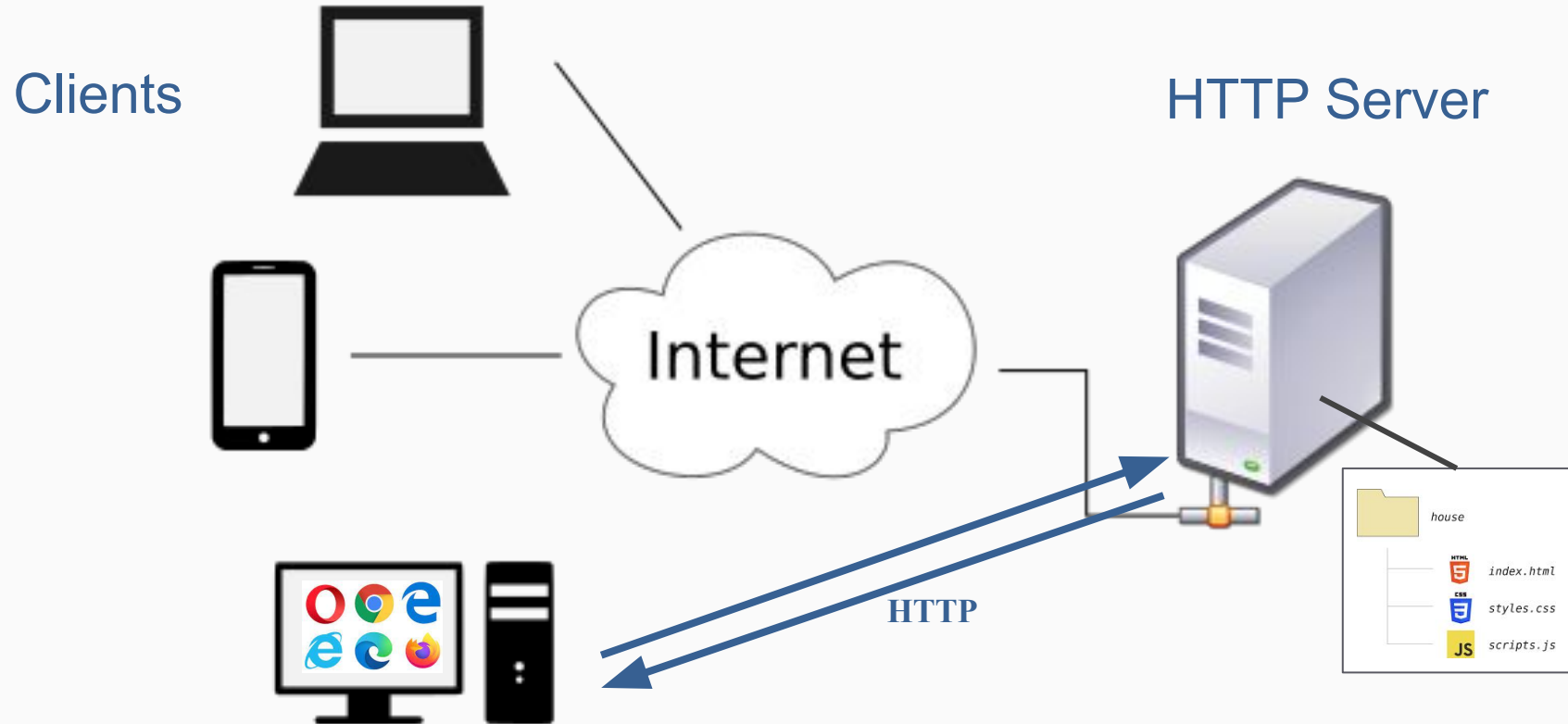


FrontEnd

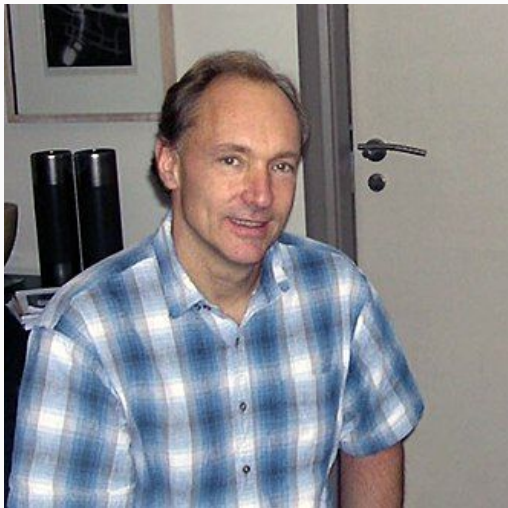
Intro





World Wide Web

World Wide Web (W3) — это инициатива по поиску информации в гипермедиа, имеющая глобальные масштабы и направленная на предоставление универсального доступа к большому количеству документов.



Tim Berners-Lee

This project is experimental and of course comes without any warranty whatsoever. However, it could start a revolution in information access.

Этот проект является экспериментальным и, конечно, поставляется без каких-либо гарантий. Тем не менее, он может начать революцию в доступе к информации.

FrontEnd. Intro

НАУЧНО-ТЕХНИЧЕСКАЯ ПУБЛИЦИСТИКА

В порядке обсуждения

С целью повышения качества публикации научных статей в мае 2011 г. в Вестнике были приняты изменения в редакционные правила. Вестник теперь публикует статьи в формате PDF-файлов. Вестник теперь публикует статьи в формате PDF-файлов. Вестник теперь публикует статьи в формате PDF-файлов.

ПРАВИЛА ПОДГОТОВКИ И ОФОРМЛЕНИЯ

В. С. Петров, Е. Н. Петрова

Публикация научных исследований и обзорных статей в Вестнике осуществляется по следующим правилам:

ВВЕДЕНИЕ

В Вестнике публикуются научные статьи, посвященные различным вопросам науки и техники. Статьи должны быть оригинальными, не опубликованными ранее, и представлять научную ценность. Статьи должны быть написаны на русском языке, иметь четкую структуру и содержать выводы. Статьи должны быть оформлены в соответствии с требованиями Вестника.

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Вестник публикует научные статьи, посвященные различным вопросам науки и техники. Статьи должны быть оригинальными, не опубликованными ранее, и представлять научную ценность.

1.2. Вестник публикует научные статьи, посвященные различным вопросам науки и техники. Статьи должны быть оригинальными, не опубликованными ранее, и представлять научную ценность.

1.3. Вестник публикует научные статьи, посвященные различным вопросам науки и техники. Статьи должны быть оригинальными, не опубликованными ранее, и представлять научную ценность.

1.4. Вестник публикует научные статьи, посвященные различным вопросам науки и техники. Статьи должны быть оригинальными, не опубликованными ранее, и представлять научную ценность.

стается несколько встроенных типов исключений, соответствующих разным типам ошибок, которые могут возникнуть во время выполнения.

Трасировка стека позволяет проследить последовательность вызовов методов, которые привели к ошибке. В качестве примера ниже приведена другая версия предыдущей программы, где вносится та же самая ошибка, но уже не в методе `main()`, а в другом методе.

```
class Excl1 {
    static void subroutine() {
        int d = 0;
        int a = 10 / d;
    }
    public static void main(String args[]) {
        Excl1.subroutine();
    }
}
```

Результат трасировки стека стандартного обработчика исключений отображает весь стек вызовов следующим образом:

```
java.lang.ArithmeticException: / by zero
at Excl1.subroutine(Excl1.java:4)
at Excl1.main(Excl1.java:7)
```

Как видите, на две строки находится седьмая строка кода из метода `main()`, в которой делается вызов метода `subroutine()`, вызвавший исключение при выполнении четвертой строки кода. Трасировкой стека удобно пользоваться для отладки, поскольку ее результат показывает всю последовательность вызовов, приведших к ошибке.

Использование блоков операторов `try` и `catch`

Стандартный обработчик исключений, предоставляемый исполняющей системой Java, безусловно, удобен для отладки, но, как правило, обрабатывать исключения приходится вручную. Это дает два существенных преимущества. Во-первых, появляется возможность исправить ошибку. И во-вторых, предотвращается автоматическое прерывание выполнения программы. Большинство пользователей будут, по меньшей мере, недовольны, если программа будет прерываться и выводить результат трасировки стека всякий раз, когда возникает ошибка. Правда, предотвратить это совсем нетрудно.

Чтобы застраховаться от подобных сбойных ситуаций и организовать обработку ошибок, возникающих во время выполнения, достаточно разместить контролируемый код в блоке оператора `try`. Сразу же за блоком оператора `try` должен следовать блок оператора `catch`, где указывается тип перехватываемого исключения. Для того чтобы показать, насколько просто это делается, в следующий пример программы включены блоки операторов `try` и `catch` для обработки исключения типа `ArithmeticException`, генерируемого при попытке деления на ноль:

```
class Excl2 {
    public static void main(String args[]) {
        int d, a;
```



Википедия
Свободная энциклопедия

Вы не представились системе. О

Статья Обсуждение Читать Текущая версия Править Править

HTML

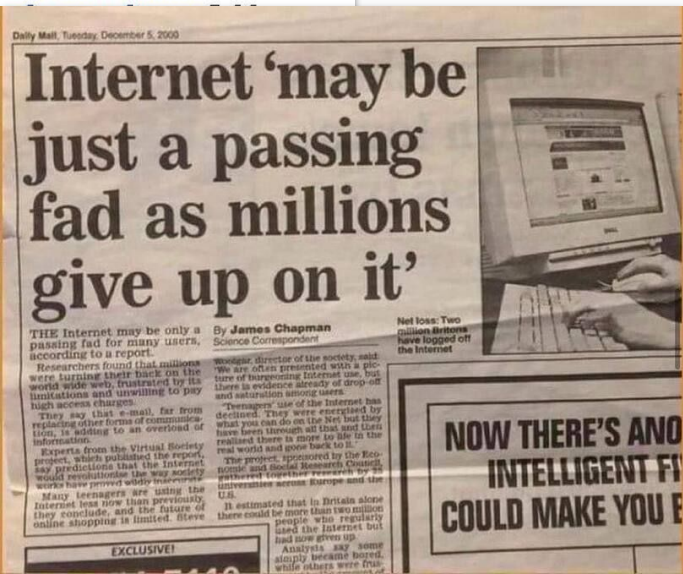
Материал из Википедии — свободной энциклопедии

HTML (от англ. *HyperText Markup Language* — «язык гипертекстовой разметки») — стандартизированный язык гипертекстовой разметки

документов для просмотра в браузере. Браузеры получают HTML документ от сервера или открывают с локального диска файл. Браузер использует интерфейс, который будет отображать документ.

Элементы HTML являются тегами, которые используются для разметки документов. Элементы HTML используются для создания заголовков, абзацев, списков, ссылок и других элементов. Элементы HTML используются для создания гипертекстовых документов.

Элементы HTML используются для создания гипертекстовых документов. Элементы HTML используются для создания гипертекстовых документов. Элементы HTML используются для создания гипертекстовых документов.



*I just had to take the hypertext idea and connect it to the TCP and DNS ideas and — ta-da!— the **World Wide Web**.*

Tim Berners-Lee



WEB PAGE

HTML



СТРУКТУРА И СОДЕРЖАНИЕ

- параграфы
- заголовки
- текст



CSS

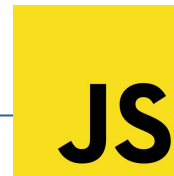


ВНЕШНИЙ ВИД

- шрифт
- цвет
- стиль



JS



ПОВЕДЕНИЕ

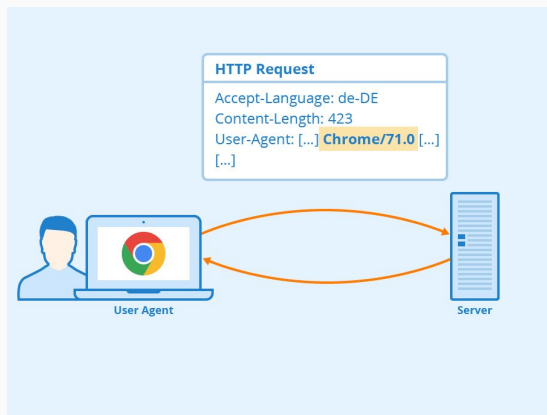
- динамика
- взаимодействие с пользователем



HTML

HTML

(ОТ АНГЛ. ***HYPERTEXT MARKUP LANGUAGE*** — «ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ») — СТАНДАРТИЗИРОВАННЫЙ ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ ДОКУМЕНТОВ ДЛЯ ПРОСМОТРА ВЕБ-СТРАНИЦ



User Agent (пользовательский агент) — это программа (чаще всего веб-браузер), которая взаимодействует с веб-сервером и интерпретирует HTML-код для отображения страниц пользователю.

- браузер
- поисковый бот (например, Googlebot),
- мобильное приложение,
- программа автоматического парсинга (например, curl, Postman).

СЕМАНТИКА — СМЫСЛОВОЕ ЗНАЧЕНИЕ.

HTML-элементы (HTML elements) — основа языка HTML.

Каждый HTML-элемент обозначается начальным (открывающим) и конечным (закрывающим) **тегом (tag)**. Открывающий и закрывающий теги содержат имя элемента.

```
<имя_тега>
```

Содержимое

```
</имя_тега>
```

HTML-элемент



Теги HTML (tags) отмечают начало и конец элемента (и считаются частью элемента).

Имена тегов принято
писать со строчной буквы

HTML-атрибуты (attributes) это специальные слова, которые управляют поведением HTML-элемента. Они добавляют дополнительную функциональность, либо меняют поведение элемента по умолчанию. Атрибуты элемента выражаются внутри начального тега элемента.

```
<p class="info" id="hello"> Hello HTML! </p>
```

атрибут

имя: **class**
значение: **"info"**

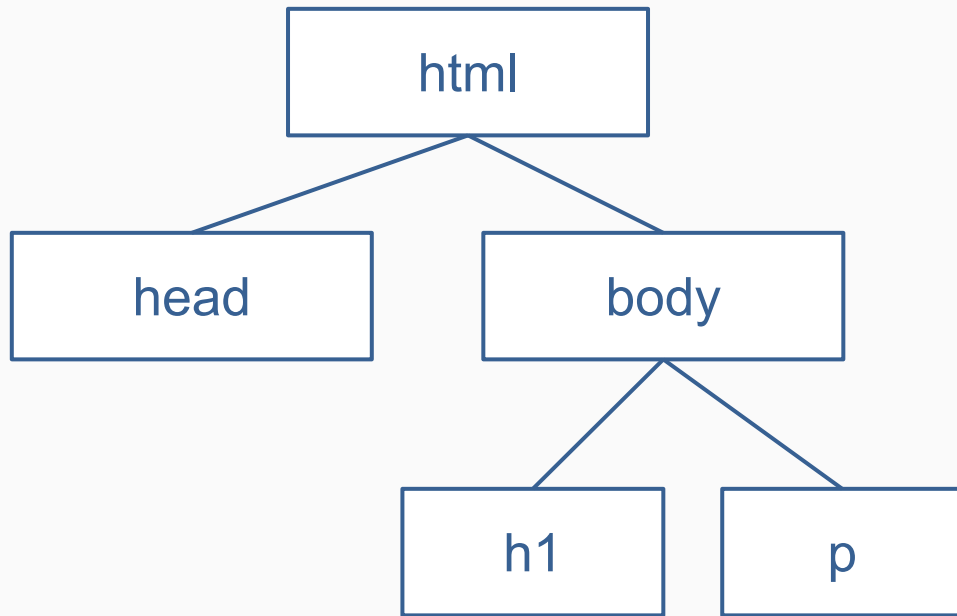
атрибут

имя: **id**
значение: **"hello"**

- Атрибут имеет **имя и значение**.
- Имена и значения атрибутов не чувствительны к регистру, но, рекомендуется использовать нижний регистр.
- Атрибуты выбираются из предопределенного набора возможных атрибутов в зависимости от элемента

HTML

Элементы, находящиеся внутри элемента `<html>`, образуют дерево документа, так называемую **объектную модель документа**, **DOM** (**document object model**). При этом элемент `<html>` является корневым элементом.



HTML

```
<!DOCTYPE html>

<html>

  <head>

    <title>HTML</title>

  </head>

  <body>

    <h1>Введение в HTML</h1>

    <p>

      HTML - стандартизированный язык гипертекстовой
      разметки документов для просмотра веб-страниц

    </p>

  </body>

</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML</title>
  </head>
  <body>
    <h1>Введение в HTML</h1>
    <p>Hello HTML!</p>
  </body>
</html>
```

Элементы HTML «вкладываются» друг в друга. Элемент, который открывается первым, закрывается последним.

При вложении элементов часто используются пробелы и табуляции, чтобы показать уровень вложенности. Однако интервал не требуется и используется только для облегчения чтения HTML людьми.

HTML

Элемент	Описание
<code><!--...--></code>	Используется для добавления комментариев.
<code><!DOCTYPE></code>	Объявляет тип документа и предоставляет основную информацию для браузера — его язык и версия.
<code><html></code> <code></html></code>	Корневой элемент HTML-документа. Является контейнером для всех остальных html-элементов.
<code><head></code> <code></head></code>	Элемент-контейнер для метаданных HTML-документа, таких как <code><title></code> , <code><meta></code> , <code><script></code> , <code><link></code> , <code><style></code> т.е. содержит техническую информацию о странице: заголовок, описание, ключевые слова для поисковых машин, кодировку и т.д.
<code><body></code> <code></body></code>	Представляет тело документа собственно, содержимое.
<code><title></code>	Заголовок HTML-документа, отображаемый в верхней части строки заголовка браузера. Также может отображаться в результатах поиска, поэтому это следует принимать во внимание предоставление названия. Длина заголовка должна быть не более 60 символов

Элемент	Описание
<code><h1> ...<h6></code>	разные уровни заголовков
<code><p></code>	Параграфы в тексте.
<code></code>	Задаёт полужирное начертание отрывка текста, не придавая акцент или важность выделенному.
<code><u></code>	Выделяет отрывок текста подчеркиванием, без дополнительного акцента.
<code><i></code>	Выделяет отрывок текста курсивом, не придавая ему дополнительный акцент.
<code>
</code>	Перенос текста на новую строку.
<code><a></code>	Создаёт гипертекстовые ссылки.
<code></code>	создаёт упорядоченный (нумерованный) список
<code></code>	создаёт неупорядоченный список
<code></code>	элемент списка