet Explorer (MS IE) (1995) [Написано на](https://www.google.com/search?sxsrf=ACYBGNQ_6ubQCvIG3UeQsix-uVn-DYpbFA:1577464894950&q=internet+explorer+%D0%BD%D0%B0%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%BE+%D0%BD%D0%B0&stick=H4sIAAAAAAAAAOPgE-LQz9U3MK4oN9BSyii30k_Oz8lJTS7JzM_TL85PKylPLEq1Ki_KLClJzVPIzFvEqp6ZV5JalJdaopBaUZCTX5RapHBh74UNF_Zf2HGxEUjvvbAPLAAAz6U0VVoAAAA&sa=X&ved=2ahUKEwjB7NXVotbmAhUlxaYKHcm8DgEQ6BMoADATegQIDRAJ&sxsrf=ACYBGNQ_6ubQCvIG3UeQsix-uVn-DYpbFA:1577464894950)**:**[C++](https://www.google.com/search?sxsrf=ACYBGNQ_6ubQCvIG3UeQsix-uVn-DYpbFA:1577464894950&q=C%2B%2B&stick=H4sIAAAAAAAAAOPgE-LQz9U3MK4oN1ACs7LSC9O1lDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrcqLMktKUvMUMvMWsTI7a2sDAJvb8rBAAAAA&sa=X&ved=2ahUKEwjB7NXVotbmAhUlxaYKHcm8DgEQmxMoATATegQIDRAK&sxsrf=ACYBGNQ_6ubQCvIG3UeQsix-uVn-DYpbFA:1577464894950)

• Netscape Navigator (компания Netscape, 1994-2007);

- Opera (1995).

- Mozilla FireFox (2002)

- Google Chrome (2008)

**Microsoft Edge** — быстрый браузер для Windows 10 (2015))

**2.2) URL универсальный способ адресации ресурсов в сети URL;** URL (uniform resource locator), универсальный указатель ресурсов, т.е. ссылки на информационные ресурсы Интернет. Каждый интернет-ресурс имеет свой собственный идентификатор местоположения или URL (Uniform Resource Locator).

Для указания адресов интернет-ресурсов применяются унифицированный идентификатор ресурса (URI) и унифицированный указатель ресурса (URL)

Теперь URL позиционируется как часть более общей системы идентификации ресурса URI.

Стандарт URL-адресов регулируется организацией **IETF** и ее отделами.

Универсальный идентификатор ресурса (URI) - это строка символов, используемая для идентификации ресурса.

Uniform Resource Locator (URL) определяет сервер для доступа, а также метод доступа и расположение ресурса на сервере.

**2.3) Протокол обмена гипертекстовой информацией HTTP;** Это название протокола, по которому взаимодействуют клиент и сервер WWW.

Данный протокол предназначен для обмена гипертекстовыми документами и учитывает специфику такого обмена.

**2.4) универсальный интерфейс шлюзов CGI –** используетсядля организации **диалога с пользователем.**

CGI была специально разработана для расширения возможностей WWW за счет подключения всевозможного внешнего программного обеспечения.

Программы используют CGI для получения информации от пользователя (через протокол HTTP), для ее обработки и отправки обратно в виде нового документа HTML. При необходимости программа CGI обращается к СУБД или другим программным системам, работающим на сервере.

### **3. Клиент-серверные Web** **технологии**

WWW работает на клиент-серверном подходе. Базовым протоколом сети гипертекстовых ресурсов Веб является протокол *HTTP*.

В его основу положено взаимодействие "*клиент-сервер*", то есть предполагается, что (рис.1):

1. Потребитель-*клиент*, инициировав соединение с поставщиком-сервером, посылает ему запрос;
2. Поставщик-*сервер*, получив запрос, производит необходимые действия и возвращает обратно клиенту ответ с результатом.

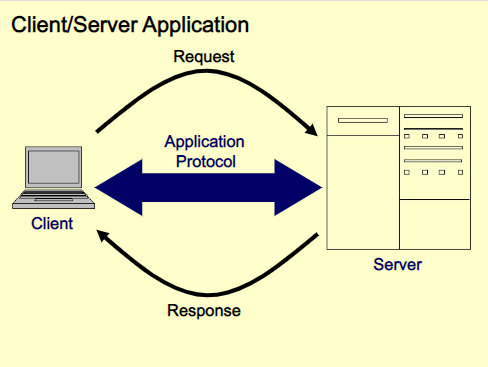


Рисунок 1

При этом возможны два способа организации работы компьютера-клиента:

* ***Тонкий клиент* - это компьютер-клиент, который переносит все задачи по обработке информации на сервер**. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями.
* ***Толстый клиент*, напротив, производит обработку информации независимо от сервера**, использует последний в основном лишь для хранения данных.

**WWW работает по принципу клиент-сервер, точнее, клиент-серверы**: существует множество серверов, которые по запросу клиента возвращают ему гипермедийный документ - документ, состоящий из частей с разнообразным представлением информации (текст, звук, графика, трехмерные объекты и т.д.), в котором каждый элемент может являться ссылкой на другой документ или его часть.

Ссылки эти в документах WWW организованы таким образом, что каждый информационный ресурс в глобальной сети Интернет однозначно адресуется, и документ, который вы читаете в данный момент, способен ссылаться как на другие документы на этом же сервере, так и на документы (и вообще на ресурсы Интернет) на других компьютерах Интернет. Причем пользователь не замечает этого, и работает со всем информационным пространством Интернет как с единым целым.

Ссылки WWW указывают не только на документы, специфичные для самой WWW, но и на прочие сервисы и информационные ресурсы Интернет.

Более того, большинство программ-клиентов WWW (browsers, навигаторы) не просто понимают такие ссылки, но и являются программами-клиентами соответствующих сервисов: ftp, gopher, сетевых новостей Usenet, электронной почты и т.д. Таким образом, программные средства WWW являются универсальными для различных сервисов Интернет, а сама информационная система WWW играет интегрирующую роль.

***Web-сервер*** – это подключенный к Интернету компьютер, на котором выполняется специальная программа, также называемая Web-сервером. В задачи этой программы входит хранение, поиск и распределение определенных файлов.

***Клиент Web*** – это браузер, запрашивающий файлы из Web.

Когда клиентский компьютер хочет получить доступ к одному из файлов, запрос направляется на Web-сервер, в зону ответственности которого попадает данный файл. Сервер отыскивает указанный файл и отправляет его клиентскому компьютеру, передавшему запрос.

**Принципы работы Web-серверов:**

- информация хранится на сервере Web в виде страниц, на которых кроме текста могут находиться различные объекты;

- страницы сервера Web содержат гиперссылки на другие страницы;

- страницы сервера Web содержат не только статичную, но и динамическую информацию;

- сервер Web способен принимать данные от пользователя, обрабатывать их и передавать их обратно.

Технология Web позволяет организовать разветвленную сеть ссылок между отдельными документами и объектами, хранящимися в каталогах серверов Web.

**Логически Web-документ** представляет собой гипертекстовый документ, объединяющий ссылками различные Web-страницы, каждая из которых может содержать ссылки на другие объекты.

**Физически Web-документ** представляет собой текстовый файл специального формата, содержащий ссылки на другие объекты и Web-документы, расположенные в любом узле сети.

**Схема взаимодействия клиента с Web-сервером**

При получении запроса от Web-клиентов программа Web-сервер запускается на компьютере-сервере и осуществляет обработку запросов.

Взаимо­действие между Web-клиентом и Web-сервером осуществляется в соответст­вии **с протоколом HTTP** (HyperText Transfer Protocol — протокол передачи гипертекста).

# *При получении запроса от Web-клиента, Web-сервер устанавливает связь по протоколу TCP/IP и обменивается информацией в соответствии с протоко­лом HTTP.*

# *В случае запроса защищенной информации Web-сервер может потребовать от пользователя введения идентификатора и пароля.*

# Защищен­ные Web-документы предоставляются только при наличии у пользователей соответствующих прав доступа (рисунок 2).

# 

# Рисунок 2. Обработка запроса от Web-клиента

**Многие Web-серверы интегрированы с базами данных.** При этом информация, хранящаяся в таких базах данных, становится доступной посетителям сервера, учитывая права доступа.

Взаимодействие браузера с сервером СУБД может осуществляться двумя основными способами:

- доступ к серверу СУБД через Web-сервер;

- доступ к серверу СУБД напрямую.

**Распределенная обработка информации на основе мигрирующих программ**

Одной из главных особенностей Web-технологий (intranet-архитектуры) является распределен­ная обработка информации на основе мигрирующих программ.

Программа навигации, выполняемая на рабочей станции, может

* не только визуализировать Web-страницы и выполнять переходы к другим ресурсам,
* но и активизировать программы на сервере,
* а также интерпретировать и запускать на выполнение программы, относящиеся к Web-документу, которые передаются вместе с этим документом с сервера.

Такой вид распределенной обработки информации позволяет сконцентрировать всю прикладную систему непосредственно на сервере.

**Существует три основных вида программ, которые могут быть связаны Web-документом и передаваться на рабочую станцию для выполнения**:

- Java-апплеты, подготовленные и используемые по технологии Java;

- программы, написанные на языке сценариев JavaScript, VBScript (Visual Basic Scripting) или VRML;

- программные компоненты ActiveX Controls, соответствующие технологии ActiveX.

Наличие нескольких разновидностей мигрирующих программ объясняется их различными возможностями, а также конкуренцией между ведущим корпорациями в области программных и сетевых технологий — корпорациями Sun Microsystems, Netscape, Microsoft и другими.

# 4 Протокол передачи гипертекста

Взаимо­действие между Web-клиентом и Web-сервером осуществляется в соответст­вии **с протоколом HTTP** (HyperText Transfer Protocol — протокол передачи гипертекста).

Данный протокол предназначен для обмена гипертекстовыми документами и учитывает специфику такого обмена.

**Web-документы, получаемые навигатором от Web-сервера, представляют собой текстовые файлы, написанные на языке *HTML***

*Так в процессе взаимодействия, клиент может получить новый адрес ресурса на сети (relocation), запросить встроенную графику, принять и передать параметры и т.п*

Управление в HTTP реализовано в виде ASCII-команд.

# Чтобы получить доступ к той или иной странице, пользователь должен указать ее адрес URL программе броузера.

# Браузер устанавливает соединение с сервером, применяя протокол http, далее страница начинает загружаться с сервера, после чего соединение разрывается. Принятая страница отображается в окне браузера.

# HTTP (HyperText Transfer Protocol - RFC 1945, RFC 2616) - протокол прикладного уровня для передачи гипертекста.

# Центральным объектом в HTTP является ресурс, на который указывает URI в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы.

# Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т. д.

# Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя изначально данный протокол предназначен для передачи символьной информации.

# В отличие от многих других протоколов, HTTP является протоколом без памяти.

# Это означает, что протокол не хранит информацию о предыдущих запросах клиентов и ответах сервера.

# Компоненты, использующие HTTP, могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами.

# Например, клиентское веб-приложение, посылающее запросы, может отслеживать задержки ответов, а веб-сервер может хранить IP-адреса и заголовки запросов последних клиентов.

# Всё программное обеспечение для работы с протоколом HTTP разделяется на три основные категории:

# Серверы - поставщики услуг хранения и обработки информации (обработка запросов).

# Клиенты - конечные потребители услуг сервера (отправка запросов).

# Прокси-серверы для поддержки работы транспортных служб.

# "Классическая" схема HTTP-сеанса выглядит так.

# Установление TCP-соединения.

# Запрос клиента.

# Ответ сервера.

# Разрыв TCP-соединения.

# Таким образом, клиент посылает серверу запрос, получает от него ответ, после чего взаимодействие прекращается.

# Обычно запрос клиента представляет собой требование передать HTML-документ или какой-нибудь другой ресурс, а ответ сервера содержит код этого ресурса.

# В состав HTTP-запроса, передаваемого клиентом серверу, входят следующие компоненты.

# Строка состояния (иногда для ее обозначения используют также термины строка-статус, или строка запроса).

# Поля заголовка.

# Пустая строка.

# Тело запроса.

# Строку состояния вместе с полями заголовка иногда называют также заголовком запроса (Рисунок 3)

# 

Рисунок 3

HTTP используется в других протоколах прикладного уровня, а также для передачи с сервера на клиент любых объектов: картинок, скриптов, CSS-файлов, файлов данных.

Также он работает и в обратную сторону — для заливки на сервер файлов, отправки форм и т.п.

Существуют протоколы, аналогичные HTTP, такие как FTP и SMTP. Чтобы идентифицировать ресурсы, HTTP-протокол использует глобальные URL.

HTTP не сохраняет своего состояния между парами "запрос-ответ". Компоненты HTTP могут сохранять информацию. Например, со стороны клиента сохраняются "куки", а на стороне сервера "сессии".

Браузер, в свою очередь, может отслеживать задержки ответов, а сервер хранит IP-адреса запросов клиентов.

### **5. Защита передачи данных HTTP**

### **Обеспечение безопасности передачи данных HTTP -** протокол *HTTP* предназначен для передачи символьных данных в открытом (незашифрованном) виде

Существует ряд расширений базового *HTTP*  протокола направленных на повышение защищенности *интернет*-трафика от несанкционированного доступа.

*HTTPS*  протокол – это расширение протокола *HTTP*, при котором данные "упаковываются" в криптографический протокол *SSL* или *TLS*, тем самым обеспечивая защиту этих данных.

*SSL* (*Secure* Sockets *Layer*) - криптографический протокол, обеспечивающий безопасную передачу данных по сети *Интернет*.

При его использовании создается защищенное соединение между клиентом и сервером. 

*SSL* изначально разработан компанией Netscape *Communications*. Впоследствии на основании протокола *SSL* 3.0 был разработан и принят стандарт *RFC*, получивший название ***TLS***.

***TLS*** протокол использует *шифрование* с открытым ключом для подтверждения подлинности передатчика и получателя.

Поддерживает *надежность* передачи данных за счет использования корректирующих кодов и безопасных хэш-функций.

Для работы *SSL* требуется, чтобы на сервере имелся *SSL* *-сертификат*.

### **6 Механизм  Cookie**

Механизм cookie позволяет серверу хранить информацию на компьютере клиента и извлекать ее оттуда.

**Ку́ки** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *cookie*, буквально — печенье) — небольшой фрагмент данных, отправленный [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и хранимый на [компьютере](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) пользователя.

Веб-клиент (обычно [веб-браузер](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80)) всякий раз при попытке открыть страницу соответствующего сайта пересылает этот фрагмент данных веб-серверу в составе [HTTP](https://ru.wikipedia.org/wiki/HTTP)-запроса.

Применяется для сохранения данных на стороне пользователя, на практике обычно используется для:

* [аутентификации](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) пользователя;
* хранения персональных предпочтений и настроек пользователя;
* отслеживания состояния [сеанса](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D0%B0%D0%BD%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)&action=edit&redlink=1)[[en]](https://en.wikipedia.org/wiki/Session_(computer_science)) доступа пользователя;
* сведения статистики о [пользователях](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C).

Поддержки браузерами cookie (приём, сохранение и последующая пересылка серверу сохранённых cookie) требуют многие сайты с ограничениями доступа, большинство [интернет-магазинов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD)[[2]](https://ru.wikipedia.org/wiki/Cookie#cite_note-2).

Настройка оформления и поведения многих [веб-сайтов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82) по индивидуальным предпочтениям пользователя тоже основана на cookie

Тем не менее, если вы когда-нибудь пользовались почтовым ящиком на [mail.ru](http://mail.ru/) или на другом сервере, предоставляющем почтовые услуги пользователям Веб, вспомните, как вел себя клиент после того, как вы создали для себя *почтовый ящик* на сервере.

Когда вы в следующий раз обратились с того же компьютера к [mail.ru](http://mail.ru/), вы, вероятно, заметили, что после загрузки веб-страницы ваше регистрационное имя уже отображалось в соответствующем *поле* ввода.

Такие сведения позволяет получить дополнительное средство под названием cookie. Инициатором записи cookie выступает *сервер*.

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 3. Web приложения**

План

1. Концепция веб-приложений

1.1 Программы, работающие на клиентском компьютере

1.2. Программы, работающие на сервере

2. Типы веб-приложений

3. Веб-приложения - мигрирующие программы

3.1 Технология Java

3.2 Скриптовые технологии

3.3 Технологии ActiveX

4 Доступ к реляционным базам данных

4.1 Доступ к серверу СУБД через веб-сервер

4.2 Доступ к серверу СУБД напрямую

# 1. Концепция Web приложений

# Веб-приложение — обеспечивает генерацию HTML-страницы на лету в зависимости от запроса пользователя. Почтовые клиенты, соцсети, поисковики, интернет-магазины, онлайн-программы для бизнеса, это все веб-приложения.

# Веб-приложение - это приложение, в котором клиент является браузером, а сервер - веб-сервером.

# Результатом веб-приложения является веб-страница, отображаемая в окне браузера.

# В этом случае само веб-приложение может работать как на клиентском компьютере, так и на сервере.

# 1.1. Программы, работающие на клиентской машине

# 1) Статичные HTML-страницы

# 2) Скрипты JavaScript - тип программ, предназначенных для выполнения на клиентском компьютере

# Скрипты (сценарии) – это отдельные последовательности действий, созданные для автоматического выполнения задачи.

# Исходный текст сценария является частью веб-страницы, поэтому сценарий JavaScript передается клиенту вместе с документом, в который он включен.

# При обработке документа HTML браузер обнаруживает исходный текст скрипта и запускает его для выполнения.

# Для всех программ, которые передаются с сервера на клиентские машины и запускаются для выполнения, предъявляется одно общее требование: эти программы должны быть защищены от доступа к ресурсам компьютера, на котором они работают. Это требование полностью оправдано.

# В конце концов, отправка по сети и запуск апплетов Java и сценариев JavaScript происходит автоматически без вмешательства пользователя, поэтому работа этих программ должна быть абсолютно безопасной для компьютера. Другими словами, языки, предназначенные для создания программ, запущенных на клиентском компьютере, должны быть совершенно непригодны для написания вирусов и подобных программ.

# 1.2. Программы, работающие на сервере

# Код программы, запущенной на сервере, не передается клиенту.

# При получении специального запроса от клиента, предполагающего выполнение такой программы, сервер запускает ее и передает параметры, включенные в запрос.

# Программа подготавливает результаты своей работы в виде HTML-документа и передает его на веб-сервер, а последний, в свою очередь, дополняет полученные данные HTTP-заголовком и передает их клиенту.

# Чтобы расширить возможности взаимодействия клиент-сервер по протоколу HTTP, можно разрабатывать приложения, плагины и сценарии, расширяющие возможности сервера - на стороне веб-сервера.

# Плагин (plug-in) - это независимо скомпилированный программный модуль, динамически подключаемый к основной программе, предназначенный для расширения или использования его возможностей. Обычно выполняется как общие библиотеки.

# Скрипт (сценарий) - это программа, которая автоматизирует некоторые задачи, которые пользователь выполняет вручную, используя программные интерфейсы.

# Серверные Скрипты выполняются на сервере перед отправкой интернет-страницы пользователю (пишутся на PHP, Perl, Visual Basic, Java, Ruby или Python)

# Веб-сервис - это программная система с идентификатором URI, открытые интерфейсы которой определены на языке XML.

# Веб-сервис - это единица модульности при использовании сервис-ориентированной архитектуры приложений.

**2. Типы веб-приложений**

Веб-приложения можно разделить на несколько типов, в зависимости от разных сочетаний его основных составляющих:

1. Backend (бэкенд или серверная часть приложения) работает на удаленном компьютере, который может находиться где угодно. Она может быть написана на разных языках программирования: PHP, Python, Ruby, C# и других.

Если создавать приложение используя только серверную часть, то в результате любых переходов между разделами, отправок форм, обновления данных, сервером будет генерироваться новый HTML-файл и страница в браузере будет перезагружаться.

1. Frontend (фронтенд или клиентская часть приложения) выполняется в браузере пользователя. Эта часть написана на языке программирования Javascript.

Приложение может состоять только из клиентской части, если не требуется хранить данные пользователя дольше одной сессии. Это могут быть, например, фоторедакторы или простые игрушки.

1. Single page application (SPA или одностраничное приложение). Более интересный вариант, когда используются и бэкенд и фронтенд. С помощью их взаимодействия можно создать приложение, которое будет работать совсем без перезагрузок страницы в браузере.

Или в упрощенном варианте, когда переходы между разделами вызывают перезагрузки, но любые действия в разделе обходятся без них.

1. Pyhon-фреймворк Django aka бэкенд

# фреймворк — это набор готовых библиотек и инструментов, которые помогают создавать веб-приложения.

5. Javascript-фреймворки aka фронтенд: **Backbone Marionette, Angular, React, Vue и другие.**

Клиентская часть приложения — это скрипты, написанные на языке программирования Javascript (JS) и исполняемые в браузере пользователя.

Раньше вся клиентская логика основывалась на использовании библиотеки JQuery, которая позволяет работать с DOM, анимацией на странице и делать AJAX запросы.

DOM (document object model) — это структура HTML-страницы. Работа с DOM — это поиск, добавление, изменение, перемещеие и удаление HTML-тегов.

AJAX (asynchronous javascript and XML) — это общее название для технологий, которые позволяют делать асинхронные (без перезагрузки страницы) запросы к серверу и обмениваться данными. Так как клиентская и серверная части веб-приложения написаны на разных языках программирования, то для обмена информацией необходимо преобразовывать структуры данных (например, списки и словари), в которых она хранится, в JSON-формат.

JSON (JavaScript Object Notation) — это универсальный формат для обмена данными между клиентом и сервером. Он представляет собой простую строку, которая может быть использована в любом языке программирования.

# С помощью манипуляций с DOM можно полностью управлять содержимым страниц. С помощью AJAX можно обмениваться данными между клиентом и сервером

# На смену JQuery пришли Javascript-фреймворки: Backbone Marionette, Angular, React, Vue и другие.

# 3. Web-приложения - мигрирующие программы

# Программа навигации (броузер), выполняемая на рабочей станции выполняет следующие действия:

# - визуализирует (отображает) Web-страницы и выполнять переходы к другим ресурсам

# - активизировать программы на сервере

# - интерпретирует и запускает на выполнение программы, относящиеся к Web-документу, которые передаются вместе с этим документом с сервера.

# Такой вид распределенной обработки информации позволяет сконцентрировать всю прикладную систему непосредственно на сервере.

# Существует три основных вида программ, которые могут быть связаны с Web-документом и передаваться на рабочую станцию для выполнения:

# - Java-апплеты, подготовленные и используемые по технологии Java;

# - программы, написанные на языке сценариев JavaScript, VBScript, VRML;

# - программные компоненты ActiveX Controls, соответствующие технологии ActiveX

# Наличие нескольких разновидностей мигрирующих программ объясняется их различными возможностями, а также конкуренцией между ведущими корпорациями в области программных и сетевых технологий — корпорациями Sun Microsystems, Netscape, Microsoft и другими

Кроме этих, существует еще ряд других Web-технологий:

- **Технология CGI** (Common Gateway Interface) - использование в составе ресурса Интернет интерактивных элементов на базе приложений, обеспечивающих передачу потока данных от одного объекта к другому.

Технология CGI обычно реализуется двумя методами: либо с использованием программ, написанных на языке PERL, либо с помощью приложений, созданных обычно с применением языка С и откомпилированных непосредственно на сервере.

- **Технология SSI** (Server Side Includes) –технология тесно переплетающаяся с технологией CGI, позволяет включать в страницы серверные файлы, а также устанавливать и использовать значения некоторых параметров.

- **Технология CSS (**Cascading Style Sheets) – каскадные таблицы стилей служат для централизованной настройки внешнего вида страниц.

- **PHP (Personal Home Page Tools)** – интерпретируемый язык, предназначенный для придания элементам Web-страницы интерактивности.

- **ASP (Active Server Page)** - активные страницы сервера, т.е. скрипт интерпретируется и исполняется непосредственно на сервере, после чего пользовательскому браузеру отправляется уже готовый html-документ с результатами работы сценария ASP.

- **Macromedia Flash –** позволяет создавать анимированные изображения, заставки. В среду Flash встроен специальный язык программирования. Написанные с его помощью программные модули импортируются в документ как апплеты и вставляются в нужный кадр анимации, где должно произойти динамическое изменение изображения.

- DHTML (Dynamic Hyper Text MakeUP Language) – расширение стандарта HTML и позволяет создавать Web-документы, включающие интерактивные элементы: движущийся фон, выпадающие меню, кнопки и др.

**3.1 Java-технология**

Технология Java была разработана компанией Sun Microsystems в начале 90-годов в связи с необходимостью разработки в компьютерных программ, ориентированных на использование в сетевой среде и интеграцию с Web-сервисом.

К таким программам изначально были предъявлены требования по мобильности, предполагающие независимость от аппаратных и операционных платформ, а также безопасность и надежность обработки информации.

В результате были разработаны язык программирования Java, а также целостная технология создания и использования мобильных программ, получившая название Java-технологии.

Язык Java является простым объектно-ориентированным языком программирования, построенным на основе языка C++.

Виртуальный Java-процессор обеспечивает среду для исполнения Java-программ.

Оттранслированные Java-программы, предназначенные для выполнения на рабочей станции в среде Web-навигатора, называют Java-аплетами или про¬сто аплетами.

По своей структуре каждый аплет представляет собой небольшую программу, в которой должно быть определено несколько обязательных функций.

Аплет загружается по сети с сервера и выполняется в среде Web-навигатора (рис. 1). Ссылки на аплеты располагаются в Web-документах, но непосредственно в состав Web-документов аплеты не входят. Они хранятся в отдельных файлах на сервере.



Рисунок 1.Схема передачи и выполнения машинно-независимых Java-программ

Независимость байт-кодов Java от аппаратно-операционных платформ достигается программной реализацией для каждой из этих платформ только виртуального Java-процессора, который и предназначен для интерпретации аплетов.

# 3.2 Технологии, основанные на использовании языков сценариев

# Параллельно с мощной Java-технологией появились технологии создания и применения мигрирующих программ, основанные на использовании языков сценариев.

# Наиболее важным отличием таких технологий от Java-технологии является покомандная интерпретация исходных текстов программ, исключающая необходимость их компиляции перед выполнением.

# Функция интерпретации мобильных программ, написанных на языке сценариев, возложена на Web-навигатор.

# Языки сценариев часто называют еще языками скриптов (script — сценарий) или макроязыками.

# К основным представителям языков сценариев, предназначенных для написания мигрирующих программ, относятся:

# - язык JavaScript, разработанный совместно компаниями Netscape и Sun Microsystems, а также подобный ему язык VBScript (Visual Basic Scripting) от Microsoft;

# - язык VRML (Virtual Reality Modeling Language — язык моделирования виртуальной реальности), разработанный компанией Silicon Graphics.

# JavaScript является упрощенным интерпретируемым языком с базовыми объектно-ориентированными функциями.

# Свойство простоты объясняется отсутствием жесткой архитектуры типов и семантики.

# Объектно-ориентированная ориентация проявляется в возможностях работы с окнами, строкой состояния и другими элементами интерфейса Web-навигатора и сетевого окружения как с объектами в иерархии, к которым можно обращаться по имени.

# JavaScript беднее языка Java, но гораздо удобнее и эффективнее для ряда задач, связанных с обработкой Web-документов и взаимодействием с пользователем при его просмотре. Он имеет большое число встроенных функций и команд. Программы, написанные с помощью JavaScript, могут выводить на экран диалоговые окна, производить математические вычисления, проигрывать различные аудио- и видеофайлы, получать новые документы, обрабатывать нажатие на кнопки в формах и многое другое.

# С помощью JavaScript можно также устанавливать атрибуты и свойства бинарных библиотек Java, a также программных модулей (plug-ins), подключенных к Web-навигатору.

# Команды JavaScript встраиваются непосредственно в Web-страницу и выполняются Web-навигатором во время загрузки этой страницы или во время определенных действий, производимых пользователем при работе с ней, *например, при щелчке мышью на одном из объектов страницы, при пози­ционировании указателя в место расположения ссылки или при вводе ин­формации в поля HTML-формы*

# К недостаткам технологии JavaScript следует отнести невысокое быстродействие JavaScript-программ, являющееся неотъемлемым атрибутом всех интерпретируемых языков программирования.

# Язык сценариев VBScript (Visual Basic Scripting) от Microsoft во многом подобен JavaScript. Он является подмножеством языка Visual Basic и также предназначен для программирования страниц Web.

# С его помощью можно заставить взаимодействовать разные объекты на Web-странице, включая программные компоненты другого типа, например, аплеты Java и программные компоненты ActiveX Controls.

# В отличие от макроязыков JavaScript и VBScript язык VRML разработан корпорацией Silicon Graphics специально для создания интерпретируемых программ, моделирующих трехмерные виртуальные миры. Интерпретаторы VRML подключаются к Web-навигаторам чаще всего в виде отдельных программных модулей (plug-ins). Исходные тексты программ на языке VRML оформляются в виде отдельного VRML-файла и вызываются по ссылке с Web-документа при его просмотре Web-навигатором. Щелчок мышью по такой ссылке приводит к открытию отдельного окна, позволяющего пройтись по расположенному в нем фрагменту трехмерной реальности.

# 3.3 Технологии ActiveX

# Под ActiveX понимается набор технологий от Microsoft, направленных на дополнение, интеграцию и унификацию существующих методов представления и обработки информации в компьютерных сетях, построенных по Web-архитектуре. Основная идея ActiveX-технологий заключается в использовании одинакового способа доступа ко всем информационным ресурсам сети (рис. 2).

# 

# Рис. 2. Схема единообразного доступа к информационным ресурсам сети

ActiveX обеспечивает не только разработку и выполнение мобильных программ, но и реализацию ряда дополнительных возможностей, например, вызов из Web-броузера функций по просмотру и редактированию документов Word, Excel и Power-Point. ActiveX предоставляет набор функций API (Application Program Interface), реализованный как для клиента, так и для сервера.

ActiveX поддерживает следующие типы мобильных программ, которые могут быть связаны с Web-документом и передаваться на рабочую станцию для выполнения (рис. 3):

- программные компоненты ActiveX Controls;

- аплеты Java;

- программы, написанные на языках сценариев JavaScript, VBScript (Visual Basic Scripting) и VRML.

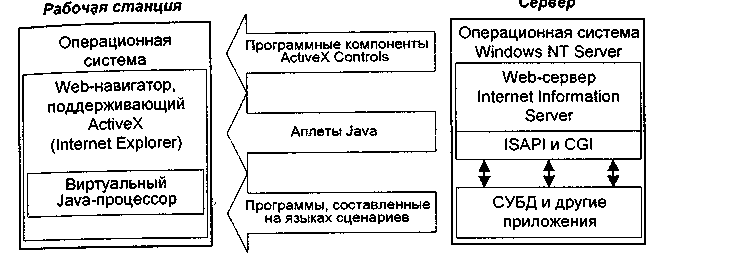


Рисунок 3. Миграция программ в технологии ActiveX

# Взаимодействие Web-сервера IIS с другими приложениями, например с системой управления базами данных (СУБД), обеспечивается за счет реализованных в нем интерфейсов ISAPI (Internet Server API) и CGI (Common Gateway Interface).

# Программные компоненты ActiveX Controls представляют собой обычные исполняемые программы, которые могут загружаться с сервера для исполнения на рабочей станции. Как и при использовании Java-аплетов ссылки на эти программы располагаются в Web-документах.

# 4. Доступ к реляционным базам данных

# В архитектуре "клиент-сервер", основанной на Web-технологии, Web-сервер доставляет информацию из разных источников, а потом с помощью Web-навигатора предоставляет ее пользователю.

# Непосредственная интеграция разнородной информации выполняется при визуализации и интерпретации Web-документов, которую реализует Web-навигатор при взаимодействии с Web-сервером, а также другими серверами по предоставлению информационных ресурсов.

# Взаимодействие Web-навигатора с сервером системы управления базами данных (сервером СУБД) может осуществляться двумя основными способами:

# - доступ к серверу СУБД через Web-сервер;

# - доступ к серверу СУБД напрямую.

# 4.1. Доступ к серверу СУБД через Web-сервер

# Для доступа Web-навигатора к серверу СУБД через Web-сервер используется система программных шлюзов (см. рис. 4)

# Программный шлюз, получив запрос от Web-сервера, выступает в качестве посредника между сервером Web и сервером СУБД.

# Программные шлюзы разрабатываются в соответствии с определенными стандартами, определяющими способы вызова Web-сервером прикладных программ или функций динамических библиотек, а также способы обмена информацией с этими программными объектами.

# Одними из наиболее распространенных стандартов данного типа являются интерфейс CGI (Common Gateway Interface — общий интерфейс шлюзов), а также его усовершенствованная спецификация, названная как FastCGI (ускоренный CGI).

# Для доступа Web-броузера к серверу СУБД через Web-сервер по стандарту CGI необходима соответствующая CGI-программа, выполняющая роль программного шлюза между Web-сервером и сервером СУБД

# CGI-приложения работают независимо от Web-сервера, а их запуск осуществляется по вызову с Web-документа при его обработке Web-навигатором. CGI-программа взаимодействует с Web-сервером посредством двустороннего обмена переменными среды через стандартные каналы ввода/вывода данного приложения.

# CGI-интерфейс (Common Gateway Interface) — одно из первых решений, созданных. для доставки динамической Web-информации, он до сих пор очень популярен в мире UNIX.

# CGI-приложения — это программы, исполняемые на Web-сервере и обычно используемые для предоставления динамической Web-информации.

# CGI-программы – это обычные консольные приложения, работающие со стандартным устройством вывода. Стандартное устройство переназначается, и всё, что посылается на него программой, выводится на Web-странице.

# CGI-программа способна выполнять практически все: получать доступ к базам данных, считывать файлы, работать с системным реестром и все остальное, что обычно делает Win32-программа. Для создания такой программы можно воспользоваться любым языком программирования или средой разработки, позволяющей создать консольное приложение, считывающее со стандартного устройства ввода и записывающее в стандартное устройство вывода. В UNIX для создания CGI-приложений применяется Perl, а среди поклонников Win32 популярна среда Borland Delphi, которая явно поддерживает CGI-Приложения, предоставляя классы, управляющие чтением и записью в стандартные устройства.

# После завершения работы CGI-программу можно так же, как и любую другую программу, изменять или удалять; однако эта возможность является и недостатком CGI. При исполнении CGI-программа загружается, а по завершении работы полностью удаляется из памяти. Много ресурсов тратится на создание и удаление процессов. Создание процесса — более ресурсоемкая операция по сравнению, скажем, с простым считыванием HTML-файла. В конечном счете, подобное создание и удаление процессов сильно снижает производительность, а также перегружает ресурсы. Когда 100 клиентов одновременно обращаются к одной CGI -программе, в памяти создается 100 экземпляров этой программы. Ресурсы Web-сервера активно расходуются, что отрицательно сказывается на масштабируемости.

# Интерфейсы API и FastCGI

# Для того чтобы обойти проблемы, связанные с быстродействием CGI, разработаны интерфейсы прикладного программирования (API).

# Корпорацией Microsoft был разработан интерфейс ISAPI (Internet Server API), а корпорацией Netscape — интерфейс NSAPI (Netscape Server API).

# Эти интерфейсы тесно интегрированы с Web-сервером, позволяя сохранять доступность постоянно используемых процессов и данных. Программы с интерфейсом ISAPI компилируются в файлы динамически подключаемых библиотек DLL.

# 4.2. Доступ к серверу СУБД напрямую

# Для доступа Web-броузера к серверу СУБД напрямую могут использоваться как Java-аплеты и программные компоненты ActiveX Controls, так и подключаемые к броузеру специализированные программные модули (plug-ins).

Для использования Java-аплетов по доступу к различным серверам СУБД разра­ботан стандартный интерфейс JDBC (Java DataBase Connectivity) – рисунок 4.

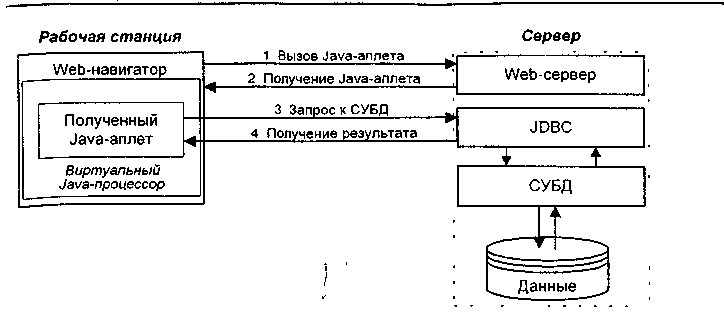


Рисунок 4. Схема доступа к СУБД с помощью Java-аплета через интерфейс JDBC

Доступ Web-навигатора к серверу СУБД с помощью программных компонентов ActiveX Controls (рис. 5) предполагает, как и в случае Java-аплетов, запрос, передачу и выполнение программы на рабочей станции.

В этом случае взаимодействие с серве­ром СУБД должно выполняться через интерфейс ODBC.

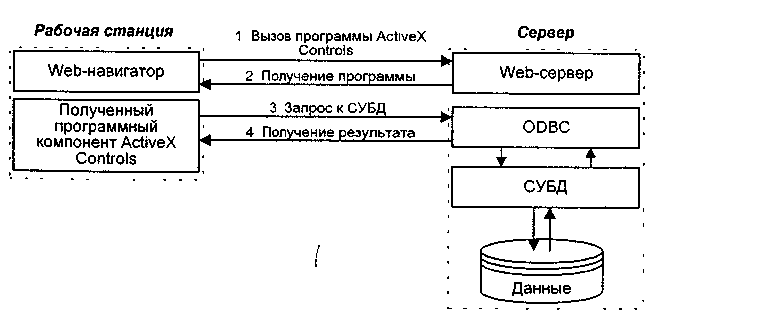


Рисунок 5.Схема доступа к СУБД с помощью программного компонента ActiveX Controls

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 4. Классификация Web ресурсов**

План

1. Основные понятия

2. Типы классификаций веб-ресурсов

* Классификация веб-ресурсов по доступности сервиса
* Классификация веб-ресурсов по физическому расположению
* Классификация веб-ресурсов по типу представления информации
* Классификация веб-ресурсов по признакам применяемой верстки
* Классификация технологий отображения

3. Порталы. Типы порталов

**1. Основные понятия**

Большинство ресурсов Всемирной паутины представляет собой гипертекст.

Гипертекстовые документы, размещаемые во Всемирной паутине, называются **веб-страницами**.

**Веб-страница** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Web page*) — документ или информационный ресурс Всемирной паутины, доступ к которому осуществляется с помощью веб-браузера. Типичная веб-страница представляет собой текстовый файл в формате [HTML](https://ru.wikipedia.org/wiki/HTML),

**Сайт**, или **веб-сайт** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *website*: *web* — «паутина, сеть» и *site* — «место», буквально «место, сегмент, часть в сети»), — одна или несколько логически связанных между собой [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0); также место расположения контента [сервера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).

Обычно сайт в [Интернете](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователем как единое целое. [Веб](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0)-сайты называются так, потому что доступ к ним происходит по протоколу [HTTP](https://ru.wikipedia.org/wiki/HTTP)

Несколько веб-страниц, объединенных общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же веб-сервере, называются **веб-сайтом**

**Веб-портал** — [сайт](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) в [компьютерной сети](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), который предоставляет пользователю различные [интерактивные](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [интернет-сервисы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81), которые работают в рамках этого сайта. Веб-портал может состоять из нескольких сайтов.

**Портал** - многокомпонентная разветвлённая структура, скомпонованная из функционально самодостаточных сайтов самостоятельных организаций или подразделений корпоративной структуры.

Веб-сайты создают с помощью языка разметки гипертекста HTML, каскадных таблиц стилей (CSS) и скриптов (сценариев на языках JavaScript, PHP и др.). HTML применяется для создания (разметки) структуры документа или веб-страницы, таблицы стилей используются для внешнего оформления (стилизации, декорирования) и позиционирования элементов документа, а скрипты - для обеспечения интерактивности и динамичности веб-страниц, а также обработки данных, которыми обмениваются браузер и веб-сервер

**2. Типы классификаций веб-ресурсов**

Существует много различных подходов к классификации веб-ресурсов. Классификация отражает не только тематику, но и технологию ресурса. Универсальной классификации не существует.

В силу разнообразия веб-ресурсов на практике удобно проводить **классификацию** **по конкретному определяющему признаку**, а именно:

* по доступности сервисов;
* по физическому расположению;
* по схеме представления информации, её объёму и категории решаемых задач;
* по технологии отображения;
* по особенностям применяемого макета.

**По доступности сервисов веб-ресурсы делятся на:**

* открытые - все сервисы полностью доступны для любых посетителей и пользователей;
* полуоткрытые - для доступа необходимо зарегистрироваться;
* закрытые - полностью закрытые служебные сайты организаций (в том числе корпоративные сайты), личные сайты частных лиц. Доступ новым пользователям обычно даётся через так называемые инвайты (приглашения).

**По физическому расположению**:

* общедоступные сайты сети Интернет.
* локальные сайты - доступны только в пределах локальной сети.

**Классификация веб-ресурсов по схеме** [**представления информации**](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD), её объёму и категории решаемых задач изображена на рисунке 1, 2.

# Похожее изображение



Рисунок 2.

Информационные ресурсы:

**Тематический сайт** - сайт, предоставляющий специфическую узкотематическую информацию по какой-либо теме.

**Тематический портал** - это очень большой веб-ресурс, который предоставляет исчерпывающую информацию по определённой тематике.

Интернет-представительства:

**Сайт-визитка** - содержит самые общие данные о владельце сайта.

**Представительский сайт** - так иногда называют сайт-визитку с расширенной функциональностью: подробное описание услуг, портфолио, отзывы, форма обратной связи и т.д.

**Корпоративный сайт** - содержит полную информацию о компании-владельце, услугах/продукции, событиях в жизни компании. Отличается от сайта-визитки и представительского сайта полнотой представленной информации, зачастую содержит различные функциональные инструменты для работы с контентом (поиск и фильтры, календари событий, фотогалереи, корпоративные блоги, форумы). Может быть интегрирован с внутренними информационными системами компании-владельца. Может содержать закрытые разделы для тех или иных групп пользователей - сотрудников, дилеров, контрагентов и пр.

**Каталог продукции** - в каталоге присутствует подробная информация товарах/услугах, сертификаты, технические и потребительские данные, отзывы экспертов и т. д., которую невозможно поместить в прайс-лист.

**Интернет-магазин** - сайт с каталогом продукции, с помощью которого клиент может заказать нужные ему товары.

**Промосайт** - сайт о конкретной торговой марке или продукте, на таких сайтах размещается исчерпывающая информация о бренде, различных рекламных акциях.

**Сайт-квест** - веб-ресурс, на котором организовано соревнование по разгадыванию последовательно взаимосвязанных логических загадок.

**Веб-сервис** - сайт, созданный для выполнения каких-либо задач или предоставления услуг в рамках сети WWW.

**Доска объявлений** - ресурс, на котором есть возможность размещения публичного объявления о товарах или услугах, также возможно оставить какую-либо информацию краткого содержания.

**Каталог сайтов** - это ресурс, на котором размещаются сайты и блоги. Каталоги могут способствовать продвижению ресурса, который размещается в каталоге сайтов.

Поисковые сервисы - например, Yahoo!, Google, Bing, Яндекс.

Почтовый сервис - например, Mail.ru и Gmail.

Файлообменный пиринговый сервис - например, Bittorrent.

**Облачное хранилище данных** - например, OneDrive.

**Сервис редактирования данных** - например, Google Docs.

**Фотохостинг** - например, Picnik, ImageShack, Panoramio, Photobucket.

**Видеохостинг** - например, YouTube, Dailymotion.

**Социальные сети** - например, Facebook, ВКонтакте.

**Специализированные социальные сети** - например, MySpace, Flickr.

Классификация веб-ресурсов по особенностям применяемого макета изображена на рисунке 3.



Рисунок 3. Классификация по особенностям применяемого макета

Классификация по технологии отображения сайты различаются:

* Статические - состоящие из статичных HTML страниц, составляющих единое целое. Пользователю выдаются файлы в том виде, в котором они хранятся на сервере.
* Динамические - состоящие из динамичных HTML страниц-шаблонов, информации, скриптов и прочего в виде отдельных файлов.

Содержимое генерируется по запросу специальными скриптами (программами) на основе других данных из любого источника.

# Сайты, созданные с применением Flash-технологий - весь сайт располагается на одной веб-странице, предназначенной исключительно для загрузки Flash-файла, а вся навигация и контент реализованы в самом Flash-ролике

**3. Порталы. Типы порталов**

**Веб-портал** - сайт в компьютерной сети, который предоставляет пользователю различные интерактивные интернет-сервисы, которые работают на этом сайте.

**Интернет-портал, или веб-портал** — многокомпонентная разветвленная структура, скомпонованная из функционально самодостаточных [сайтов](https://tcinet.ru/press-centre/glossary/article.php?ELEMENT_ID=5200) (например, объединение сайтов организаций или подразделений корпоративной структуры, или же различные интерактивные интернет-сервисы, которые работают в рамках одного сайта).

**Концепция веб-порталов** — предоставление максимального количества интернет-сервисов например, [поиска](https://tcinet.ru/press-centre/glossary/article.php?ELEMENT_ID=5204), [электронной почты](https://tcinet.ru/press-centre/glossary/article.php?ELEMENT_ID=5194), ленты новостей и т.п.) в одном месте для привлечения наибольшего числа пользователей.

Они могут помогать пользователям в поиске нужной информации, представляя информацию из различных источников в единообразном виде (используя внешние ссылки).

## Типы порталов – рисунок 4.

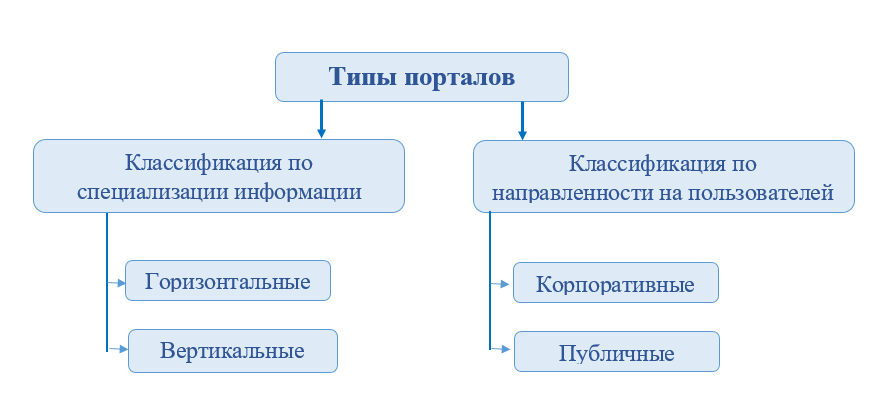


Рисунок 4.

**Классификация по назначению (**по специализации информации**)**

**Горизонтальный портал** (*универсальный*, *общего назначения*, *general portal*) — портал, охватывающий множество тематик, представляющий набор сервисов (обслуживающих, по возможности, все темы) и ориентированный на максимально широкую аудиторию — на максимальный охват её интересов. Наиболее известные горизонтальные порталы: [Yahoo!](https://ru.wikipedia.org/wiki/Yahoo!) (англоязычные); [Mail.ru](https://ru.wikipedia.org/wiki/Mail.Ru) (русскоязычный); [Yandex](https://ru.wikipedia.org/wiki/Yandex)

Такие порталы, как правило, сочетают в себе разнообразные функции, предлагают разноплановый контент и различные сервисы (новостные, финансовые, развлекательные, игровые и т. д.).

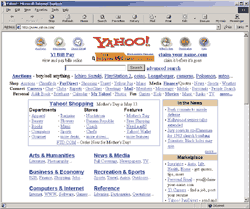
**

Рисунок 5. Типичный пример горизонтального портала

**Вертикальный портал** (*нишевый*, *vertical portal*, *niche portal*) — портал узкой тематической направленности, предоставляющий различные сервисы для пользователей сети по определённым интересам и ориентированный на полный охват определённой тематики или области деятельности. Примерами таких порталов могут служить, например, туристические агентства, предоставляющие услуги по бронированию мест в гостиницах, заказу и доставке билетов, доступу к картам и сведениям об автомобильных маршрутах и т.д., либо порталы типа B2B (business-to-business), позволяющие своим клиентам реализовывать совместные бизнес-операции (например, выбирать поставщиков и осуществлять закупку товаров, проводить аукционы и т.д.).

Если тематика вертикального портала довольно интересная, вокруг него может сложиться [интернет-сообщество](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D1%81%D1%82%D0%B2%D0%BE) (community) — более-менее постоянная группа лиц, которые систематически общаются между собой (например, в чате такого портала) (рисунок 6).

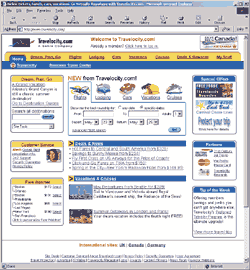


Рисунок 6.

**Классификация по направленности на пользователей**

[Корпоративный портал](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BF%D0%BE%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BE%D1%80%D1%82%D0%B0%D0%BB) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *corporate portal, enterprise portal*) — совокупность информационных систем и [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) предприятия, организации или учреждения, представленных в интернете.

Корпоративный портал предоставляет сотрудникам компании (или её постоянным партнерам) строго определённые права доступа к автоматизированной системе управления (информационной системе подготовки принятия решений, экспертной системе, системе совместной работы, системе управления бизнес-процессами и т. д.).



**Публичный портал** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *public portal*) является доступным для неограниченного круга пользователей.

Публичный портал предоставляет любому посетителю любую информацию и любые сервисы.

Как правило, публичный портал принадлежит какой-либо компании и является частью её бизнеса (например, Mail.Ru). Однако его нельзя отнести к корпоративным сайтам, вследствие чего на таком портале редко можно увидеть данные о самой этой компании.

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 5. Структура Web сайта**

План:

1 Основные элементы веб-страницы

2 Типы позиционирования

3. Логическая и физическая структура сайта

**1 Основные элементы веб-страницы**

Любая web-страница содержит набор стандартных элементов, являющихся обязательными компонентами каждого ресурса Интернета. Безусловно, ассортимент и количество подобных объектов могут варьироваться в зависимости от тематической направленности сайта.

***Основные элементы web-страницы***

Заголовок / Логотип (Шапка)

Поиск

*Рекламный Баннер*

Контент / Содержание (текстовое поле)

Элементы навигации

Информация о разработчике сайта

*Счетчик посещаемости*

**Header -**Заголовки страниц

Заголовки обеспечивают идентификацию сайта и глобальную навигацию с помощью поиска и, возможно, других инструментов.

Заголовки являются наиболее видимым компонентом идентичности сайта.

Первым элементом web-страницы является ее  **Заголовок**. Он может быть выполнен как в текстовом, так и в графическом варианте, однако и в том и в другом случае он должен располагаться в верхней части документа.

**Логотип**  или название сайта располагается на заглавной (первой) странице.

Логотип сайта должен находиться в верхней части страницы, обычно в верхнем левом или правом.  Это объясняется тем, что логотип представляет весь сайт в целом, и, следовательно, он занимает самую верхнюю позицию в логической иерархии сайта (рисунок 1)

# http://24ikt.ru/html/6/pic/clip_image002.jpg

# Рисунок 1.

В визуальной иерархии каждой страницы можно сохранить эту доминирующую позицию двумя способами: либо сделать логотип самым заметным объектом на странице, либо поместить его так, чтоб он выполняли роль рамки для содержания страницы.

# Домашняя ссылка

# Размещение логотипа вашей организации или сайта в верхнем левом углу страницы

# В канонической форме заголовков страниц преобладают горизонтальные полосы ссылок и инструментов, одна или несколько идентификационных графических изображений и реклама на многих сайтах.

Непосредственно под / над заголовком документа, как правило, располагается пространство,  отведенное для размещения рекламного  **баннера**.

Включение баннера именно в верхнюю часть web-страницы в большинстве случаев является обязательным условием регистрации сайта в службах баннерного обмена - системах, рекламирующий созданный вами ресурс в обмен на показ на страницах вашего сайта рекламы других участников баннерообменной сети.

**Глобальная навигация**

Заголовки являются наиболее частым местом для глобальных навигационных ссылок, которые охватывают сайт.

Идеальным вариантом является использование html-списка ссылок в стиле css для горизонтального распределения по разделу заголовка.

Это дает вам:

Удобство использования: глобальные ссылки, где пользователи ожидают их увидеть

Семантическая логика: коллекция глобальных ссылок должна быть помечена как список

Доступность: формат списка ссылок появляется в начале списка кода, где он должен быть

Видимость поиска: коллекция ваших основных навигационных ключевых слов, связанных и находящихся вверху списка кода, идеально подходит для поисковой оптимизации.

**Вкладки** - еще одно широко используемое и понятное соглашение для глобальной навигации. Лучший способ реализовать вкладки - это оформить обычный HTML-список с более сложной обработкой CSS, чтобы сформировать «вкладку» вокруг каждой ссылки.

 Этот тип маркера «вы здесь» необходим для ориентации пользователей на сайте. Вкладки также можно использовать для реализации двухуровневой схемы навигации, в которой вторичный горизонтальный список ссылок отображается под выбранной вкладкой, снова в виде простого html-списка со стилем CSS, чтобы сохранить семантику, доступность и видимость поиска

<https://webstyleguide.com/wsg3/6-page-structure/3-site-design.html>

**Breadcrumbs**

Хлебные крошки (навигационная цепочка, англ. Breadcrumbs) - это элемент навигации по сайту, который представляет собой пользовательский путь к текущей странице.

Хлебные крошки обычно представлены в верхней части страницы, обычно под шапкой сайта:

Главная страница / Раздел / Подраздел /… / Текущая страница.

**Поиск**

Поле поиска на основе заголовка (рисунок 2)



Рисунок 2

**«Оформить заказ корзин», интернет-магазины «тележки»**

**Корзина покупок** помещена в заголовок, потому что именно там большинство людей будут искать

**Боковая панель**

дизайн колонок стал согласованным в основных веб-сайтах контента и продаж

Сканирование столбцов

Разделение поля страницы на функциональные области является фундаментальной характеристикой современного графического дизайна,

Сканирующие столбцы также полезны в качестве местоположений для полей веб-поиска, почтового адреса и контактной информации и других более мелких, но необходимых элементов страницы. Исследования показывают, что левый столбец сканирования под локальными ссылками навигации - это второе место, где большинство пользователей будут искать функции поиска после того, как они посмотрят в правую область заголовка.

Левый или правый столбцы сканирования для навигации?

левый столбец для навигации просто потому, что это самая распространенная практика.

Почтовый адрес и контактная информация

Отображайте свою контактную информацию на видном месте, например в столбце сканирования, на каждой странице.

Реклама и сканирование столбцов

Пользователи часто игнорируют контент, похожий на рекламу, когда видят его в столбце сканирования. Используйте форматы, которые не кричат «реклама!». Никогда не делайте содержимое или навигационную колонку вашего сканирования похожими на обычные рекламные баннеры, иначе пользователи могут никогда не заметить этого.

**Content**

**Текстовое** **поле**   - основная часть документа - участок, где и размещается смысловое наполнение страницы: содержательный информационный текст и иллюстрации (content - Содержание).

Расположение текстового поля зависит в первую очередь от того, каким образом web-дизайнер разместит остальные элементы документа (рисунок 3, 4).

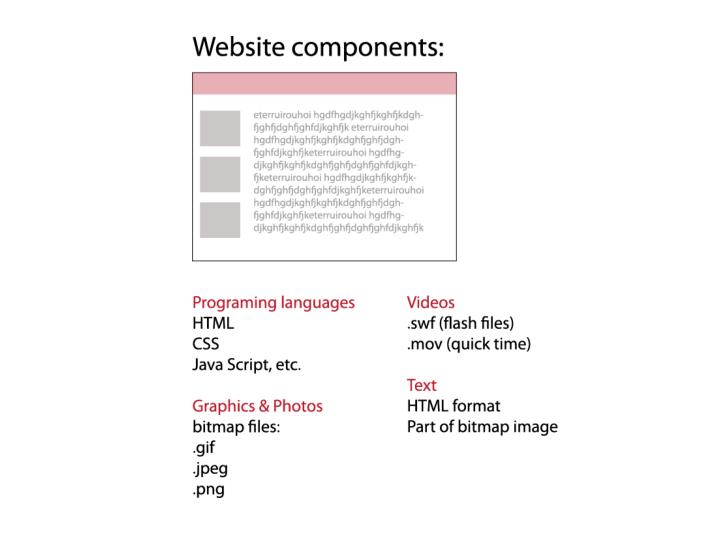


Рисунок 3



Рисунок 4

**Web** **-контент**

Заголовки страниц. Каждая страница нуждается в видимом имени в верхней части.

Навигация. Верхняя часть области содержимого является наиболее распространенным местом для навигации по хлебным крошкам.

Перейти к началу ссылки. Переходные ссылки - хорошее уточнение для длинных страниц.

Пейджинговая навигация. В многостраничных последовательностях удобно иметь простые текстовые ссылки вверху и внизу страницы, чтобы переместить читателя на предыдущую или следующую страницу последовательности. В более длинных последовательностях полезно предоставить информацию, описывающую, где они находятся в серии.

Даты: отображение даты последнего обновления в нижней части области содержимого

**Элементы навигации** - гиперссылки, связывающие данный документ с другими разделами сайта. Элементы навигации могут быть выполнены в виде текстовых строк, графических объектов, то есть кнопок, либо активных компонентов, например Java-апплетов.

Элементы навигации представляют собой те же кнопки, которые умеют реагировать на движения мыши, выполняя при наведении на них курсора какие-либо несложные действия (включение подсветки, создание эффекта "нажатия", изменение формы и т. д.).

Располагать элементы навигации следует таким образом, чтоб они всегда были "на виду", то есть так, чтоб пользователю не приходилось "отматывать" страницу назад, если текстовое поле занимает по высоте несколько физических экранов, после чего подолгу искать ссылки на другие разделы.

# Наиболее устоявшимся подходом является размещение элементов навигации в левой границы страницы и / или верхней.

# Slider

# Слайдер - блок на странице, в котором с установленной периодичностью происходит демонстрация анонсов новостей, статей или изображений

# Footer

# В нижней части документа принято публиковать информацию о разработчике сайта и адресов электронной почты, по которому посетители ресурса могут направит владельцам странички свои отклики, предложения и пожелания (рисунок 5).

# http://24ikt.ru/html/6/pic/clip_image004_0001.jpg

# Рисунок 5

**Нижние колонтитулы**

Нижние колонтитулы страницы в основном касаются ведения домашнего хозяйства и юридических вопросов.

Эти элементы должны быть на странице, но разместите их где-нибудь в стороне:

* Автор страницы или, на крупных корпоративных сайтах, ответственная сторона
* Заявление об авторском праве
* Контактная информация, электронная почта
* Ссылки на связанные сайты или на более крупное предприятие
* Избыточные навигационные ссылки для длинных страниц

Если web-страница является стартовым документом, в нижней ее части также размещают  **счетчик** **посещений**  - небольшой сценарий, вызывающий установленный на сервере CGI-скрипт, который фиксирует каждое открытие документа в браузере пользователей, изменяя значение индикатора счетчика.

# Счетчик посещений устанавливается только на первой странице, вызываемой при обращении к сайту, в остальных документах ресурса он отсутствует. Не рекомендуется также размещать на одной странице несколько разных счетчиков.

# Шаблон (макет) веб страницы - предназначен для размещения элементов на странице

# 2. Типы позиционирования элементов

# Пример компоновки web-страницы с левым позиционирование элементов навигации (рисунок 6)



Рисунок 6

Пример компоновки web-страницы с правым позиционирование элементов навигации (рисунок 7)





Рисунок 7

Логотип в этом случае помещен на один уровень с заголовком документа, а рекламный баннер позиционировано Относительно центра страницы. При таком подходе рекомендуется выдерживать графическое оформление заголовка, логотипа и поля для размещения рекламы в едином цвете и художественном стиле - тогда несимметричность положения данных объектов будет не столь очевидна

# Пример компоновки web-страницы с верхним позиционирование элементов навигации (рисунок 8)

# Картинки по запросу Basic elements of a web page

# http://24ikt.ru/html/6/pic/clip_image002_0002.jpg

# Рисунок 8

# В этом случае все объекты страницы гармонично "вписываются" в заданную ширину невидимой таблицы, при этом подготовка самой таблицы значительно упрощается.

# Единственным недостатком подобного подхода является необходимость продублировать элементы навигации в нижней части документа, поскольку при вертикальной прокрутки страницы они исчезают за верхней границей экрана, и, чтоб добраться до них, пользователю придется "отматывать" экран назад, что, согласитесь, весьма неудобно.

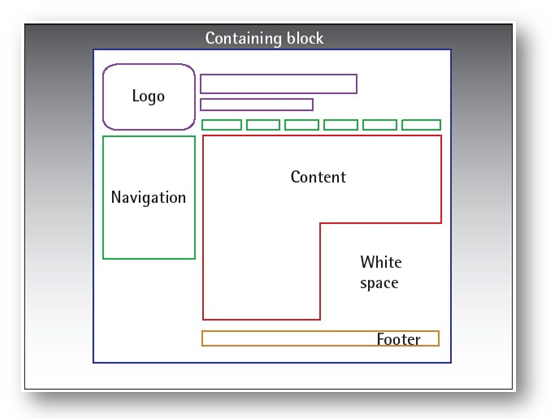
# Пример "смешанной" компоновки web-страницы (рисунок 9)

# 4.gif

# Рисунок 9

В данном примере часть управляющих элементов встроен непосредственно в заголовок. Основной блок элементов навигации позиционировано Относительно левой границы документа и вверху. Текстовое поле разделены на две несимметричные колонки, причем в правой размещены краткие анонсы составляющих ресурс тематических рубрик, включая ссылки на эти разделы (рисунок 10).

# http://24ikt.ru/html/6/pic/clip_image005.gif



# Рисунок 10

**3. Логическая и физическая структура сайта**

Каждый ресурс Интернета, от любительской домашней странички до большого информационного портала, содержит несколько тематических рубрик, соединенных между собой гиперсвязями. Как правило, ссылки на все разделы сайта с краткими анонсами их содержимого приводится на первой, так называемой **стартовой странице**, которой присваивается имя index.htm (.html). Если тематические рубрики содержат собственные подразделы, каждая из них также имеет свою стартовую страницу, называющуюся index.html.

*Примечание.* Такое имя файла рекомендуется присваивать всем стартовым документам сайта, поскольку в противном случае при обращении к какому-либо разделу посредством сокращенного URL без указания названия стартовой страницы (например, http://www.mysite.ru/photos/ вместо [http://www.mysite.ru/photos/ startpage.html](http://www.mysite.ru/photos/startpage.html)) броузер отобразит не саму web-страницу, а перечень хранящихся в данной папке файлов.

Набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса называется **логической структурой сайта**.

**Физическая структура**, напротив, подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой опубликован ваш сайт.

Пример сравнения логической и физической структур одного и того же ресурса Интернета показан на рис. 11.

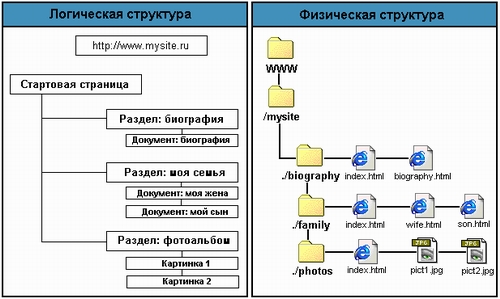


Рисунок 11. Сравнение логической и физической структуры сайта

Логическая и физическая структуры могут не совпадать, поскольку в общем случае физическая структура ресурса разрабатывается, исходя из удобства размещения файлов.

*Совет.* Рекомендуется размещать все графические изображения, являющиеся элементами проекта, в отдельной папке с названием "Images", расположенной в корневой директории сайта. Такой подход позволит обновлять хранящиеся в других тематических разделах документы HTML без переноса графики, использовать одни и те же графические файлы во всех разделах сайта и при необходимости удалять целые директории.

Для того чтобы все гиперссылки на вашей домашней страничке или web-сайте работали корректно, все документы открывались правильно и броузер не выдавал ошибок при обращении к каким-либо разделам ресурса, при создании его физической структуры следует соблюдать несколько простых правил.

*СОВЕТ* Назначайте имена директорий, имена и расширения документов HTML и графических файлов с использованием символов только латинского алфавита и только в строчном регистре.

Старайтесь, чтобы имена созданных вами файлов и директорий не превышали по длине восьми символов.

# Из всего сказанного становится очевидным, что физическая структура сайта скрыта от посетителей вашего ресурса: они могут наблюдать только логическую структуру, причем именно так, как она представлена при помощи элементов навигации.

# Отсюда следует вполне логический вывод: строение системы навигации должно если не полностью повторять, то хотя бы максимально соответствовать разработанной вами логической структуре сайта.

# Домашняя страница

# Домашние страницы имеют четыре основных элемента:

# тождественность

# навигация

# Своевременность или ориентация на содержание

# Инструменты (поиск, каталоги)

# Домашние страницы часто имеют отличительную тему, в которой доминирует один фактор.

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 6. Процесс разработки Web сайта**

*План*

1. Этапы разработки сайта

Этап 1. Исследования и планирование

Шаг 1. Определение цели проекта

Шаг 2. Создание идеи проекта

Шаг 3 Разработка структуры проекта (карта сайта)

Шаг 4. Планирование и управление проектом

Этап 2: Дизайн

Шаг 1. Разработка каркаса проекта

Шаг 2. Визуальный дизайн

Шаг 3. Принципы интерактивного и визуального дизайна

Шаг 4. Тенденции веб-дизайна

Шаг 5. Создание контента

Этап 3. MarkUp (Макет) и разработка сайта

Шаг 1. Макет сайта

Шаг 2. Разработка

Этап 4. Тестирование и запуск, поддержка

2. Поддержка сайта: отзывы пользователей и регулярные обновления

**1. Этапы разработки веб сайта**

Условно процесс создания сайта (web-проекта) можно разделить на 4 этапа:  
**Исследование и** [Планирование](https://habr.com/ru/post/273795/#planning)

[Дизайн](https://habr.com/ru/post/273795/#design)

Верстка и [Разработка](https://habr.com/ru/post/273795/#coding)

Тестирование и запуск, поддержка

<https://www.mediatopia.co.uk/blog/a-detailed-article-explaining-the-web-design-and-development-process>

# Этап 1. Исследование и [Планирование](https://habr.com/ru/post/273795/#planning)

# Шаг 1. Определение цели проекта

Создаваемый Web-узел может быть предназначен для решения одной из следующих задач.

* Привлечь новых потребителей.
* Предоставить существующим потребителям новые средства приобретения товаров.
* Сделать каталоги товаров и услуг доступными в Internet.
* Предоставить широкой публике информацию о компании.

# Чтобы выяснить цели Web-сайта нужно найти ответы на вопросы у заказчика:

* Каково основное назначение Web узла?
* Кто ваши конкуренты? достоинства и недостатки их узлов?

Проведите анализ сайтов конкурентов: структура, разделы, дизайн. Какие функции реализованы в сайте конкурента?

Обладает ли узел какими-либо особенностями, отличающими его от других Web –узлов

* Какие кратковременные и долговременные цели преследует создание нового узла?
* Какую информацию о вашем бизнесе вы хотите сделать доступной широкой публике посредством Web -узла?
* Какие продукты или услуги вы предлагаете потребителям?
* С какими другими узлами вы хотели бы связать ваш узел посредством гипертек­стовых ссылок?
* Какой объем информации должен быть представлен на Web -узле?
* Есть ли у вас зарегистрированный товарный знак? Надо ли помещать его на Web -узел?
* Какие ресурсы имеются в наличии в настоящее время (например, фотографии, текстовые файлы, логотипы и т.д.)?
* Какие основные разделы должен содержать Web -узел?
* Как часто вы предполагаете обновлять содержимое Web -узла?
* Каков предполагаемый состав посетителей Web -узла?

# Структура и Дизайн Web- узла зависит от целевой аудитории.

# Хорошим считается тот Web-узел, содержимое которого поймут большинство пользователей и внешний вид которого понравится им.

**Целевая аудитория** — один из основных факторов, определяющих соотноше­ние текста, изображений, мультимедиа-данных и интерактивных элементов в составе Web - узла

# После того как вы узнали потребности заказчика и определили масштаб проекта, вы должны составить график выполнения проекта и бюджет.

**Шаг 2. Создание идеи проекта**

# На данном этапе нам необходимо определиться с тематикой проекта (сайта, сервиса).

# Далее, в соответствии с выбранной темой, необходимо выполнить следующие шаги:

# 1) Определение контента:

# - соберите соответствующие материалы: текстовые, графические.

# - соберите текст, логотипы, фотографии, иллюстрации, аудиофай­лы и прочие данные.

# - определите какая информация наиболее важна и как ее следует разместить на Web -узле.

# - Распределите данные по темам.

# - Продумайте способ представления данных.

# - определите какую часть содержимого узла должен состав­лять текст и как много изображений необходимо представить на узле.

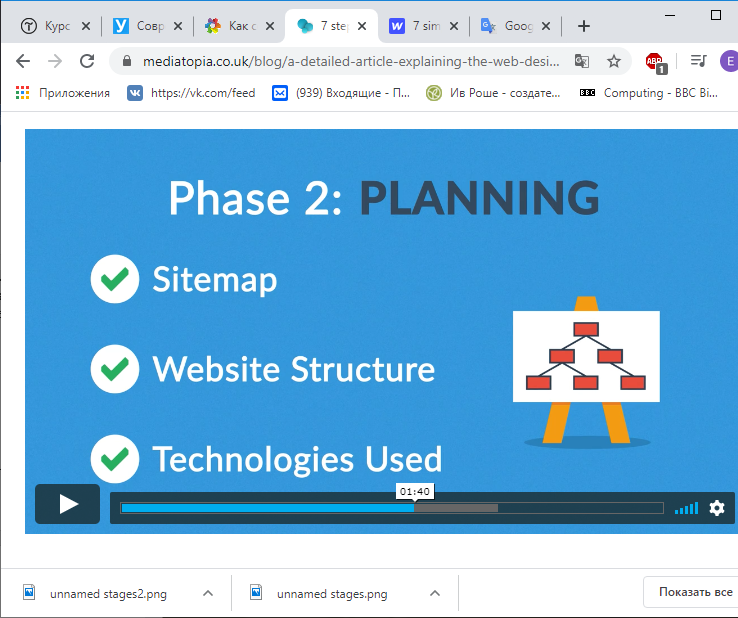
# - определите нужны ли до­полнительные материалы, графические данные, таблицы или интерактивные компо­ненты?

# 2) Определение размера сайта

# - Оцените объем информации и подумайте о том, как наилучшим образом распре­делить ее в пределах узла.

# - Объем информации определяет количество Web -страниц, их внешний вид и средства навигации.

# - Подумайте о том, как эффективно представить данные на узле.



**Шаг 3 Разработка структуры проекта**

# 3) разработка информационной архитектуры (логической структуры) сайта

# Информационная архитектура позволяет организовать содержимое Web-yзлa и опре­деляет, каким образом пользователь получает доступ к информации.

# Процесс создания информационной архитектуры начинается с разделения информации на категории и определения приоритетов данных. Чтобы разделить данные на категории, оцените, насколько важна информация, которая должна быть представлена.

# В идеале Web-узел должен иметь иерархическую структуру. На верхних уровнях иерархии должны быть представлены самые общие категории. Подкатегории, распо­ложенные на следующих за ними уровнях, соответствуют подтемам, и на нижних уровнях располагаются конкретные вопросы, интересующие пользователей (рисунок 1).

# Иерархическая структура

# Линейная структура

# Гибридная структура

# Рисунок 1

# Подобное расположение информации упрощает не только работу пользователя с Web-узлом, но и сам процесс создания Web-узла.

# Желательно разделить все данные на 5—7 категорий.

# Процесс разделения на категории принято называть *формированием блоков* (chunking). Число категорий ограничено тем, что интерфейс, представляющий более 7 категорий, создать трудно, и такой сайт будет перегружен информацией.

# Создание карты сайта

# Пример карты сайта –рисунок 2.

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\sitemapxb-rus.png

# Рисунок 2

# Шаг 4. Планирование и управление проектом.

# - Разработка требований к проекту,

# - Определение состава группы разработчиков проекта (рисунок 3)

# Похожее изображение

# Рисунок 3

# - определение бюджета

# - разработка графика выполнения работ и согласования его с заказчиком

# - Формирование физической структуры сайта

# Используется специальное ПО – MS Project Management

## Этап 2. Дизайн

## Шаг 1. Разработка макета проекта

# После определения информационной структуры проекта нужно составить макет проекта.

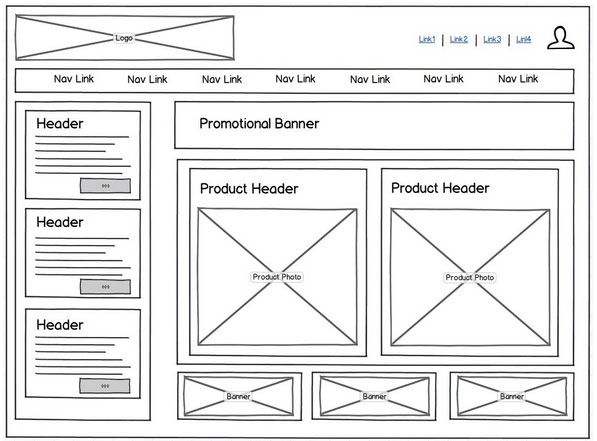
# Макет (wireframe или mock-up).

# Макет представляет из себя визуальное представление будущего интерфейса сайта.

# Но, в отличие, от шаблона, макет не содержит элементов дизайна, таких как цвет, логотипы, и т.п.

# Макет  описывает, какие элементы будут помещены на страницу и как они будут расположены.

# Макет представляет собой своего рода набросок будущего сайта. Вы можете использовать один из доступных онлайн-сервисов для создания макетов – рисунок 4.



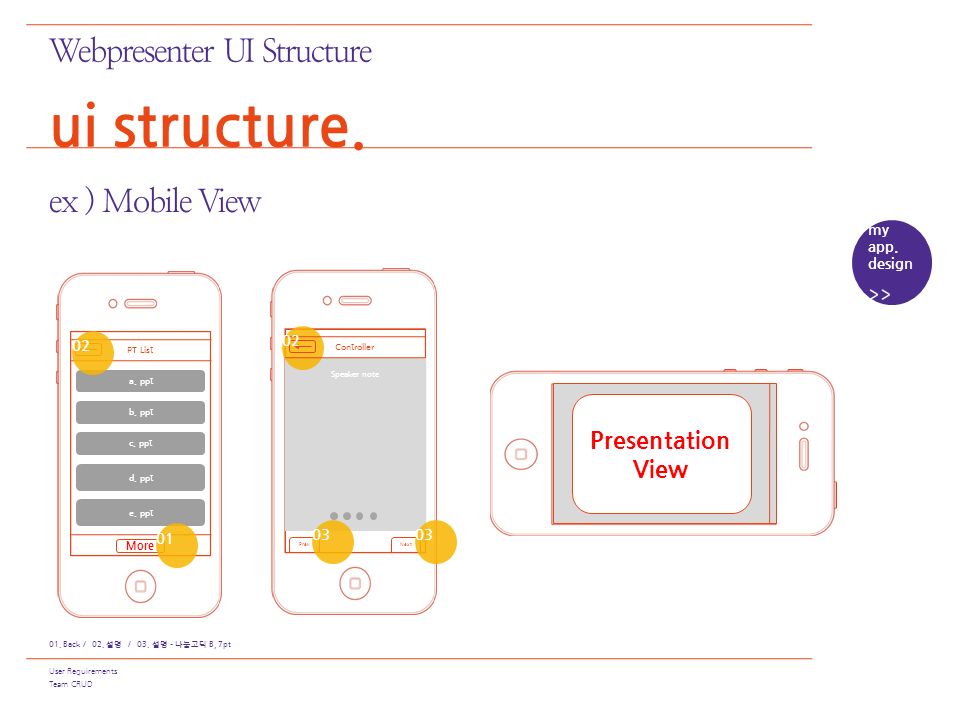


Рисунок 4

**Инструменты для создания макетов**

Gomockingbird

Balsamiq Mockups

Axure

Gliffy

iPhone mockup

InDesign

Photoshop

Dreamweaver

Процесс создания дизайна

<https://www.youtube.com/watch?v=r34XQDoZ0y0>

### **Макеты веб-страниц**

Среди всего многообразия составления макета веб-страницы можно выделить четыре наиболее распространённых:

Навигация в левом столбце – рисунок 5

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\7.jpg

Рисунок 5

# Навигация в правом столбце – рисунок 6

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\8.jpg

Рисунок 6

# Навигация в трёх столбцах - рисунок 7

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\9.jpg

Рисунок 7

# Горизонтальная навигация. На данном этапе сайты с таким типом навигации составляют большинство. Удобство такого подхода легко объяснить тем, что в данном случае у нас остаётся больше пространства для контента, составляющего наш сайт (- рисунок 8)

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\10.jpg

# Рисунок 8

### **Основные элементы страницы**

Основными элементами страницы являются: содержащий блок (wrapper, container), логотип, навигация, контент, футер (нижний колонтитул), свободное пространство (по сути свободное пространство — это не элемент дизайна, но понятие, помня о котором при составлении макета страницы, наш проект не будет выглядеть как нагромождение блоков) рисунок 9.

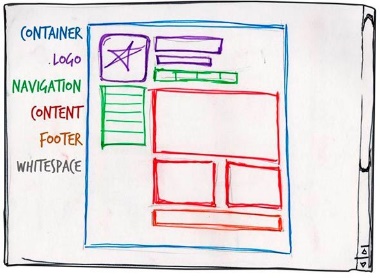


Рисунок 9

#### **Содержащий блок (контейнер)**

Роль контейнера на странице может выполнять непосредственно элемент body или же div. Ширина содержащего блока может быть резиновой (fluid), а может быть фиксированной (fixed).

**Логотип**

Текстовая или графическая составляющая проекта и выделяющая его среди других.

Логотип чаще всего располагается в верхнем левом углу страницы или же посередине (в зависимости от идеи, макета).

#### **Навигация**

Основная навигационная панель содержит ссылки на основные разделы сайта. Навигационная панель часто располагается в верхней части страницы

#### **Контент**

Контент – это основная составляющая веб-страницы.

#### **Нижний колонтитул (footer)**

Данный элемент располагается внизу страницы и обычно содержит информацию о правообладателе, контактные и юридические данные, ссылки на основные разделы сайта (зачастую дублирует основную навигацию), ссылки на социальные сети, форму обратной связи и пр.

Примеры сайтов с примерами других работ:

<http://unmatchedstyle.com/gallery>

* [http://cssdrive.com](http://cssdrive.com/)
* <http://foundation.zurb.com/templates.html>
* <http://zurb.com/patterntap>

# Шаг 2. Визуальный Дизайн

После создания макета проекта можно переходить непосредственно к созданию дизайн-макета. На этом этапе создается шаблон страницы (page layout).

Основное назначение шаблона — визуализировать структуру страницы, ее содержимое, а также отобразить основной функционал. В отличие от макета, в шаблоне используются элементы дизайна. Шаблон содержит цвета, логотипы и изображения.

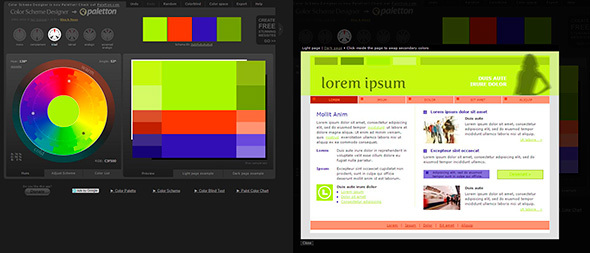
**1) Определение цветовой гаммы проекта**.

Один из способов определения основного цвета в проекте – это составление mood board (доска настроения).

Для этого необходимо выписать себе все синонимы, связанные с темой проекта, а затем каждый синоним набрать в поиске по картинкам Google или Yandex. На основе найденных изображений выписать себе цвета, которые чаще всего встречаются на них (каких цветов больше). Найденные цвета будут составлять визуальное восприятие нашего проекта и вызывать у пользователя соответствующие чувства.

Для работы с выбранным цветом и составлением палитры цветов сайта можно использовать следующие инструменты:

**Color Scheme Designer 3** (<http://colorschemedesigner.com/csd-3.5/>). Помимо выбора цветовой схемы данный сервис позволяет посмотреть пример того, как выбранные цвета будут смотреться на сайте (рисунок 10)



# Рисунок 10

# Adobe Color CC (<https://color.adobe.com/ru/>). Данный ресурс, в отличие от Color Scheme Designer 3, позволяет создавать палитры ещё и на основе загруженных изображений (которые, например, могли появиться у нас на этапе составления mood board). Так же данный сервис обладает большим архивом палитр других пользователей (рисунок 11).

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\13.jpg

# Рисунок 11

# Важно отметить, что при подборе цветов для палитры всегда стоит выбирать как минимум 2 контрастирующих цвета. Достижение нужного контраста между цветами – необходимое условие для того, чтобы у вас получился хороший интерактивный дизайн.

**Шаг 3.** **Интерактивные и визуальные принципы дизайна**

## Элементы Call to Action

Понятие призыва к действию относится к интерактивным элементам сайта: кнопки, баннеры и пр. Данные элементы оформляются таким образом, чтобы пользователю хотелось на них нажать. Например, это может быть кнопка с призывом к действию (Нажми, Купи, Сэкономь), яркий баннер с заманчивым предложением, яркой картинкой и пр. Данное понятие хорошо вписывается в принцип AIDA (Attraction Interest Desire Action).  
**AIDA:**

* Привлечение внимания
* Интерес
* Желание
* Действие

Данное понятие применяется чаще при дизайне главных страниц, страниц акций и пр., где необходимо подтолкнуть пользователя к тому или иному действию: подписка, покупка и пр.

# 2) Визуальные принципы просмотра страницы

# - схема просмотра страницы (наиболее, естественный путь движения глаз по странице),

# - визуальные направляющие.

# Схемы просмотра страницы

# Z-схема просмотра страницы.

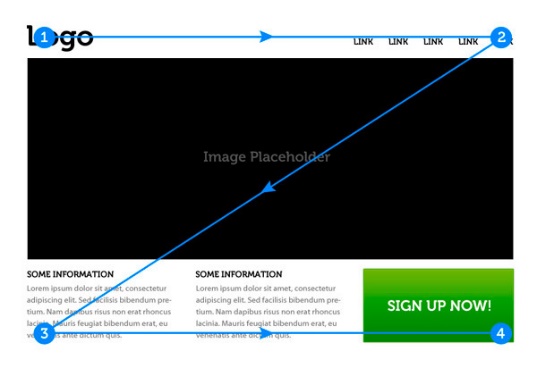
В соответствии Z-схемой, элементы страницы обычно располагают следующим образом: логотип слева вверху, меню справа вверху, информационные блоки, картинки слева внизу, кнопка с призывом к действию справа внизу

Люди чаще всего сканируют по Z-паттерну те страницы, на которых текст не является ключевым элементом. (Когда текст играет главную роль — как, например, в статьях или в поисковой выдаче — лучше использовать [F-паттерн](https://uxplanet.org/f-shaped-pattern-for-reading-content-80af79cd3394)).

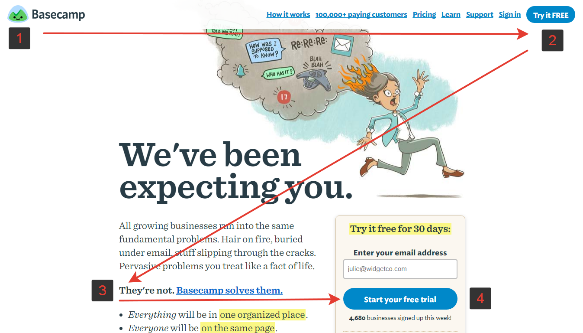
Таким образом, **Z-паттерн хорошо подходит для простых дизайнов, где минимум текста и несколько элементов, на которые нужно обратить внимание.**

Например, Z-паттерн можно применить при создании минималистичного лендинга, построенного вокруг одного-двух ключевых элементов (рисунок 12).

# https://miro.medium.com/max/1344/1*Wi60qIw6LydjM1RaB6x6lA.png



# https://miro.medium.com/max/1350/1*EMo8FtityuhRePU3tXX2zw.png



# Рисунок 12.

Рецепт создания Z-лейаута достаточно прост: *постройте на странице букву “Z”.*

В идеале, люди должны сначала увидеть самую важную информацию, а потом — чуть менее важную. Поэтому ключевые элементы нужно разместить на траектории сканирования и в нужном порядке.

*Очень важно создать поток*

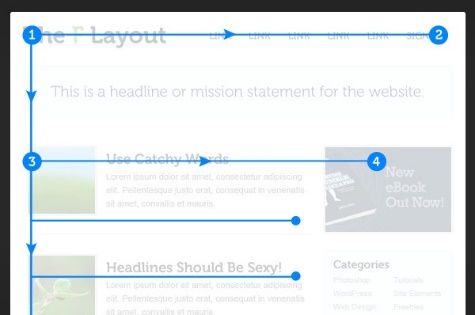
Суть потока в том, чтобы управлять взглядом пользователя, задавать траекторию его движения. Мы создаем поток за счет визуального веса и направления взгляда. Вот несколько практических приемов, о которых следует помнить при создании потока:

* **Точка №1.**Начальная точка пути пользователя. Это лучшее место для вашего логотипа.
* **Точка №2.** Здесь нужно разместить элементы, на которые вы хотите обратить внимание в первую очередь. Поскольку взгляд естественным образом скользит по Z-паттерну, важно разместить в конце верхней горизонтали второстепенный “призыв к действию”. Точке №2 стоит добавить визуального веса (пусть там будет цветная кнопка или другой яркий элемент) — таким образом вы задаете траекторию движения по вашему Z-паттерну.
* **Центральная часть страницы.**Фокус в том, чтобы разместить здесь интересный контент, который, в то же время, будет направлять взгляд пользователя дальше по Z-паттерну — к нижней горизонтали. К примеру, можно разместить между верхней и нижней горизонталью крупное изображение — оно будет визуально отделять верх от низа, при этом направляя взгляд пользователя по Z-паттерну.
* **Точка №3.** Задача точки №3 — отправить пользователя на точку №4, где располагается главный призыв к действию. К примеру, если у вас продуктовая страница, то в точке №3 можно разместить текст, который побуждает пользователей купить еще до того, как они увидят кнопку “Купить”. Так что в точке №3 можно размещать преимущества продукта или другую полезную информацию.
* **Точка №4.** Финишная черта. Все элементы, расположенные на горизонтали между точками 3 и 4, должны направлять взгляд в правый угол. Точка №4 — это идеальное место для главного призыва к действию (рисунок 14).

# https://miro.medium.com/max/1347/1*gSXy2vu8lzDbUvcrEScY9g.png

# F-паттерн

F-паттерн хорошо описывает поведение пользователей только в случае с текстами или контентом, размещенным по монотонной сетке (рисунок 15).



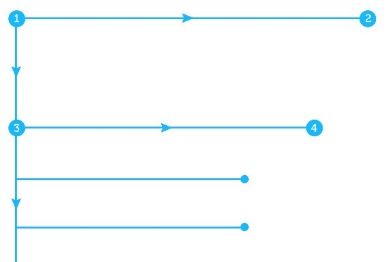


Рисунок 15

**Советы**:

• Придумайте цепляющий заголовок, который отражает содержание статьи.

• Разместите важную информацию в первых двух абзацах текста.

• Менее важную информацию оформляйте в виде коротких абзацев, подзаголовков, списков вдоль основной оси F – рисунок16, 17.

# https://netology.ru/ckfinder/userfiles/images/2%20fokus.png

# https://miro.medium.com/max/785/1*At90KhGJuaZugc1fnidL-w.jpeg

# Рисунок 16

# https://miro.medium.com/max/1344/1*Mb9UjeysdkRdtbdV0zTSMw.png

# Рисунок 17

### **Визуальные направляющие**

# Визуальными направляющими называют декоративные элементы страницы, которые перенаправляют взгляд пользователя на те или иные элементы дизайна, формы, кнопки и пр.

# В качестве визуальных направляющих могут выступать стрелки, направление взгляда человека на изображении, направление указательного пальца, в общем всё, что может как-то указывать в ту или иную сторону (рисунок 18). На первом изображении взгляд невольно следует за указательным пальцем мужчины, а его прямой взгляд невольно привлекает внимание к себе при первом взгляде на страницу.

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\16.jpg

# Рисунок 18

# Тот же эффект продемонстрирован на втором изображении с наложенной тепловой картой: в первом случае нас привлекает в основном прямой взгляд ребенка (прямо в глаза), во втором случае направление взгляда ребенка и его поза невольно обращают взгляд пользователя на блок справа (рисунок 19).

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\17.jpg

# Рисунок 19

# Шаг 4. Тренды веб-дизайна

# При разработке визуального дизайна сайта необходимо учитывать современные тенденции веб-дизайна (*Тренды веб-дизайна рассмотрим в следующей лекции*)

# Шаг 5. Создание контента

# - Написание нового контента. Эта стадия включает в себя также создание привлекательных и броских заголовков, написание и редактирование текста, компиляция существующих текстов и т.д.

# Как правило, заказчик предоставляет контент, уже готовый к размещению на сайте. Важно, чтобы весь контент был подготовлен до или во время стадии разработки.

# - Подготовка контента к переносу на сайт

# - SEO -оптимизация

### **Этап 3. Верстка и разработка сайта**

**Шаг 1. Верстка сайта**

Верстка сайта происходит в соответствии с разработанной картой сайта.

Выбирается требуемый тип верстки сайта.

**Типы верстки сайта**

### **Резиновый и фиксированный макет**

#### **Фиксированный макет**

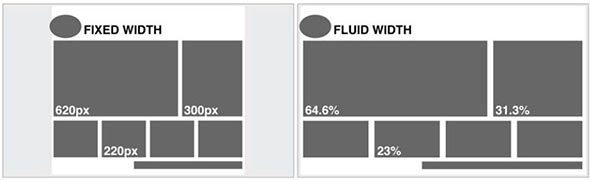
Фиксированный макет подразумевает под собой, что в независимости от разрешения экрана пользователя ваш сайт всегда будет занимать одинаковую ширину (рисунок 20).  
****

Рисунок 20

**Резиновый макет**

# «Резиновый» макет подразумевает, что страница сайта будет стараться занять всё доступное ей пространство на экране пользователя, подстраиваясь под разрешение.

# В настоящее время выделяют:

# **отзывчивый веб-дизайн (Responsive Web Design, RWD)**  - понятие укладывается в концепцию «резинового» и означает, что при изменении размера экрана ваш сайт подстраивается под него,

# **Адаптивный веб-дизайн (Adaptive Web Design, AWD** означает, что при разработке вы определяете основные разрешения (размеры экрана), под которые будет подстраиваться (адаптироваться) ваш контент.

# В обоих случаях следует разрабатывать не один, а несколько макетов, которые будут соответствовать разным разрешениям экрана. Часто создаётся 3 макета под разрешения iPhone (Android Phone), iPad (Android Tablet) и Desktop.

### **Модульная (координатная) сетка**

# Модульная сетка подразумевает под собой разделение страницы на отдельные колонки по вертикали и выстраивание контента, при разработке дизайн макета, именно по этой сетке.

# Наиболее популярной системой является модульная сетка 960 Grid System ([http://960.gs](http://960.gs/)), которая максимально делит страницу на 12, 16 и 24 колонки.

# Максимум в ширине сетка имеет 960 пикселей. Данное решение основано на том, что большинство современных мониторов, *на момент создания сетки*, имели разрешение не меньше 1024 на 768 пикселей (рисунок 21).

# Создание макета на основе данной сетки, в дальнейшем, поможет ускорить процесс разработки (вёрстки).

# F:\2020 Лекции\Web Tehcnology 2020\6_7 Theme Process development site\1\5.jpg

# Рисунок 21

# Так же стоит отметить, что при разработке «резинового» макета страницы существует понятие максимальной ширины.

# Данное утверждение основывается на удобстве восприятия информации. Если предположить, что наш сайт не имеет максимального значения по ширине, то на больших мониторах информация будет сильно растягиваться и её неудобно будет читать. Чаще всего ограничиваются шириной в 1280 пикселей.

# Благодаря модульной сетке блоки контента и элементы будут располагаться на определённом расстоянии друг от друга, будут иметь удобоваримую ширину, что в дальнейшем визуально будет приятно пользователю и не будет вызывать у него какие-либо неудобства в восприятии сайта.

# Модульная сетка – это некая визуальная абстракция, визуальное деление страницы на равные по ширине столбцы с равными отступами между ними.

# Визуализировать данную модель можно посредствам направляющих или отдельного слоя, на котором будут изображены эти столбцы.

# Шаг 2. Разработка

# Создание страницы делается поэтапно:

# HTML-верстка (HTML-код)

# добавляются стили CSS,

# а после, если необходимо, пишутся скрипты (JS),

# добавляются необходимые плагины и библиотеки.

# или

# использование систем упарвления контентом CMS (WordPress, Drupal, Joomla, Bitrix)

# использование конструкторов сайтов

**Этап 4. Тестирование и запуск**

Тестирование:

- Каждая ссылка должна быть проверена, каждая форма и каждый скрипт должны быть протестированы.

- Текст должен быть проверен программой проверки орфографии для выявления возможных опечаток и ошибок.

- Валидаторы кода используются для того, чтобы быть уверенным, что созданный на предыдущем этапе код полностью соответствует современным веб-стандартам.

Оnline средства:

* Для проверки html: [https://validator.w3.org](https://validator.w3.org/)/
* Для проверки CSS: <http://jigsaw.w3.org/css-validator/>
* Для проверки JS: <http://www.jslint.com/>

После тестирования сайт может быть загружен на сервер.

Обычно для этого используется FTP-клиент. После загрузки сайта на сервер, необходимо провести еще один тест для того, чтобы быть уверенным, что во время загрузки не произошло непредвиденных ошибок и все файлы целы и невредимы.

### **2. Поддержка сайта: отзывы пользователей и регулярные обновления**

Важно быть уверенным в том, что все работает, как и было запланировано, а пользователи удовлетворены конечным продуктом.

Нужно также быть готовым быстро вносить изменения, если это будет необходимо.

Система отзывов позволит вам выявлять возникшие проблемы, с которыми сталкиваются посетители сайта. Самой критичной задачей в подобных случаях будет решение возникших проблем настолько быстро, насколько это возможно. В противном случае, ваши пользователи скорее предпочтут другой ресурс, чем будут мириться с неудобствами.

Также не следует забывать о регулярном обновлении CMS. Регулярные обновления избавят вас от ошибок и проблем с безопасностью.

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 7 Web дизайн**

План:

1. Веб-дизайн (Визуальный дизайн)

2. Принципы веб-дизайна

3. Топ-20 тенденций веб-дизайна 2020 года

4. Способы сделать сайт заметным и удобным для посетителей

# 1. Web дизайн. Web Design (Visual Design)

После создания макета проекта можно переходить непосредственно к созданию дизайн-макета. На этом этапе создается шаблон страницы (***page layout***).

Основное назначение шаблона — визуализировать структуру страницы, ее содержимое, а также отобразить основной функционал. В отличие от макета, в шаблоне используются элементы дизайна. Шаблон содержит цвета, логотипы и изображения.

1. Определение цветовой гаммы проекта.
2. Выбор тренда WеЬ-дизайна
3. Определение/ выбор ключевых особенностей «фишек» проекта

2.  Принципы WеЬ-дизайна

Хороший дизайн является результатом тщательного изучения особенностей поставленной задачи и следования основным правилам.

Принципы выбора элементов, композиции и использования цвета в основном совпа­дают с принципами, используемыми при создании изображений на традиционных носителях.

Для достижения хороших результатов при разработке Web-узла необходи­мо придерживаться следующих принципов.

* Удобство восприятия. Текст, изображения и условные знаки, используемые на Web-сайте, должны быть понятны всем пользователям.
* Согласованность. Все элементы должны сочетаться друг с другом так, чтобы Web-сайт был выдержан в едином стиле.

Необходимо правильно выбирать на­чертание и размеры шрифтов для заголовков различного уровня, использовать гармонирующие друг с другом цвета, а графические изображения должны есте­ственным образом дополнять текстовую информацию.

# **Использование Шрифтов.** Шрифты — это один из элементов графического дизайна. Существуют два способа отображения шрифтов в броузере:

# - использование стандартных шрифтов

- разработка оригинальных рисованных шрифтов (т.е. текст в виде изображения), например, помощью графической программы Adobe Photoshop. Такой способ обеспечивает индивидуальность, полный контроль над шрифтами; внешний вид символов сохраняется в любом броузере

1. **Используйте меньше шрифтов.** В идеале 1, максимум 2-3. Главное, чтобы в тексте была гармония.
2. **Забудьте о центрировании.** Не нужно выравнивать весь текст по середине. Вспомните о сетках и направляющих.
3. **Соблюдайте баланс декоративных элементов.** Завитки, кружочки выглядят необычно, но надо использовать их с умом.
4. **Следите за размерами.** Некоторые дизайнеры расставляют акценты с помощью увеличения

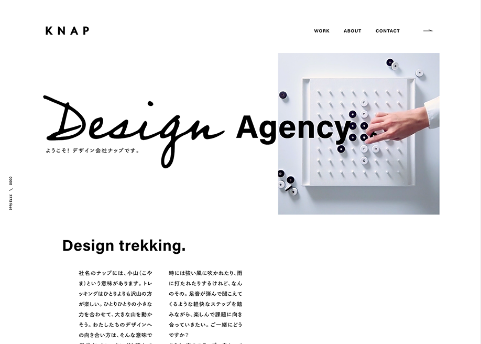


Рисунок 1

* **Простота.** Не следует перегружать Web-cтpaницы второстепенной информацией. Со­держимое необходимо представлять небольшими порциями, удобными для воспри­ятия. На Web -страницах должны оставаться пустые места; **незаполненное пространство -**  «**white space**» - благодаря этому глаза пользователя меньше устают.

Вместо того чтобы заполнять все доступное простран­ство графическими данными, ограничьтесь несколькими изображениями, которые наилучшим образом иллюстрируют текст, представленный в документе.

* **Структура.** Каждая Web-страница и весь Web-сайт должны иметь иерархическую структуру, обеспечи­вающую выделение наиболее важных элементов, группировку сходных компо­нентов. Информация должна распределяться по категориям и подкатегориям

# Основная задача, стоящая перед разработчиком, — создать Web -узел так, чтобы он воспринимался не как набор различных частей, а как единое це­лое. Объекты, выполняющие сходные функции, необходимо объединять вместе, и для представления их использовать один и тот же стиль.

Метафоры и темы

Метафора — это оборот речи, включающий скрытое уподобление, образное сбли­жение слов на базе их переносного значения.

**В Web-дизайне метафорой называется символ, обозначающий некоторое понятие или действие.**

Метафоры могут быть лите­ральными или абстрактными.

Хорошая метафора направляет действия пользователя и способствует объединению материала, представленного на Web-узле.

Метафоры должны быть знакомы пользователям и соответствовать выражаемым понятиям.

Цель использования метафоры — поддерживать центральную тему Web-узла или группу документов, содержащихся на нем.

Выбор и создание метафоры — непростая задача, так как метафоры уместны не на всех Web-узлах.

Метафора хо­рошо работает тогда, когда она представляет собой общепринятый символ, на­пример, изображение мусорной корзины в Macintosh или “корзинки” покупателя в интерактивном магазине.

Не всегда метафору можно получить простой заменой текста пиктограммой. Хорошая метафора помогает разместить информацию именно там, где пользова­тель ожидает найти ее. Плохая метафора заставляет пользователя тратить допол­нительное время для того, чтобы понять, что же именно он может сделать в дан­ной ситуации. Примером хорошей метафоры является витрина магазина, которая

отображается на узле, поддерживающем интерактивный магазин. Щелкая мышью на продуктах, выставленных в витрине, пользователь может выяснить их характе­ристики, а щелкнув на вывеске или эмблеме магазина, пользователь может полу­чить информацию о компании.

Как правило, Web-узлы, содержащие метафоры, более привлекательны для пользо­вателей, чем те, на которых используются только текстовые ссылки. В то же время избегайте использования слишком большого количества метафор или метафор, плохо соответствующих понятиям, которые они представляют; это затрудняет работу пользо­вателей. Если вы применяете в качестве метафоры изображение, позаботьтесь о том, чтобы оно имело привлекательный вид.

***Тема —* это визуальное или понятийное представление времени, места, стиля, среды и т.д.**

Тема увеличивает применимость узла, в особенности если она используется на всем узле и способствует согласованию его элементов.

Например, на Web-узле, посвященном продаже шоколадных изделий, может быть использована тема кондитерского магазина, в частности, навигацион­ные кнопки могут иметь вид леденцов.

**Навигационная схема**

Основой навигационной схемы может служить информационная архитектура – карта сайта. Навигационная схема представляет собой реальный план перехода от одной страницы к другой.

**Основные правила создания навигационной схемы:**

* Минимизируйте число гипертекстовых ссылок на странице (количество ссылок не должно быть больше десяти). Навигационные элементы не должны перегру­жать пользователя информацией и создавать для него проблему выбора.
* Размещайте гиперссылки на разных Web-страницах в одной и той же позиции.
* Обеспечивайте согласованность использования навигационных элементов.

Одинаковое расположение ссылок, выполненных в виде изображений, и соблюдение определенной системы в размещении текстовых ссылок уп­рощает работу с Web-узлом. Так, например, если основные ссылки располагаются в специальной области в левой части экрана, проблема их поиска не возникает. Систе­ма должна соблюдаться не только в размещении ссылок, но и в действиях с ними. Например, пользователь быстро определяет, что после щелчка на “корзинке” покупа­теля на экране отображается ее содержимое. Если на другой странице то же действие пользователя приведет к выводу информации о порядке приобретения товаров в инте­рактивном магазине, эффективность работы на узле снизится.

**Разработка цветовой схемы:** Цвета, выбранные для Web-узла, должны соответствовать образу компании-заказчика и способствовать восприятию идеи (message) — основной информации, представленной на Web-узле.

* **Ис­пользуйте ограниченную палитру гармонирующих друг с другом цветов для фона, текста, элементов**
* Использование контрастов для текста, изображений. Контрасты помогают пользователю воспринимать не­обходимые ему элементы.
* Использование акцентов используются для привлечения внимания к требуемым компонентам: блокам текста, схемам и т.д. с помощью цвета, шрифтов, размеров, композиции

Если на Web -странице черный текст отображается на белом фоне, читать его удобно. Использование различных шрифтов помогает пользователю отделять заголовки разделов от обычного текста

**3. Топ-20 тенденций веб-дизайна 2020 года**

## 1. СОЧЕТАНИЕ ярких цветов

**Сейчас модно использовать сочные и яркие цветовые схемы.** Главное – уместное использование и гармоничное соединение оттенков, применение дополнительных техник для их поиска.



Рисунок 2

## 2. ТЕХНИКА ДУПЛЕКС

Дизайн в рамках двух цветов и их полутонов смотрится стильно и в целом весьма достойно. Новомодная технология уже завоевала популярность среди ведущих веб-дизайнеров.

Такой **инновационный прием часто внедряют при необходимости ребрендинга** в сжатые сроки, оставляя в целом концепцию и структуру сайта без изменений.

Такое незначительное дополнение поможет обновить сайт буквально до неузнаваемости.



Рисунок 3

## 3. МЯГКИЕ МНОГОТОННЫЕ ГРАДИЕНТЫ

Градиент: Мягкие переходы насыщенных или приглушенных тонов – настоящий тренд текущего года.

**Самые стильные – двухцветные градиентные переходы, состоящие из гармоничных оттенков.**

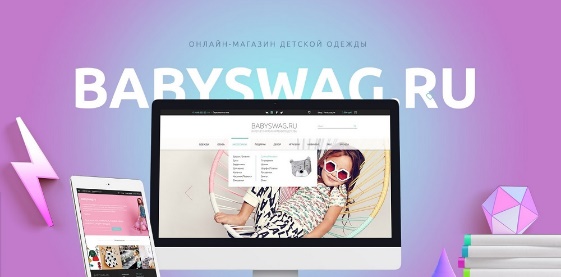


Рисунок 4

## 4. ТЕКСТУРИРОВАННЫЕ ДЕТАЛИ

Чтобы сделать веб-дизайн интересным, можно использовать дополнительно микс текстур. **Металл и дерево – самые популярные варианты.**

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\2\444444456709889729835634.jpg

Рисунок 5

## 5. МИНИМАЛИЗМ

Концепция простоты позволяет не отвлекать пользователя от главного – изучения основной информации.

**Лаконичные дизайнерские решения применяют для фоновых и дополнительных элементов.**

Белый фон продолжает выглядеть уместно в большинстве проектов.

Монохром (когда фон и основной объект по цвету идентичны, различаются при помощи тени) – также достаточно результативный прием.

Отсутствие лишних элементов позволяет пользователю быстрее совершать действия, сразу приступать к немедленному действию, не отвлекаться на излишки декора.

**Простой дизайн может выглядеть выигрышнее, чем множество тяжеловесных макетов.**



Рисунок 6

## 6. ПРИСУТСТВИЕ ДВИЖЕНИЯ: Анимация, Gift-картинки и синемаграфика В современном дизайне веб-сайтов широко используется трехмерная типографика, Gift-картинки, анимация.

**3D-типографика** заставляет веб-пространство «ожить», концентрирует внимание посетителей, выгодно выделяет дизайн на фоне других.

Подобные детали используются в логотипах; **синемаграфика** (когда на картинке оживает не все, а лишь отдельный элемент).

* Примером эффектной демонстрации приема может стать фотоснимок блюда, размещенный на сайте кафе, ресторана, от которого вверх подымается пар. Создается эффект свежеприготовленной пищи, которая возбуждает аппетит и желание ее немедленно попробовать.
* **Синемаграфический прием позволяет создать эффект просмотра видео**, при этом сохранить высокую производительность платформы, ведь этот тренд на скорость загрузки значительно не повлияет.

**Анимированные сценарии присутствуют в веб-пространстве повсюду, они заставляют пользователей платформы совершать действие** – кликать (лайкать, переходить на страницу, приобретать услугу или товар).

Поэтому **3D-компоненты повышают конкурентноспособность, конверсию и помогают персонализировать внешний вид веб-сайта**.



Рисунок 7

## 7. ЛЕГКАЯ АНИМАЦИЯ ПОВЕРХ ФОНА

**Двигающиеся пузырьки, атомы, волны могут сделать ресурс интересным** без больших временных затрат.

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\2\09789677687.png

Рисунок 8

## 8. ОБЪЕМ Объектов

Плоские объекты при помощи удачно подобранных теней можно сделать объемными и максимально привлекательными, разместить их в нише общей композиции или выдвинуть вперед из фонового изображения.

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\2\88888.jpg

Рисунок 9

## 9. Качественнный ФОТОМАТЕРИАЛ

**Ключевыми составляющими веб-дизайна являются элементы, помогающие донести информацию без слов, в том числе Фото.**

**Выглядят трендово:**

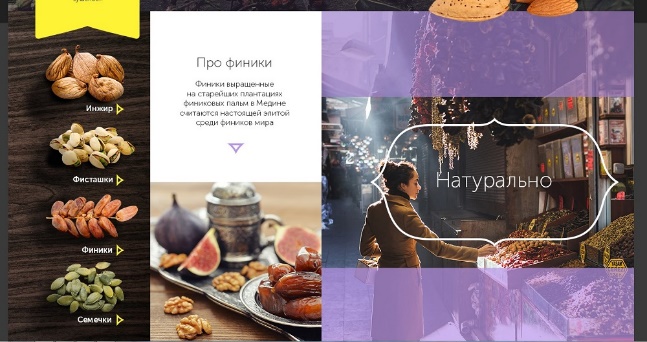
* экстраординарные фотографии в необычной обстановке;
* фото, совмещенные с рисованными элементами;
* снимки с настоящими текстовыми блоками (вывесками, пластиковыми буквами, стендами, табличками, плакатами);
* реальные фотоснимки без обработки и постановочных сцен (честность в фото ценится в текущем году, на пике популярности фотографии, сделанные спонтанно в непринужденной атмосфере и естественной обстановке);
* уникальные панорамные фото на главной странице;
* сolor-gel фотоснимки – уникальная техника создания необычных фото при помощи цветных источников света.
* 

Рисунок 10

## 10. ТВОРЧЕСКИЕ ДУДЛЫ

**Иллюстрации, нарисованные вручную, придают сайту особый шарм, стилистику и атмосферу.** Здесь нет места пропорциям и четкости.

В дизайне могут встречаться грубые линии, неаккуратные штрихи, небрежные надписи, мазки.

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\2\1010.jpg

Рисунок 11

## 11. Строгий Брутализм

## Брутальный стиль:

## - применяются квадратные, прямоугольные формы

## - отсутствие плавных линий

## - отсутствие множества мелких элементов

## - все блоки создаются крупного размера,

## - используется сдержанная палитра цветов (белый, черный, коричневый, серый)

## - классические жирные шрифты

# - броские крупные заголовки

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\2\1111.jpg

Рисунок 12

## 12. МЯГКИЙ ДИЗАЙНЕРСКИЙ ШАРМ

Противоположностью предыдущей тенденции становится **изысканное сочетание плавных линий, мягких теней, пастельных тонов**.

«Легкий» веб-дизайн привлекает воздушностью, дает взгляду отдохнуть, не вызывает у пользователей раздражения.

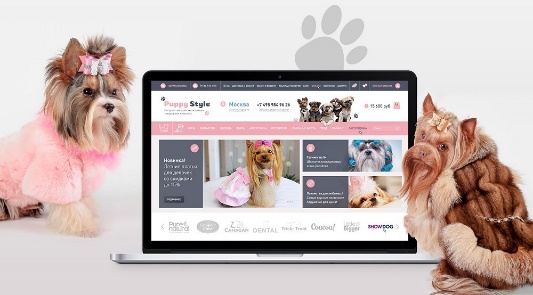


Рисунок 13

## 13. ВОЗВРАЩЕНИЕ ИЗ ПРОШЛОГО

Узоры и цветовая палитра, модная в 80-90-х годах, активно прослеживается в современности. **Речь идет о резком контрасте, кислотных тонах интерфейса, фона, кнопок.**

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\2\1313.jpg

Рисунок 14

## 14. КРЕАТИВНЫЙ ЭКРАН ЗАГРУЗКИ

**Переход между страницами** – неотъемлемая составляющая поведения заинтересованного пользователя.

Веб-дизайнеры стремятся сделать эти секунды комфортными для пользователя. Чтобы заинтересовать посетителя, они активно избавляются от скучных и устаревших экранов загрузки, внедряя необычные варианты. **Подобные изобретения делают платформу высокоуникальной.**



Рисунок 15

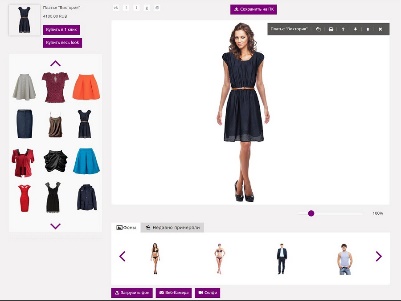
## 15. Использование ТРЕХМЕРНЫХ ОБЪЕКТОВ

Полезные интерактивные элементы могут сделать веб-ресурс востребованным.

**3D-примерочные в онлайн-магазинах одежды, обуви и аксессуаров** – способ подробно рассмотреть вещь со всех сторон.

**Такая новинка мультифункциональна, решает множество задач:**

* привлекает внимание покупателей;
* увеличивает прибыль (пользователи охотнее приобретут вещь, которую смогли тщательно рассмотреть);
* делает сайт эксклюзивным, улучшая его внешний вид.

****

****

Рисунок 16

## 16. PARALLAX – эффект

Параллакс (от греческого слова Parallax) означает смену, чередование. В природе так названо изменение видимого положения объекта по отношению к удаленному фону - в зависимости от того, где находится наблюдатель.

В веб-дизайне - **особый эффект, когда изображения на дальнем фоне двигаются медленнее, чем те, что находятся на переднем плане.**  Вот, например, [как это выглядит](https://cyclemon.com/).



Рисунок 17

А если использовать сразу несколько слоев, поиграть со скоростью движения элементов - возможности этого способа просто неисчерпаемы.

## 17. ЭКСПЕРИМЕНТАЛЬНЫЕ ШРИФТЫ

Геометрический, кастомный шрифт: к**роме типа шрифта, также в моде эксперименты с направлением текста.**

 В последнее время участились случаи разработки шрифтов на заказ, потому что надписи, тексты способны заинтриговать и продемонстрировать ключевые преимущества продукта/услуги.

# F:\SPIRINA\ЛЕКЦИИ\WEB technology 2017\! Web Technology 2019\Web Tehcnology 2020\6 Theme\1\2\1717.jpg

Рисунок 18

## 18. МИНИАТЮРНЫЙ ВИДЕОБЛОК

Видеоролик на онлайн-ресурсе – модный способ преподнести информацию.

Статистика доказала, что пользователи считают: лучше один раз увидеть, чем множество раз слушать.

**Видеоблок должен быть коротким, информационно емким,** снятым таким образом, чтобы при его повторе граница конца и начала ролика максимально скрывалась.

## 19. ПАНТОН – ЦВЕТ года

Ультрафиолет признан цветом №1 в 2019 году. **В его основе содержится совмещение синего и фиолетового цвета,** в результате чего получается цвет потрясающей красоты и глубины. Лавандовый, сиреневый и лиловый тоже в тренде. Эти цвета пробуждают внимание и способствуют высокой концентрации.

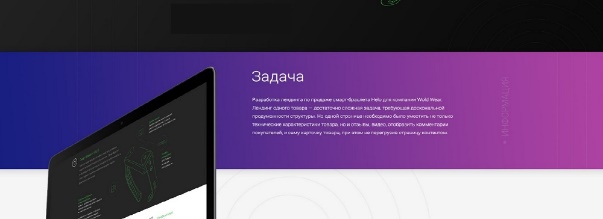


Рисунок 19

Цвет года 2020 г - **19-4052 Classic Blue** (Классический синий)

Успокаивающее присутствие, внушающее спокойствие, уверенность и чувство сопричастности

## 20. СМЕЩЕНИЕ В СТОРОНУ ВЗАИМОДЕЙСТВИЯ ЭЛЕМЕНТОВ САЙТА С ПОЛЬЗОВАТЕЛЕМ

Практичность – тенденция, без которой гармоничный и визуально привлекательный веб-сайт не сможет удержать своих посетителей. **Веб-дизайн должен быть функциональным: адаптивным, содержать полезные опции** (онлайн-чат, поиск).

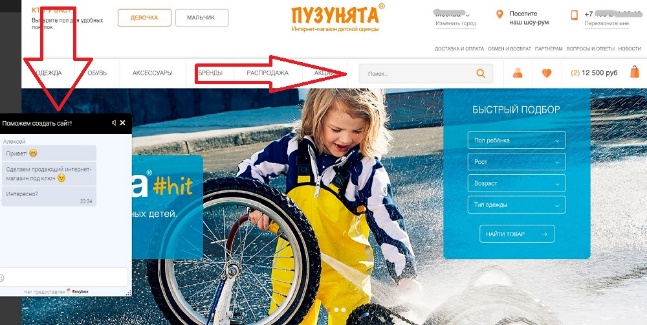
****

Рисунок 20

**Представленные тренды реализуются в различных вариантах, нечасто пересекаются,** так как каждый из них – самодостаточная идея для разработки стоящего проекта. В некоторых случаях смешивание и творческая обработка двух или трех описанных тенденций помогает создать по-настоящему запоминающийся дизайн.

Выбор комбинаций зависит от задумки и направления веб-ресурса, а также таланта и видения веб-дизайнера.

# ****4 Способы сделать Веб-сайт запоминающимся и удобным для посетителей****

## Зачем нужны эффекты?

**Чтобы выделится среди конкурентов.** Эффекты способны выделить сайт из сотен других.

**Чтобы увеличить трафик - количество посещений сайта.**

1. **Чтобы увеличить продажи.** Многие эффекты не просто красивы, но и функциональны. Интересные решения побуждают посетителя принять решение в вашу пользу.

## Необычные фишки в веб-дизайне

### **1. Игра “Выбери тариф”**

Начнем с самого простого.

Сайт оператора предлагает пользователям выбрать варианты тарифов и наглядно увидеть разницу. **Создается впечатление, что клиент сам решает, что выбрать.**

И не обязательно, что это будет самый дорогой тариф - пожалуйста, хозяин барин, выбирайте самые простые варианты.

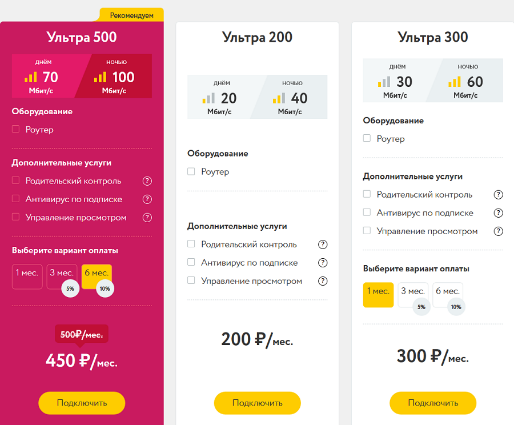
****

Рисунок 21

### 2. Прием “Сравнение”

Лучший способ мотивировать клиента сделать заказ, совершить покупку - показать ему все плюсы вашего предложения.

Лучший способ - **наглядно показать, что он получит и сколько сэкономит**.

Этот прием можно использовать практически в любой тематике.

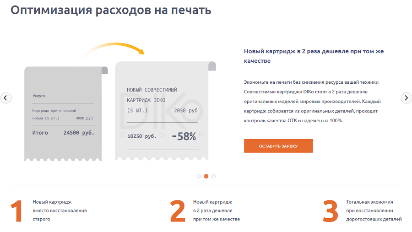
****

Рисунок 22

### **3. Необычное меню**

Существует некоторые виды креативного меню, которые привлекут внимание пользователей. Один нюанс: консервативные люди могут и не оценить. Они привыкли, чтобы все было “как раньше”, привычно и понятно.

Поэтому **такой прием лучше использовать, если ваша целевая аудитория - люди молодые и современные,** с восторгом относящиеся к новым тенденциям. Им точно понравится.

**Обратите внимание на**[адаптивную версию](https://idbi.ru/blogs/blog/adaptatsiya-sayta-pod-smartfony-i-planshety)**:** необычное меню должно быть не только удобным (человек должен разобраться в нем самостоятельно или с помощью ваших подсказок), но и отображаться на всех носителях: ПК, смартфонах, ноутбуках, планшетах.

Если система глючит - лучше с таким меню не играть.

### **4. Нover-эффект**

Нover-эффект: **наводишь курсор или двигаешь мышкой - а картинки меняются**.

Двигаются, приближаются и удаляются, из черно-белых становятся цветными, а то и вовсе пропадают.

Такое зрелище завораживает: пользователи способны долгое время провести на сайте, наблюдая за изменениями.

Вам же лучше: улучшаются поведенческие факторы, ваш сайт получает одобрение от поисковых систем. А сам дизайн становится более живым.

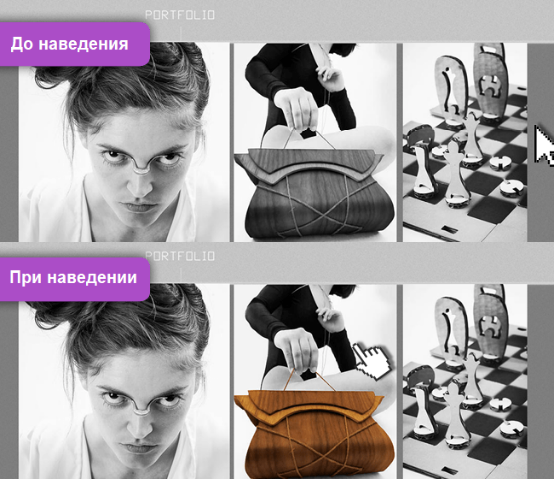
****

Рисунок 23

### **5. Уникальные иллюстрации**

Где взять уникальные картинки для своего сайта?

- Сделать фото самим. Отличный вариант, но придется потратиться на профессиональное оборудование и услуги фотографа.

- **нарисовать иллюстрации вручную**. Это практически беспроигрышный вариант: сейчас в моде все крафтовое - то есть сделанное вручную. Понятно, что **придется пригласить специалиста-иллюстратора** и не раз согласовывать варианты рисунков, но дело того стоит.

Если получится угадать миссию и основную идею сайта, дополнить весь остальной контент авторскими иллюстрациями - получится очень интересно и необычно.

### **6. Сплит-макеты**

**Модный эффект - разделение экрана на две равные части.**

Такой дизайн решает сразу несколько проблем: привлекает внимание посетителя к каждому элементу, дает возможность показать в 2 раза больше информации на одном экране (особенно это актуально для лендингов, интернет-магазинов с широким ассортиментом да и везде, где много информации, и посетитель может растеряться). А еще дает возможность выбора - вспоминаем прием “Сравнение”.

Возможностей для реализации такой идеи тоже множество. **Игра на контрастах,**[анимация](https://idbi.ru/blogs/blog/animatsiya-v-veb-dizayne)**, 3D-эффекты,**[необычная типографика](https://idbi.ru/blogs/blog/tipografika-v-veb-dizayne) - все к вашим услугам.

### **7. Виртуальная реальность**

**Виртуальные карты, онлайн-трансляции, 3D-путешествия по вашему офису, складу привлекают пользователей.** Это возможность заглянуть одним глазком в чужую жизнь, как в замочную скважину (видеомаркетинг).

Можно просто снимать и вставлять видео на сайт - главную страницу, карточки товаров, а можно сделать 3D-панораму, которая будет показываться посетителю постоянно. Например, продаете вы туристические путевки - показывайте карту стран, достопримечательности, прямые включения из городов мира, создавайте у себя на ресурсе эффект присутствия.

Это и есть виртуальная реальность - кто-то очень умный догадался использовать эту популярную технологию и в веб-дизайне.

Можно “перенестись” куда захочешь, “побродить” по виртуальному пространству, ощутить прикосновения, послушать звуки, послушать VR-помощника, который расскажет о том, что происходит. Некоторые сайты смотришь, как кино: тут вам и эффект присутствия, и 360- градусные панорамы, и игры - что пожелаете.

**Наиболее эффективно этот прием работает, когда человеку предлагается что-то протестировать.** Как выбирать машину, если не едешь в автосалон? Легко: в виртуальной реальности [тест-драйва для Volvo XC90](http://www.volvocars.com/us/about/our-stories/google-cardboard) можно было оценить габариты автомобиля, почувствовать его скорость, отрегулировать сиденья.

<https://www.volvocars.com/us/about/our-stories/google-cardboard>

**Более демократичный вариант - панорама в 2D пространстве.** Это не совсем виртуальная реальность, но уже с эффектом присутствия - например, стоит покрутить мышкой - и как будто поворачиваешь голову влево, вправо, вниз и вверх.

### **8. Оптические иллюзии**

Оптические иллюзии вдохновляли, наверное, каждого человека. Движущиеся круги, картинки с двойным смыслом. Функциональной нагрузки они практически не несут: главная цель оптических иллюзий - привлечь посетителя, заставить приковать взор к вашему сайту и остаться там навсегда (шутка).

**Оптические иллюзии** **притягивают внимание, развивают творческое воображение и вообще круто смотрятся.**

Основные приемы при создании иллюзий: изменение привычных представлений о форме и пространстве, выход за рамки стандартов, хитрые контрасты, способность увидеть то, что на первый взгляд недоступно.

**Вот как это можно применить на сайте.** Прямо сейчас наведите курсор мыши на картинку и покрутите колесико вниз и вверх - чувствуете вибрацию?

**Есть в оптических иллюзиях и функциональность, в основном это придание элементам двойного смысла,** что подчеркивает креативность бренда.

Часто этот прием используется при создании [фирменного стиля](https://idbi.ru/blogs/blog/chto-takoe-firmennyy-stil-i-kak-ego-razrabotat), особенно [логотипов](https://idbi.ru/blogs/blog/pravila-idealnogo-logotipa).

Вот посмотрите на лого компании FedEx - **между буквами E и X расположена белая стрелка.** Это и игра воображения, и вполне конкретный посыл от бренда: мы движемся вперед, come together, my friend.

## Как правильно внедрять интересные фишки?

1. **Главный принцип - не навреди.**Описанные эффекты - достаточно сложная штука.
2. **Постоянно тестируйте.** Если дизайнер справился, все получилось отлично, нужно еще проверить, как отреагирует аудитория. Вдруг людям не понравится?
3. Поэтому любую фишку нужно протестировать, а уж потом запускать в работу.
4. Приготовьтесь к тому, что дизайн придется менять, 2, 5 или 100 раз - пока не найдется оптимальный вариант, который устроит всех.
5. Не стесняйтесь спрашивать напрямую у пользователей, нравятся ли им нововведения.
6. Это укрепит обратную связь и создаст вам репутацию лояльной компании, которой интересно мнение клиентов.
7. **Не копируйте бездумно.** Часто так бывает - понравилась идея, надо срочно утащить ее себе на сайт. Но, во-первых, помните об авторских правах, а во-вторых, любую хорошую идею можно чуть изменить, дополнить, подстроить под себя. Просто адаптируйте ее к вашей нише, специфике.

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 8. Инструменты разработки Интернет приложений**

План:

1 Классификация инструментов разработки веб-сайтов

2 Средства разработки приложений веб-клиента

3 Специализированные языки программирования

4. Веб-редакторы.

5. Графические редакторы и мультимедийные системы

6. CMS - Система управления контентом

7. Онлайн конструкторы сайтов

**1 Классификация средств разработки Web-сайтов**

Существуют различные средства для разработки Web-сайтов, в зависимости от типа и назначения разрабатываемого ресурса:

- HTML;

- CSS;

- Javascript. Фреймворки Javascript

- Специализированные языки программирования

- Web-редакторы или Редакторы [HTML](https://t-webstudio.ru/zakazat-sayt-v-html/)

- CMS;

- Онлайн конструкторы сайтов - Wix;

По технологии отображения контента сайты различаются:

* Статические - состоящие из статичных HTML страниц, содержание которых не меняется. Пользователю выдаются файлы в том виде, в котором они хранятся на сервере. Средство разработки - HTML
* Динамические - состоящие из динамичных HTML страниц-шаблонов, информации, скриптов и прочего, в виде отдельных файлов. Содержимое генерируется по запросу специальными скриптами на основе других данных из любого источника.

- Сайты, созданные с применением Flash-технологий - весь сайт располагается на веб-страницах, предназначенных для загрузки в Flash-файл, а вся навигация и контент реализованы в самом Flash-ролике.

# 2. Инструменты разработки клиентского веб-приложения:

# HTML - язык разметки гипертекста - это стандартизированный язык разметки документов, средство структурирования и форматирования документов в Интернете.

# Cascading style sheets CSS

# Технология CSS (каскадные таблицы стилей) - каскадные таблицы стилей используются для форматирования внешнего вида веб страниц.

# Таблицы стилей - это содержимое файлов CSS. Правила таблицы стилей определяют внешний вид одного или нескольких элементов страницы.

# В то время как HTML структурирует документ, CSS дает браузеру инструкции о том, как отображать конкретный элемент - дизайн, размещение в пространстве и позиционирование.

# JavaScript

# JavaScript можно использовать для разработки программ под широкий спектр разнообразных платформ.

# На этом языке можно писать серверные сценарии (используя .NET или Node.js), обычные приложения (способные работать в любых операционных системах) и расширения для приложений, приложения для мобильных устройств и сценарии командной строки.

# JavaScript отличается от многих других языков программирования.

# В нем отсутствуют классы, а функции являются обычными объектами, которые используются для решения самых разных задач.

# Любой браузер включает интерпретатор JavaScript, а также код на языке JavaScript можно встраивать в HTML-документы, в теги <script>, и при загрузке HTML- документа этот код будет выполняться браузером.

**Фреймворки Javascript: Angular, React, Vue и другие**

JavaScript-фреймворки стали неотъемлемой частью каждого современного веб-проекта, наиболее часто используемые Vue.js, React, Angular

### Javascript фреймворки в первую очередь сравниваются по рендерингу страницы (Рендеринг – отображение конечного результата).

### Современная архитектура допускает два вида: выполнение на стороне клиента (страница отрисовывается за счет мощностей ПК пользователя) или на стороне сервера.

Каждый из фреймворков подходит по-своему к обработке DOM что и оказывает влияние на рендеринг конечной страницы, отображаемой на экране пользователя.

[DOM](https://ru.wikipedia.org/wiki/Document_Object_Model) – Document Object Model – объектная модель документа, позволяющая считывать и менять содержимое, оформление и даже структуру html-документов.

**Vue.js и React** создают копию DOM, обрабатывают ее, а затем результат сравнивается с исходной версией. В конечном документе (то есть на экране пользователя) заменяются только те части страницы, которые отличаются от результатов обработки.

Это значительно ускоряет загрузку и рендеринг страницы. Соответственно сокращается объем трафика, что особенно важно для пользователей мобильных устройств.

[Фреймворк Angular версии 1.x и выше](https://mkdev.me/posts/zachem-nuzhen-angular-js-i-pochemu-imenno-on) использует другой подход к обработке DOM. Здесь происходит разделение на два потока, причем за рендеринг DOM «отвечает» браузер (клиентская часть), а за создание директив, загрузку кода и сервисов – общий поток (серверная часть).

# 3. Специализированные языки программирования

# PHP - это серверный язык сценариев.

# PHP - язык программирования, специально разработанный для написания web-приложений (скриптов, сценариев), исполняющихся на Web-сервере. Синтаксис языка во многом основывается на синтаксисе C, Java и Perl.

# Сценарии на языке PHP могут исполняться на сервере в виде отдельных файлов, а могут интегрироваться в html страницы.

# Perl (Practical Extraction and Report Language) - интерпретируемый язык программирования, предназначенный для обработки текстовой информации и написания различных системных программ в операционной системе UNIX.

# - ASP (Active Server Page) - активные страницы сервера, т.е. скрипт интерпретируется и исполняется непосредственно на сервере, после чего пользовательскому браузеру отправляется уже готовый html-документ с результатами работы сценария ASP.

**4. Web-редакторы.**

**Adobe Dreamweaver.**

Adobe Dreamweaver - программное обеспечение для создания профессиональных веб-сайтов. Программа Adobe Dreamweaver содержит интуитивно понятный визуальный интерфейс, позволяющий создавать и редактировать веб-сайты и мобильные приложения.

Dreamweaver поддерживает следующие языки программирования: ActionScript, C#, CSS, XML, HTML, Java, JavaScript, PHP, Visual Basic, VBScript и другие. Также имеет подсветку синтаксиса.

Dreamweaver предоставляет библиотеку элементов, которые можно применять на веб-страницах, например, панель навигации, дескриптор авторского права и др.

# Программа имеет функции для добавления мультимедийных средств на веб-страницы и различных типов файлов от Flash и Java до RealAudio, интерактивных элементов, таких как поисковые системы, форумы, системы электронной коммерции.

# В режиме интерактивного просмотра в Dreamweaver отображаются пользовательские элементы HTML, пользовательские свойства CSS и многое другое.

# Microsoft SharePoint Designer.

# Microsoft SharePoint Designer - это программа для разработки сайтов под управлением SharePoint. В составе пакета: редактор дизайна, среда разработки и ряд дополнительных инструментов для облегчения задачи создания и настройки корпоративного портала

# Microsoft Office SharePoint Designer - это типичный веб-редактор, сочетающий характерные элементы управления веб-узлом MS Sharepoint, а также инструменты редактирования веб-страниц. Он может с успехом применяться и для создания несложных статических или динамических HTML-страниц, а также активных серверных страниц стандарта ASP.NET.

# 

Стартовое окно Microsoft SharePoint Designer

# Adobe Muse.

# Adobe Muse - это отличный выбор для дизайнеров и тех, кто на профессиональном уровне, проектирует веб-страницы. Программа обладает серьёзным функционалом и соответствующе высоким порогом вхождения.

# Muse позволяет создавать адаптивные сайты (лендинги и визитки) на HTML5. Сайт можно сохранять на жёстком диске, выгружать на хостинг или в какое-нибудь облако. Muse поддерживает шаблоны.

# Adobe Muse имеет библиотеку готовых мини-приложений.

# 

# Bluefish Editor.

# Bluefish Editor является мощным редактором, ориентированным на программистов и веб-разработчиков, с большим количеством опций для написания веб-сайтов, скриптов и программного кода.

# 

# 5. Графические редакторы и системы мультимедиа

# Графические редакторы и системы мультимедиа являются частью разработки веб-сайтов.

# Adobe Flash.

# Adobe Flash - это полноценная среда для подготовки многофункционального интерактивного контента. Платформа Flash широко применяется для создания привлекательных приложений благодаря богатым возможностям по использованию видео- и аудиоматериалов, графических объектов и анимации.

# Оригинальный контент можно создать непосредственно в Adobe Flash или импортировать из других программ Adobe, таких как Photoshop или Illustrator, быстро создавая анимации и мультимедийные элементы, а также разрабатывать сложные интерактивные проекты на языке ActionScript 3.0.

# Flash используется для создания многоцелевых сайтов, автономных приложений или приложений для устройств под управлением операционных систем Android и iOS.

# 

# Adobe Photoshop.

# Adobe Photoshop - самая популярная и широко распространенная программа редактирования изображений.

# Adobe Photoshop - это эталон качества цифровых изображений, обеспечивает высокую производительность, мощные возможности редактирования графики и имеет интуитивно понятный интерфейс.

# 6. CMS - Система управления контентом

# Система управления контентом (Content management system, CMS) - информационная система или компьютерная программа, используемая для обеспечения и организации процесса создания, редактирования и управления содержимым (контентом)

# Система управления - программа, предоставляющая инструменты для добавления, редактирования, удаления информации на сайте или приложении.

# Существуют разнообразные системы управления, среди которых встречаются бесплатные и платные, построенные по разным технологиям

В общем случае CMS делятся на:

* ECMS (Enterprise Content Management System) - система управления неструктурированным содержанием предприятий.

Такие системы предназначены для управления контентом масштаба предприятия. ECMS имеют глубокую внутреннюю классификацию по предметным областям (HRM, DMS, CRM, ERP и т. д.).

* WCMS (Web Content Management System) - система управления веб-контентом.

# Подобные CMS позволяют управлять текстовым и графическим наполнением сайта или приложения, предоставляя пользователю интерфейс для работы с его содержимым, удобные инструменты хранения и публикации информации, автоматизируя процессы размещения информации в базах данных и её выдачи в HTML

# Термин CMS заместил собой WCMS, превратившись в синоним системы управления сайтами.

Рассмотрим несколько примеров самых распространенных CMS:

* WordPress.

WordPress - это современная бесплатная CMS-платформа

WordPress используется для разработки интернет-ресурсов практически любого типа: от обычных блогов до новостных порталов со сложной структурой

**Joomla!**

Веб-сайт на базе Joomla состоит из нескольких элементов, которые все вместе образуют веб-страницу. Основными являются три элемента - содержимое, модули и шаблон.

Содержимое - это основа веб-сайта. Информационное наполнение для основной части веб-страницы Joomla загружается из компонентов.

# Шаблон определяет внешнее представление веб-сайта. Шаблон играет роль фильтра (или линзы). Он управляет всеми аспектами представления веб-страниц. Шаблон не содержит никакого информационного наполнения, но может включать логотипы.

# Drupal - это открытая система, которая используется для создания веб-сайта, а также платформа управления его содержимым с помощью дополнительных модулей расширения.

# Drupal содержит множество программных компонентов мощной системы управления контентом: средства входа и регистрации пользователей; механизмы определение типов пользователей и контента; различные права доступа; инструменты создания, редактирования, систематизации и управления данными и др

# 1С-Битрикс - коммерческая CMS, которая по праву получила широкое распространение в электронной коммерции.

# CMS 1С-Битрикс - это многопользовательская система, которая обеспечивает возможность одновременной работы нескольких редакторов и большого количества авторов с разными правами доступа

# 7. Онлайн конструкторы сайтов

# Существуют различные конструкторов сайтов: Wix, Weebly, uKit (uCoz), Google Sites, Webnode, Jimdo, uCraft, Webstarts, Tilda и др.

# Различают онлайн-конструкторы и программы-конструкторы. Онлайн-конструкторы работают полностью в браузере, а данные сохраняют на собственном сервере. Чаще всего, услуга хранения платная, и пользователи сильно ограничены тарифными планами услуги.

# Программы-конструкторы устанавливаются на компьютер, как программа. Оффлайн-конструкторы сайтов похожи на графические редакторы ‒ с той разницей, что на выходе клиент получает архив со страницами будущего веб-сайта.

# Его можно залить на любой хостинг, купить доменное имя ‒ и, таким образом, опубликовать в интернет.

# Read more: <https://www.plerdy.com/ru/blog/website-builders-review/>

# [WIX](http://ru.wix.com/) – один из самых мощных и функциональных конструкторов. Возможностей хватает для [создания форумов](https://texterra.ru/blog/kak-sozdat-i-raskrutit-forum-poshagovyy-plan.html), сайтов- визиток, одностраничников - лендингов, магазинов, блогов.

# Приложение позволяет клиентам выбрать дизайн с уже готовым функционалом среди более 500 шаблонов в конструкторе сайтов, работающем на технологии Drag&Drop.

# 

# WIX

# Weebly

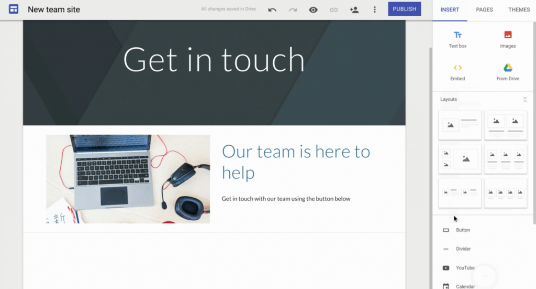
# Weebly — лучшее приложение конструктора сайтов для блога, веб-представительства или интернет-магазина. В отличие от WIX, этот конструктор позволяет переключаться между темами прямо в редакторе. Более гибкий drag&drop-интерфейс ‒ изюминка Weebly.

# Read more: <https://www.plerdy.com/ru/blog/website-builders-review/>

# Weebly

Google Sites

Пользовательский конструктор вебсайтов предлагает выбор из 6 шаблонов, пропитанных духом минимализма в каждом пикселе. Шаблоны адаптируются под экран мобильных устройств. Редактирование на мобильном телефоне доступно так же, как и работа с Google Docs.



Read more: <https://www.plerdy.com/ru/blog/website-builders-review/>

# uCraft

# uCraft - простой конструктор сайтов, предоставляющий широкие возможности для настройки собственного веб-сайта-визитки, с перспективой дальнейшего развития и улучшения.

# uCraft

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с

**Лекция 9. Usability (юзабилити) Web сайта**

План

1. Понятие юзабилити

2. Принципы юзабилити

3. Основные правила юзабилити сайта

**1. Понятие юзабилити**

# Юзабилити (от английского usability — удобство использования) — это качественная оценка простоты и комфорта работы с сайтом.

# Пользователь должен без труда находить нужную информацию, не теряясь в функционале и многочисленных страницах, и при этом получать эстетическое удовольствие от работы с ресурсом

## <https://convertmonster.ru/blog/uzabiliti-blog/osnovy-juzabiliti-sajta/>

<https://webformyself.com/principy-i-pravila-yuzabiliti-sajta-dlya-masterov-veb-dizajna/>

**Юзабилити веб сайта** оценивается **по 5 основным качественным компонентам**:

**1. Ориентация:** насколько просто новым посетителям веб-сайта совершать элементарные действия.

**2. Эффективность:** насколько быстро пользователь может ориентироваться на сайте и совершать необходимые ему действия.

**3. Запоминаемость**: насколько легко будет пользователю сориентироваться на сайте после продолжительного отсутствия на сайте.

**4. Ошибки:** количество ошибок, совершенных посетителем сайта, способы ликвидации этих ошибок и простота в устранении последствий этих ошибочных действий.

**5. Удовлетворенность:** субъективная степень удовлетворенности пользователя, эмоциональное восприятие сайта:

совокупность информации+дизайна+навигации+отображения в браузере.

**Полезность**: показатель функциональности сайта для пользователя: дает ли вебсайт пользователю то, что он ищет.

Вообще, **юзабилити и полезность** – это два самых главных критерия для **определения практичности** чего-либо.

Но при определенных условиях:

* этот сайт должен быть действительно нужен посетителю
* интерфейс сайта не должен мешать пользователю совершить [целевое действие](https://convertmonster.ru/blog/landing-page-blog/celevoe-dejstvie-lending-page/) и получить то, что он хочет.
* Вот ***краткая формула для выявления практичности вашего сайта:***

**Полезность сайта** — обладает ли сайт ответами на ваши вопросы.

**Юзабилити** — насколько легко и удобно управлять сайтом.

**Практичность сайта** = юзабилити + полезность.

# [Анализ юзабилити](https://www.ashmanov.com/usability/) важен для того, чтобы исправить недоработки текущего сайта и повысить продажи, и еще важнее - чтобы с самого начала [разработать удобный для пользователей сайт](https://www.ashmanov.com/web-dev/sozdanie-internet-magazina/).

# 2. Принципы Usability

# Разработка юзабилити сайта базируется на главном правиле: «чем проще — тем лучше».

# Чем более привычную структуру и интерфейс видит пользователь — тем легче ему сориентироваться и тем меньше времени он потратит, чтобы найти нужный раздел, т.к. будет чувствовать себя более уверенно.

### **1. Дизайн — залог доверия к сайту**

Не перегружайте визуальную часть множеством деталей, цветов и динамичных элементов.

Пользователя ничего не должно отвлекать от восприятия информации. А при длительном нахождении на ресурсе он не должен уставать от обилия контраста, плохой читабельности текста, разноплановой графики.

Желательно минималистичный дизайн: в сумме с уникальностью следует продумать сочетание цветов фона и шрифта, число и вид изображений.

Грамотно продуманный «внешний вид» сайта — залог доверия со стороны потребителя, попавшего к вам впервые.

Простота и доступность использования, цветовые сочетания, стиль, подходящие вашей целевой аудитории картинки, качество контента, отсутствие ошибок — все это влияет на расположение к ресурсу.

# image3.png

*Пример сайта, перенасыщенного графикой. Дизайн пестрит рассеивающими внимание цветовыми сочетаниями, обилием картинок.*

### **2. Логичный структурированный контент**

Один из главных факторов юзабилити — удобное восприятие текстов.

**Параметры контента**

- Структура, оформление, информативность, логическая разбивка, оптимизация, отсутствие ошибок.

Если это длинный текст-руководство — сделайте оглавление в виде ссылок, чтобы пользователь мог быстро найти нужный раздел.

# image5.png

# Важна продуманная структура графики и текста: выделение приоритетной информации, последовательность расположения.

### **3. Расположение важной информация**

Посетители просматривают web страницу по Z-паттерну или F-патерну (шаблону): внимание концентрируется в верхнем левом углу и ограничивается первыми несколькими абзацами текста в убывающем порядке.

Потому при распределении важной информации использование такой схему увеличит вероятность того, что посетитель быстро найдет нужные данные, увидит самое важное, бегло пробежавшись по разделу взглядом.



### **4. Общение с пользователем**

Успешная регистрация, принятый заказ, несуществующая страница — пользователь должен быть всегда в курсе происходящего.

Причем во всех подробностях. Если он зарегистрирован в системе — расскажите, что делать дальше, если заказ принят: укажите номер, время ожидания звонка, контактные данные; если попал на страницу 404 — как найти нужную информацию или куда вернуться.

*Попав на такую страницу 404, пользователь не растеряется и, скорее всего, не сразу покинет ресурс.*

### **5. Отмена действия**

У человека должна быть возможность передумать.

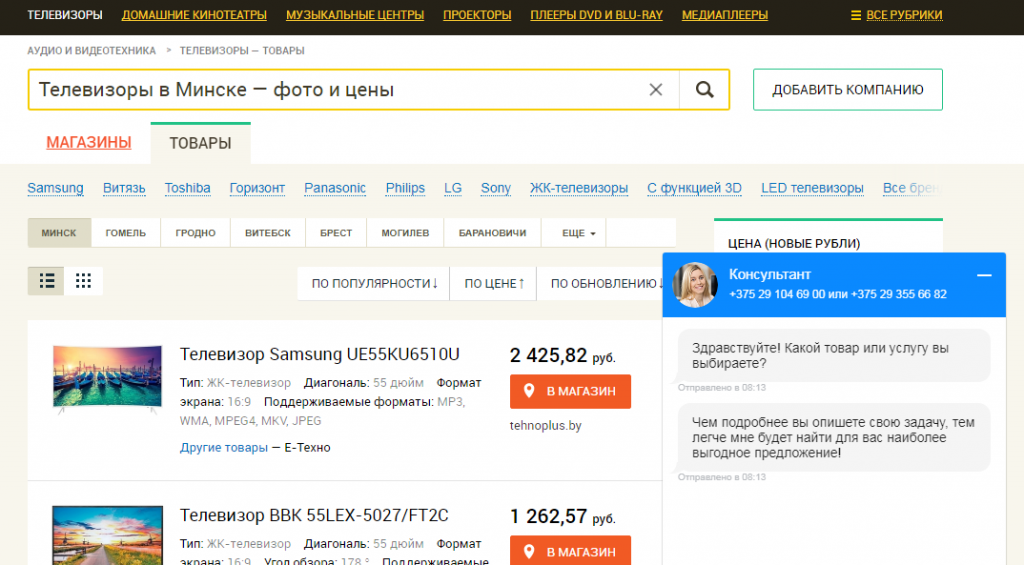
К примеру, социальная сеть ВКонтакте позволяет восстановить страницу после ее удаления. Некоторые интернет-магазины — изменить или удалить заказ после его оформления, откорректировать способ доставки и оплаты. Все это — составные части юзабилити веб сайта.

### **6. Ненавязчивое предложение помощи**

Не нужно настаивать на общении.

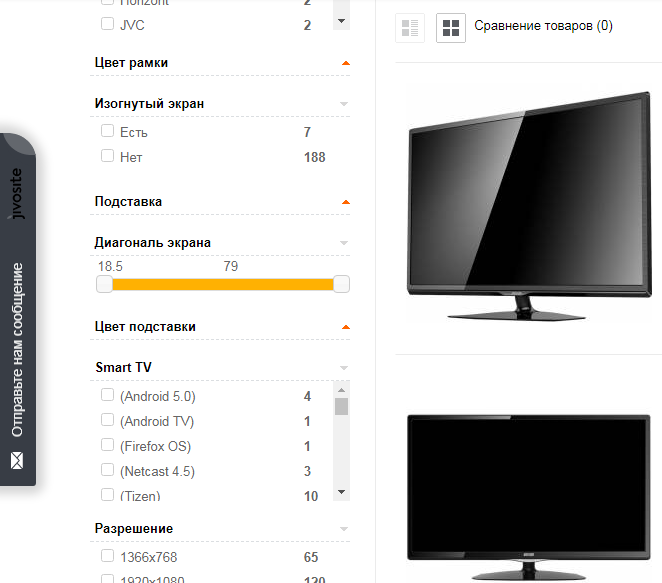
Большие выскакивающие окна с предложением помочь в выборе, цветная полоска с номером телефон, занимающая треть экрана — подобные элементы не только смотрятся навязчиво, отталкивают посетителя, но и закрывают часть функционала.

В примере ниже — форму подбора товара. Окно помощи должно легко скрываться, потому крестики и другие подобные значки нужно делать заметными, а не полупрозрачными или сливающимися с фоном.



Лучше, чтобы изначально данная форма была в свернутом состоянии, но при этом заметной — к примеру, в виде яркой боковой вкладки.

И при этом не перекрывала часть контента на странице. Последнее относится и к кнопкам социальных сетей.

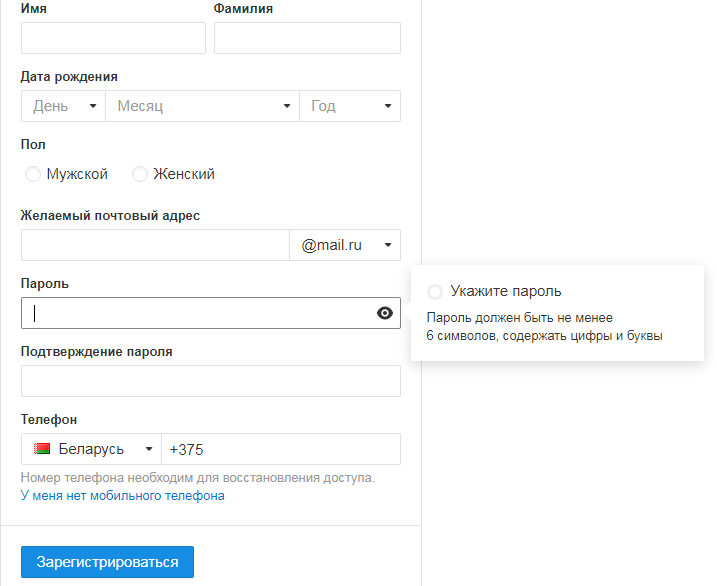


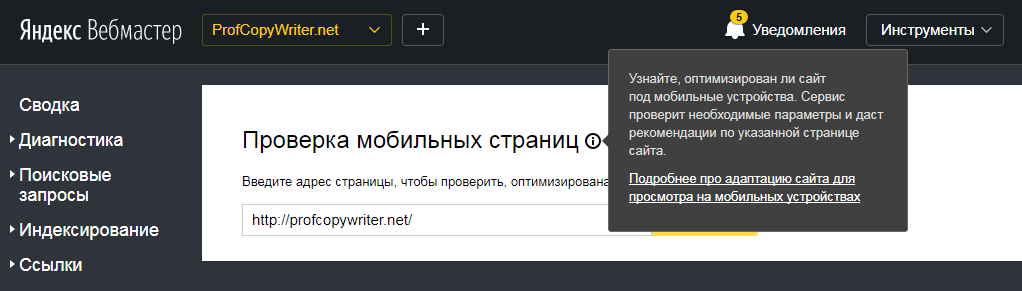
### **7. Понятный контент**

Не используйте специализированные термины — говорите проще, чтобы пользователю не пришлось тратить время на толкование непонятного слова.

Если требуются узкие или профессиональные термины — разрабатывайте всплывающие окна - подсказки.

Это же касается заполнения форм: если вы диктуете определенные условия в виде длины имени, пароля, формата телефона, допустимой географии — укажите это в виде подсказок.





### **8. Отказ от резких перемен - не меняйте привычные названия разделов или пиктограмм**

Это касается пунктов меню, названия и вида кнопок, их расположения.

Представьте перестановку в любимом супермаркете: после года ежедневных покупок вы вдруг замечаете, что руководство решило поменять местами отделы, товары, сделать редизайн. И теперь молоко стоит на прилавке для сметаны, каши — там, где раньше были масла, а вместо замороженной вишни вы находите свиные почки. Указатели стали красивее, но совершенно другими — как и привычные значки скидок. Итог: вы теряетесь и тратите намного больше времени, чем обычно, чтобы изучить новое оформление и сориентироваться, заново привыкнуть.

То же самое относится к веб-дизайну: пользователь не только потратит больше времени, но и разочаруется, заметив, что вместо значка корзины теперь — «Добавить товар», вместо «Каталог» — «Продукция», а привычные объемные кнопки обтекаемой кнопки стали плоскими и рублеными.

Потому задумав «перестановку», не шокируйте посетителя — изменения должны быть постепенными и не менять кардинально привычную структуру.

### **9. Наглядность.**

### Все возможности — на виду

Удачное решение — это когда пользователь видит все разделы, в которые может перейти, т.к. в любой момент может быстро сориентироваться, где находится и как вернуться обратно.

Если на сайте планируется много контента — резонно предусмотреть горизонтальное верхнее, вертикальное меню (и то, и другое — возможно, с подразделами), а также «подвал».

Если речь идет о минималистичной «визитке» или landing page — достаточно только верхнего горизонтального меню.

Если категорий очень много — добавьте пункт «Карта сайта», который позволит пользователю сориентироваться, какие разделы содержит ресурс и как ему туда попасть.

**Используйте «хлебные крошки» — навигационную цепочку в верхней части блока контента, которая подсказывает, по какой схеме человек попал на страницу.**



*Пример удачной карты сайта*

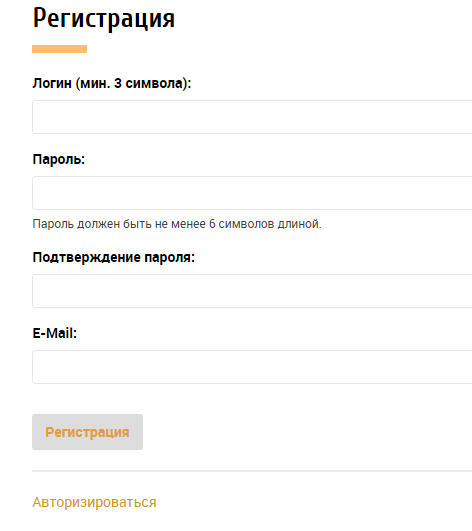
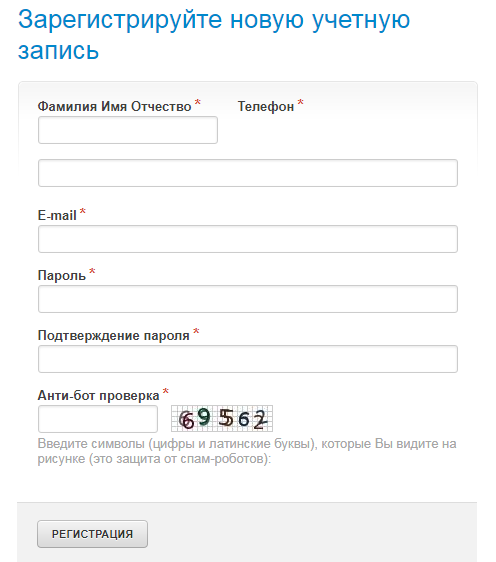
### **10. Форма регистрации и заказа**: чем лаконичнее — тем лучше

Чем больше полей для заполнения — тем меньше пользователю хочется тратить на это время.

Причем компоновка полей, надписей, подбор шрифтов могут быть идеальными, а вот длина — короткой, что неудобно.

Современные основы юзабилити заставляют веб-разработчиков сокращать длину этих форм до минимума, нещадно выкидывая вопросы об образовании, уровне дохода, профессии и т.д.

Но! Сравните две формы: несмотря на наличие всего пары дополнительных полей, заполнять первую хочется меньше (неудобно), чем вторую. Не говоря уже о требовании обязательно указать номер телефона.



### **Строка поиска: Заметное длинное поле поиска**

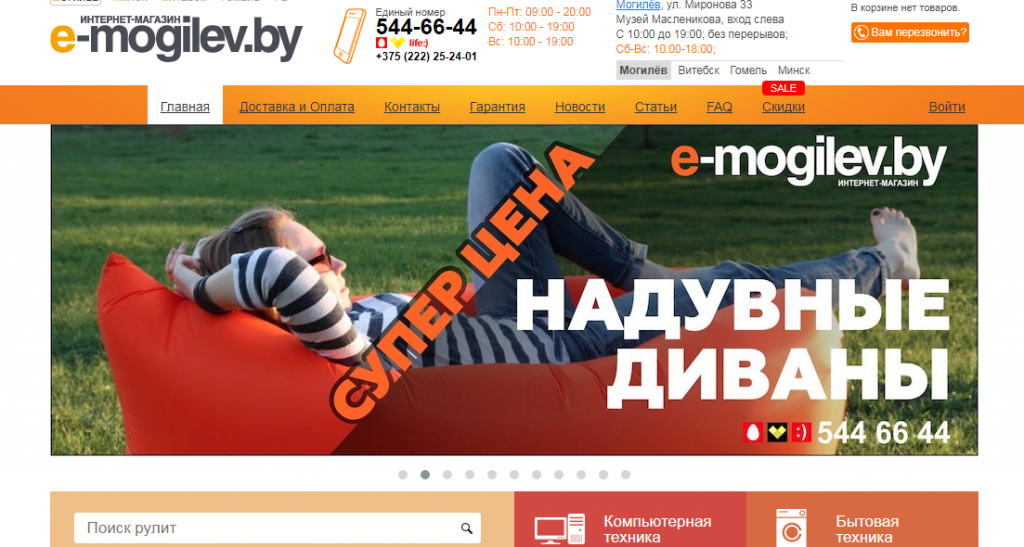
Кнопка поиска — лучший помощник посетителя. Особенно если речь идет о крупном ресурсе.

Строку поиска пользователь должен обнаружить без труда. Чтобы при первом знакомстве он не тратил на обнаружение соответствующей строки даже несколько секунд.

Еще один важный момент — длина поля поиска.

Согласно исследованию Якоба Нильсена, в среднем туда вмещается 18 символов, а оптимальная длина — 27 символов. В 90% случаев сюда поместится даже длинный запрос, чтобы пользователь его полностью увидел и исправил при надобности.

Для установки нужной ширины используйте относительную единицу «em», равную размеру используемого шрифта.



*Строка поиска — не в шапке сайта, а под презентацией, внизу страницы. Не самый удобный вариант, учитывая, что пользователь автоматически ищет эту опцию в верхней части сайта.*

### **12. Больше свободного пространства**

Сайт можно сравнить с жилой комнатой: если каждый квадратный метр чем-то занят — человек будет чувствовать себя неуютно.

Современный дизайн склонен к минимализму: на странице — только самое важное.

Если страница будет переполнена полезными блоками, разделами, ремарками и прочей информацией — посетитель потеряется и не поймет, куда устремить внимание в первую очередь.



Множество блоков с практически незаметными рамками, обилие разнообразной графики, отсутствие выделения приоритетной информации не дают понять, на что обращать внимание в первую очередь при попадании на сайт.

После того как все принципы юзабилити учтены, переходят к тестированию, которое проводится постоянно, т.к. любая ошибка может оттолкнуть потенциального клиента.

**Резюме**

* Отсутствие ошибок
* Качественный текст
* Поиск по сайту.
* F- и Z-паттерны просмотра страниц сайта
* Используйте знакомые всем элементы либо элементы, назначение которых очевидно.
* Элементы, содержащиеся в составе Web-страниц, должны действовать так, как того ожидает пользователь.
* Web-сайт должен обеспечивать работу с ним большинства пред­полагаемых пользователей.
* Пользователь должен получать подтверждение своих действий.
* Согласованное использование цветовой схемы, символьных обозначений и согласованное размещение элементов.
* В рамках Web-сайта должен поддерживаться единый принцип использования на­вигационных элементов.

**В заключении сформулируем основные правила юзабилити сайта**

**Основные правила юзабилити сайта:**

1. **Гарантии надежности**. Вкладка «О компании», контактные данные, сертификаты, адрес офиса, отзывы клиентов, кейсы, банковские реквизиты – всё это дает нелишнее доказательство посетителю, что компания имеет статус и вполне компетентна в своей сфере деятельности.
2. **Кнопка Поиска** - важный элемент для сайта. Кнопка «Поиск» — это первый помощник посетителя и лучший друг на любом сайте.
3. Как минимум – если юзер в течение первых 2 секунд уже нашел эту кнопку, он чувствует себя в безопасности и уровень его напряженности и тревоги к сайта резко спадет.
4. **Навигация («хлебные крошки»).** Правило, согласно которому пользователь всегда должен иметь представление, где он находится: тема, раздел, номер страницы.
5. **Отображение сайта.** Корректное отображение элементов страницы на различных устройствах снижает процент отказов посетителей сайта и увеличивает глубину просмотра, что, в свою очередь, напрямую влияет на показатель конверсии сайта.
6. **Простота использования сайта.** Этот принцип состоит из правил, как скрасить пребывание посетителей на сайте.
7. Вот некоторые из этих идей: Правило 3 кликов, Правило 1 секунды, Правило перевернутой пирамиды, Закон близости элементов, Синдром утёнка.
8. **Дизайн**. Минималистичный дизайн, без отвлекающих элементов, не мешающий посетителю воспринимать информацию. Стоит отметить, что дизайн и юзабилити сайта – это два показателя, которые очень часто пересекаются между собой.
9. **Контент**. Оформление и структура текста, логическая последовательность и оптимизация контента – всё это должно соответствовать определенным правилам, помогающим пользователям быстро найти необходимую информацию в куче букв.

Таким образом, юзабилити важный этап при разработке сайта и необходимо следовать правилам веб дизайна и юзабилити

***Основная литература***

1. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] : практическое пособие / Владимир Дронов. - СПб. : БХВ-Петербург, 2011. - 414 с. : ил. - (Профессиональное программирование). -

2. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL: джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2011. - 890 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Профессиональное программирование).

3. [Гоше, Х. Д.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%93%D0%BE%D1%88%D0%B5,%20%D0%A5.%20%D0%94.)  HTML5. Для профессионалов [Текст]: научное издание / Хуан Диего Гоше. - 2-е изд. - СПб.: Питер, 2015. - 559 с.: ил. - (Для профессионалов). - Пер. изд.: HTML5.For masterminds / J.D.Gauchat

4. Пьюривал Сэмми. Основы разработки веб-приложений (Learning Web App Development). - СПб: Питер, 2015 – 272 с.

5. Дженкинс, С. М. Web-дизайн: научное издание / пер. с англ. . - М. : Эксмо, 2010. - 477 с.

6. Jeremy Keith.  HTML5 For Web Designers. – New-N: A Book Apart

***Дополнительная литература:***

7. [Евсеев Д. А.](http://webirbis.ksu.kz/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=IBIS&P21DBN=IBIS&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%B2%D1%81%D0%B5%D0%B5%D0%B2,%20%D0%94.%20%D0%90.)     Web - дизайн в примерах и задачах [Текст] : учеб. пособие / Д. А. Евсеев, В. В. Трофимов ; под ред. В. В. Трофимова. - М. : КНОРУС, 2010. - 263 с.

8. Основы web - технологий [Текст]: учеб. пособие / П.Б. Храмцов [и др.]. - 2-е изд., испр. - М.: Интернет-Университет Информационных технологий: Бином. Лаборатория Знаний, 2010. - 374 с