



WEB-ПРОГРАММИРОВАНИЕ

HTML & CSS

2 занятие

ДОЛ «Байтик», 2021



По всем

вопросам

Зайцев Даниил Николаевич

dannu2001@gmail.com

ЧТО МЫ СЕГОДНЯ ИЗУЧИМ



01

Введение в веб-разработку

Основные понятия сети интернет

03

Структура HTML-документа

<!DOCTYPE>, <html>, <head>, <body>.



02

HTML

Основные понятия: теги и атрибуты

04

Редакторы кода

Редакторы кода и их характеристики





01

Введение в веб- разработку

Основные понятия сети интернет

История интернета

История интернета начинается в 60-годы XX века. Впервые концепцию описал один американский учёный и эпично назвал её «Галактическая сеть».

В 1969 американское агентство DARPA начало создавать экспериментальную сеть «с коммутацией пакетов». Её назвали ARPANET.

Коммутация пакетов - способ передачи данных по сети. Принцип работы очень простой: делим информацию на маленькие пакеты и отправляем их независимо друг от друга. Это нужно для надёжности, скорости и эффективности.

В декабре 1970 года Network Working Group придумала протокол управления сетью, а в 1971 - 1972 его реализовали в ARPANET. Благодаря этому, появилась возможность создавать сетевые приложения. Первым приложением стала электронная почта, её сделали в 1972-м.

Но это всё научные исследования. Интернет, каким мы его знаем, придумал Тим Бернерс-Ли. Он изобрёл технологии URI/URL, HTTP, и HTML.

URL

Первая важная технология, которая позволила появиться интернету - URL. Она применяется для обозначения адресов почти всех ресурсов Интернета: сайты и сервера приложений вроде электронной почты. URL сочетает в себе две технологии:

*URI - стандарт записи уникального адреса. Например, адрес сайта это URI:
<http://google.com>*

DNS - система доменных имён. У любого устройства в сети есть свой числовой адрес, он называется IP-адрес. Но людям удобнее запоминать буквы - доменные имена. DNS просто помогает найти IP-адрес по доменному имени. Пример: 173.194.44.40 это IP-адрес, а google.com - доменное имя, которое ему соответствует.

HTTP

HyperText Transfer Protocol (гипертекст трансфер протокол) - протокол передачи гипертекста. Ох уж эти учёные, любят называть простые вещи сложными именами. Давай разбираться.

HTTP (эйч-ти-ти-пи) - протокол (то есть регламент, набор условных обозначений), который устанавливает определённый формат общения между клиентом и сервером. Суть его заключается в том, что клиент посылает запросы на URL, а сервер ему отвечает.

Клиент - тот, кто посылает запросы. Когда ты открываешь сайт, ты - клиент.

Сервер - тот, кто принимает запросы и отвечает на них. Когда ты открываешь сайт гугла, сервер гугла посылает тебе веб-страничку - отвечает на твой запрос.

HTTP

А теперь про запросы. Что значит "послать запрос"? Если не углубляться в сложные научные определения, это означает отправить по сети сообщение с каким-нибудь требованием. Это работает так:

1. ты посылаешь запрос
2. между тобой и сервером устанавливается соединение
3. сервер формирует ответ, посылает его тебе
4. соединение закрывается

Вот и всё.

Так вот, протокол передачи гипертекста. "Текста", потому что сервер отвечает на запрос текстом определённого формата. А так как любые данные можно представить в виде текста, то получается, что через HTTP можно пересылать всё: картинки с котиками, гифки с котиками, и даже видео. С котиками.

HTML

HyperText Markup Language (гипертекст маркап лэнгуидж) - язык разметки гипертекста. Опять разбираемся в сложных терминах. В этой аббревиатуре нам интересно слово "разметка".

Разметка - что это вообще такое? Представь, что ты передаёшь текст по сети. Как сделать в тексте заголовков? Выделить абзац? Подчеркнуть слово? Самый простой вариант - пометить начало и конец выделяемого фрагмента условными метками. Например:

```
<заголовок>HTML</заголовок>
```

```
<полужирный>HyperText Markup Language</полужирный> <курсив>(гипертекст маркап лэнгуидж)</курсив> - язык разметки гипертекста.
```

Это разметка.

HTML позволил создавать документы, в которых были заголовки, абзацы, ссылки, таблицы, картинки и куча всего прочего.



Небольшой тест

Что такое коммутация пакетов?

Задание на закрепление, чтобы не забыть.

Select one option from the list

- Модель формирования пакетов сетевых данных
- Способ передачи данных по сети, в котором вся информация передаётся в специальном контейнере - пакете
- Способ передачи данных по сети, в котором сообщение делится на части (пакеты)
- Принцип обработки информации, в котором единицей информации является пакет

Correct answer from **86,710** learners

Total **67%** of tries are correct



Небольшой тест

Что такое коммутация пакетов?

Задание на закрепление, чтобы не забыть.

Select one option from the list

Good job.

Да. "Галактическая сеть", как её назвал Ликлайдер, автор концепции

Correct answer from **86,710** learners

Total **67%** of tries are correct

- Модель формирования пакетов сетевых данных
- Способ передачи данных по сети, в котором вся информация передаётся в специальном контейнере - пакете
- Способ передачи данных по сети, в котором сообщение делится на части (пакеты)
- Принцип обработки информации, в котором единицей информации является пакет



Небольшой тест

Установи соответствия

Повторение – мать учения.

Match two lists

Correct answer from **83,280** learners
Total **47%** of tries are correct

URI

DNS

URL

 позволяет установить соответствие между доменным именем и IP-адресом ^ v

 стандарт записи сетевого адреса ^ v

 отвечает за адресацию в интернете ^ v





Небольшой тест

Установи соответствия

Повторение – мать учения.

Match two lists

✓ Yes!

Correct answer from **83,280** learners

Total **47%** of tries are correct

URI

стандарт записи сетевого адреса

DNS

позволяет установить соответствие между доменным именем и IP-адресом

URL

отвечает за адресацию в интернете



Небольшой тест

Что такое HTTP?

Выбери один правильный вариант.

Select one option from the list

- Это протокол, позволяющий устанавливать соединение между узлами сети
- Это язык программирования для создания сайтов
- Это адрес сайта в интернете
- Это набор правил, который может быть использован для передачи котиков в текстовом виде по сети

Correct answer from **81,751** learners

Total **48%** of tries are correct



Небольшой тест

Что такое HTTP?

Выбери один правильный вариант.

Select one option from the list

Great work!

- Это протокол, позволяющий устанавливать соединение между узлами сети
- Это язык программирования для создания сайтов
- Это адрес сайта в интернете
- Это набор правил, который может быть использован для передачи котиков в текстовом виде по сети

Correct answer from **81,751** learners

Total **48%** of tries are correct



Основные понятия: теги и атрибуты

02

HTML

Теперь мы знаем, что такое разметка, давай узнаем, что такое HTML. Есть вещи, которые проще показать, чем объяснить. HTML - одна из них.

Всё гениальное просто. Вот как бы выглядел текст этого урока на HTML:

```
<h1>HTML</h1>
```

```
<p>Теперь мы знаем, что такое <i>разметка</i>, давай узнаем, что такое HTML. Есть вещи, которые проще показать, чем объяснять. HTML - одна из них.</p>
```

```
<p>Всё гениальное просто. Вот как бы выглядел текст этого урока на HTML:</p>
```

h1 - заголовок первого уровня (header 1)

p - абзац (paragraph)

ЗАДАНИЕ

Разметь текст

Война и мир

✦ чтобы он стал заголовком первого уровня

Теги

Как ты уже догадался, в HTML для разметки используется особый набор символов. Он называется тег.

Что такое тег

Тег – это синтаксическая единица языка HTML, которая выделяет или создаёт элемент. Это набор символов, с помощью которого браузер понимает, где элемент создается, начинается и заканчивается. Есть 2 вида тегов: двойные и одинарные.

Двойные теги

Двойные теги показывает начало и конец элемента. Начало элемента обозначается открывающим тегом `<...>`, а конец - закрывающим `</...>`.

Двойной тег обязательно должен быть закрыт. Даже несмотря на то, что современные браузеры умеют в некоторых случаях понимать разметку без закрытых тегов, лучше всегда закрывать их.

Одинарные теги

Одинарные теги просто не имеют пары. Примеры: тег переноса строки `
` или горизонтальной линии `<hr>`.

Старые браузеры требовали закрывать одинарные теги: `
`, сейчас таких браузеров практически не осталось и допустимо использовать оба варианта синтаксиса.

Вложенность тегов

Теги можно вкладывать друг в друга. Пример:

```
<p>  
  <em>Курсив внутри абзаца</em>  
</p>
```

Но при вложении тега нужно всегда помнить о том, что внутренний тег нужно закрывать раньше внешнего. То есть вот такой код недопустим:

```
<p><em>Я уже говорил тебе, что такое безумие?</p></em>
```

Небольшой тест

Выбери примеры корректной разметки

Обрати внимание на выделенный текст в предыдущем шаге.

Select all correct options from the list

- `<p> – Случалось видеть сон, казавшийся реальностью?</p>`
- `<p>Нет на свете мук сильнее муки слова: <bg> Тщетно с уст порой безумный рвется крик,</p>`
- `<h1>Давным-давно....<h1>`

Correct answer from **65,416** learners

Total **41%** of tries are correct

Небольшой тест

Выбери примеры корректной разметки

Обрати внимание на выделенный текст в предыдущем шаге.

Select all correct options from the list

Totally right.

Correct answer from **65,416** learners

Total **41%** of tries are correct

`<p> – Случалось видеть сон, казавшийся реальностью?</p>`

`<p>Нет на свете мук сильнее муки слова: <bг> Тщетно с уст порой безумный рвется крик,</p>`

`<h1>Давным-давно....<h1>`

Небольшой тест

Выбери примеры корректной разметки

Тяжело в учении – легко в бою.

Select all correct options from the list

- `<p>Lorem ipsum</h1> dolor sit amet </p>`
- `И нас с тобой не отыскать.
<i>Все, что мы есть, – вода</i>`
- `<i>Never gonna give you up
Never gonna let you down
Never gonna run around and desert you</i>`
- `Quick brown fox<i>jumps over the lazy dog</i>`

Correct answer from **62,974** learners

Total **30%** of tries are correct

Небольшой тест

Выбери примеры корректной разметки

Тяжело в учении – легко в бою.

Select all correct options from the list

Totally right.

You just solved a difficult problem, congratulations! You can [help others](#) in the comments below.

Correct answer from **62,974** learners

Total **30%** of tries are correct

- `<p>Lorem ipsum</h1> dolor sit amet </p>`
- `И нас с тобой не отыскать.
<i>Все, что мы есть, – вода</i>`
- `<i>Never gonna give you up
Never gonna let you down
Never gonna run around and desert you</i>`
- `Quick brown fox<i>jumps over the lazy dog</i>`

Атрибуты

Атрибуты — это свойства тега. С помощью них мы задаём параметры тега.

Сразу возьмём пример: тег `<a>` — ссылка. Для задания адреса, куда будет вести эта ссылка, нам понадобится атрибут `href`. Вот так будет выглядеть ссылка на страницу Вконтакте:

```
<a href="https://vk.com">Вконтакте</a>
```

Атрибут указывается внутри тега, значение атрибута указывается внутри кавычек. Атрибуты отделяются друг от друга пробелами. Пример ссылки на страницу, которая откроется в новой вкладке:

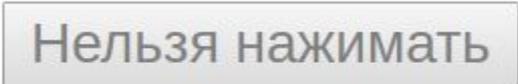
```
<a href="https://vk.com" target="_blank">Вконтакте</a>
```

Атрибуты

У атрибута может не быть значения, тогда наличие атрибута включает какой-то параметр, а отсутствие - отключает. Например, атрибут `disabled`. Если кнопке `<button>` задать атрибут `disabled`, она станет серой и на неё невозможно будет нажать.

`<button disabled>Нельзя нажимать</button>`

Результат:



Нельзя нажимать



Небольшой тест

Выбери верные утверждения

Без труда не выловишь и рыбку из пруда.

Select all correct options from the list

- атрибут обязательно должен иметь какое-то значение
- атрибуты - необязательные свойства тега

Correct answer from **62,052** learners

Total **65%** of tries are correct



Небольшой тест

Выбери верные утверждения

Без труда не выловишь и рыбку из пруда.

Select all correct options from the list

You are right, well done!

атрибут обязательно должен иметь какое-то значение

атрибуты - необязательные свойства тега

Correct answer from **62,052** learners

Total **65%** of tries are correct

Особенности интерпретации HTML

При преобразовании HTML-кода в веб-страничку есть некоторые особенности, в которых мы сейчас разберёмся.

Перенос строки только через тег

Возможно, у тебя возник вопрос, зачем нужен тег переноса строки, если можно просто нажать энтер. Дело в том, что HTML воспринимает перенос строки как пробел. Это нужно потому, что редакторы кода не переносят строки, которые не помещаются в экран - так удобнее писать код. Поэтому чтобы длинный текст влезал в экран, в коде ставятся переносы строки, которые не нужны, когда страница показывается в браузере.

Несколько пробелов, идущих подряд, считаются за один

Это происходит по той же причине, что и с переносом строки. Так просто удобнее форматировать код в редакторе. Из-за того, что теги вкладываются друг в друга, для удобного восприятия кода вложенность показывают отступами - пробелами. Пример:

```
<article>
  <h1>В Индонезии после землетрясения началось извержение вулкана</h1>
  <p>
    В Индонезии на острове
    Сулавеси проснулся вулкан Сопутан,
    выпустив столб пепла высотой около четырех метров, пишут
    «Новые Известия». Извержение вулкана началось сегодня утром.
    Местным жителям рекомендовали не приближаться к нему и
    использовать респираторы.
  </p>
  <p>
    Подземные толчки стали причиной возникновения цунами,
    высота волн достигала 7 м. По последним данным, в результате
    стихийных бедствий погибли более 1400 человек. На оказание
    помощи пострадавшим от землетрясения в Индонезии ООН выделила
    $15 млн.
  </p>
</article>
```

Произвольный регистр

`
` даст такой же результат, что и `
`, и `
`, и `
`.

Несмотря на это, писать разметку лучше в нижнем регистре - это негласное правило.

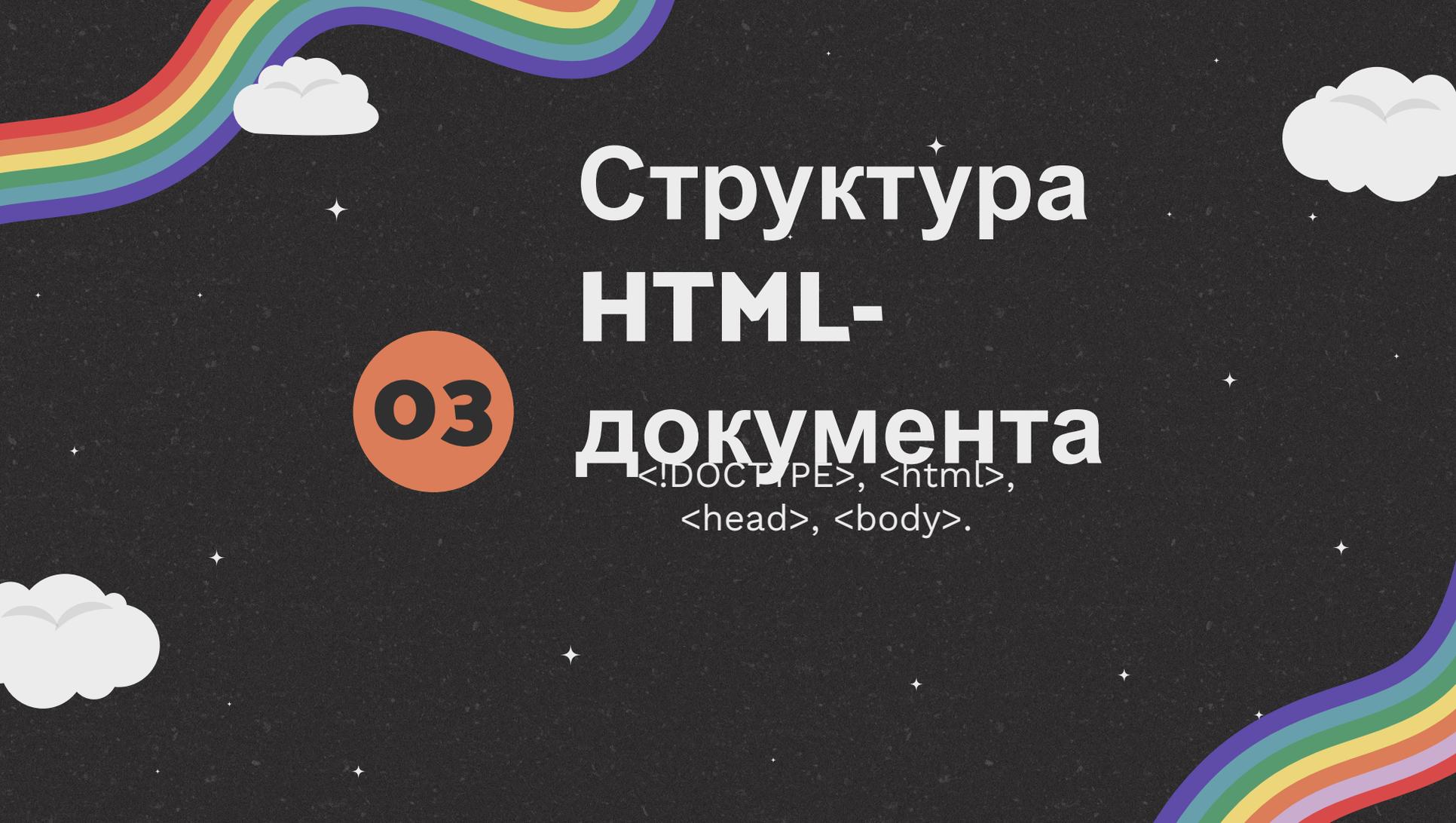
Перенос строки в теге

При определении тега и его атрибутов можно переносить строку. Это полезно для длинных определений.

Например, для этого изображения:

```

```



03

Структура HTML- документа

`<!DOCTYPE>`, `<html>`,
`<head>`, `<body>`.

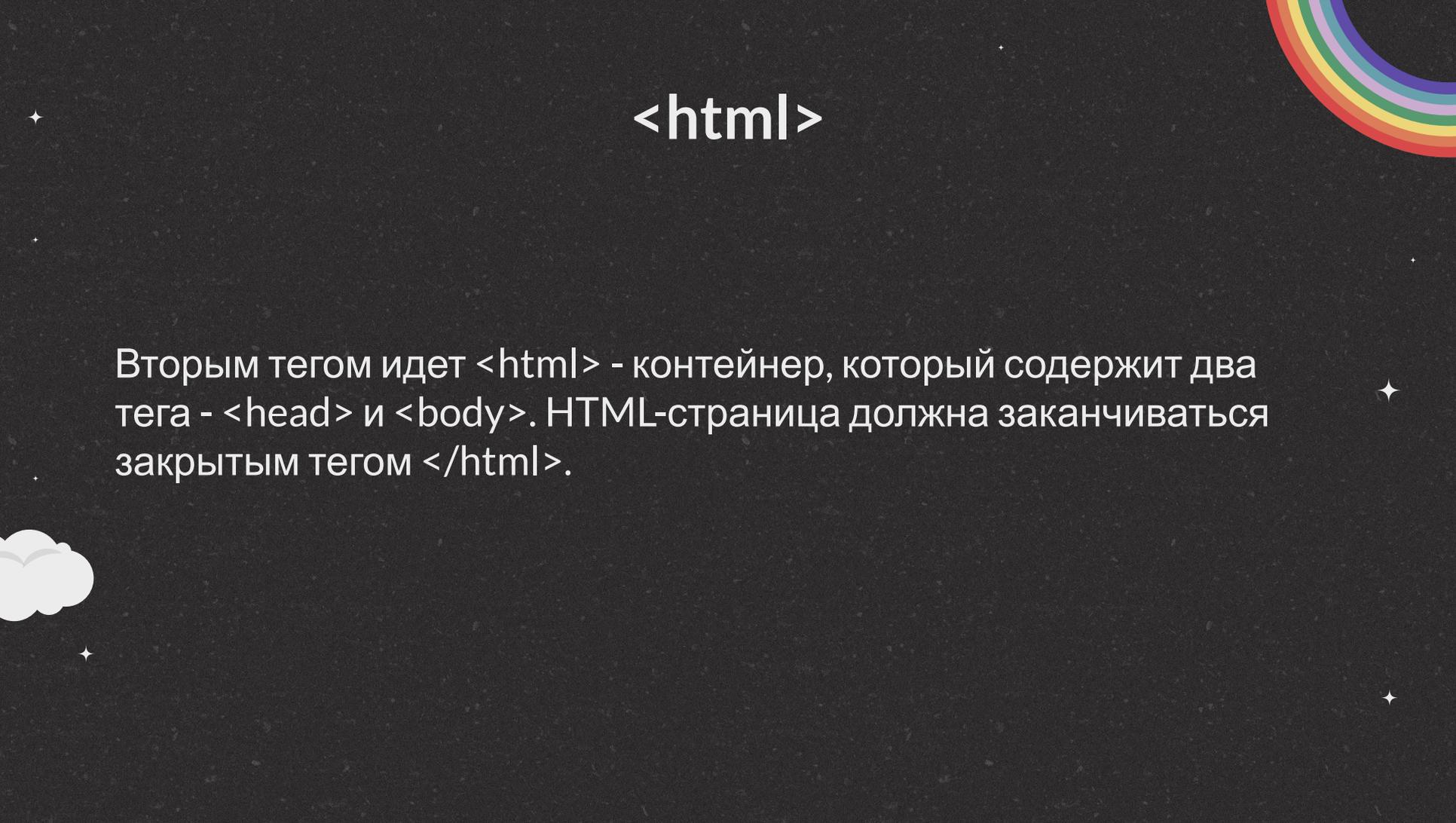
Структура HTML-документа

Структура HTML документа - скелет, на основе которого строится вся страница:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Страница</title>
  </head>
  <body>
    <h1>...</h1>
    <p>...</p>
  </body>
</html>
```

<!DOCTYPE>

Первым тегом в любом HTML документе должен идти тег <!DOCTYPE>. Он говорит браузеру, по какому стандарту написана страница. На рассвете веба HTML существовал в разных несовместимых версиях, поэтому для их одновременной поддержки нужно было указывать версию явно. Сейчас все пришли к одному стандарту - HTML5. Поэтому для всех сайтов, которые создаются сегодня, нужно указывать <!DOCTYPE html> - так обозначается HTML5.



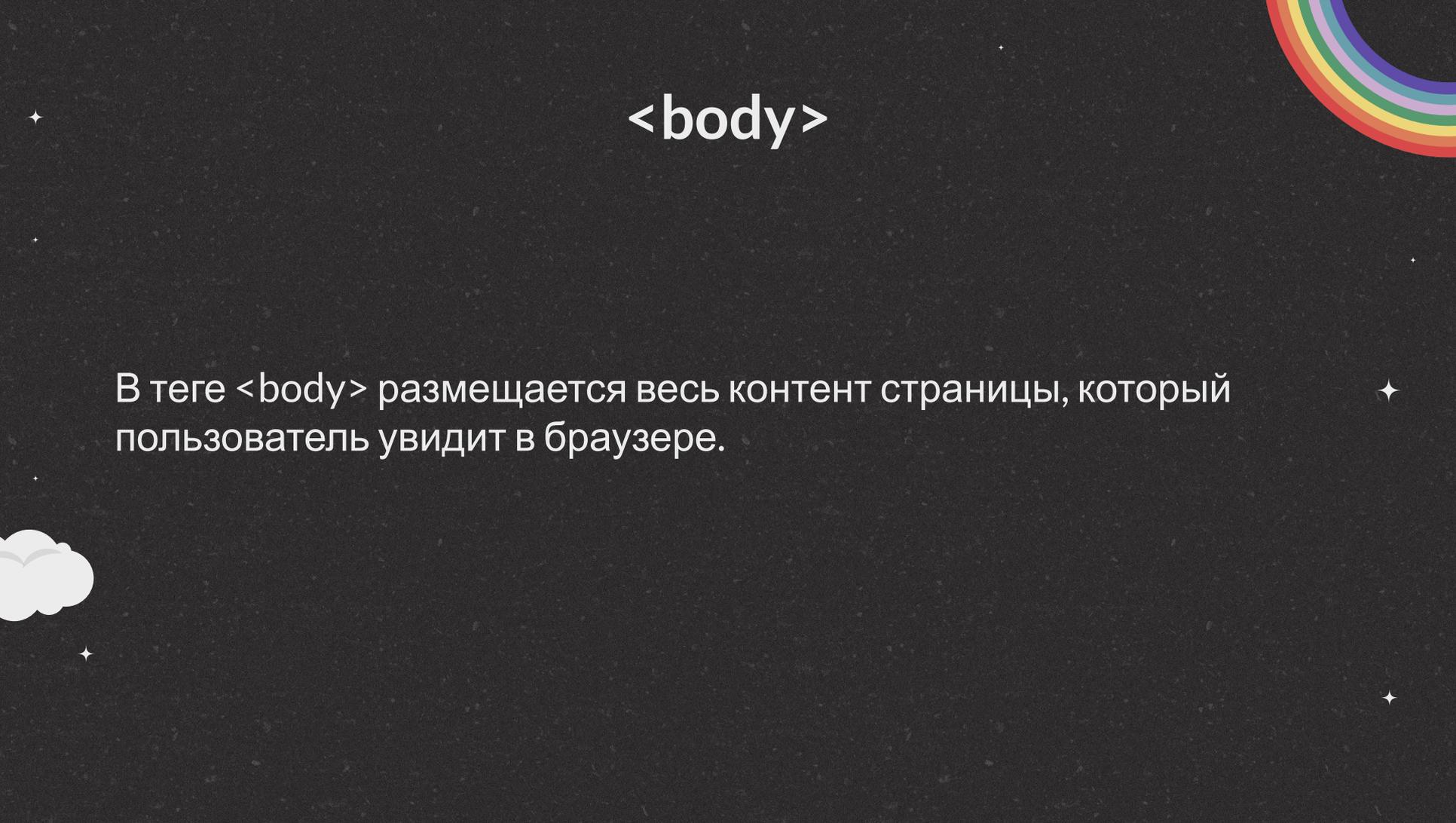
`<html>`

Вторым тегом идет `<html>` - контейнер, который содержит два тега - `<head>` и `<body>`. HTML-страница должна заканчиваться закрытым тегом `</html>`.

<head>

В теге <head> хранится информация о странице. Здесь указывают кодировку <meta charset="...">, имя страницы <title>...</title>, специальную информацию для поисковиков, а ещё тут подключаются стилевые файлы и скрипты.

 Тег <head> не отображается. Его цель — сказать браузеру информацию о странице.



`<body>`

В теге `<body>` размещается весь контент страницы, который пользователь увидит в браузере.



Небольшой тест

Тест на структуру страницы

Соотнеси теги `head`, `body`, `html`, `doctype` с их порядком объявления в HTML-документе.

Correct answer from **54,883** learners

Total **80%** of tries are correct

Match two lists

1234^ v^ v^ v^ v

Небольшой тест

Тест на структуру страницы

Соотнеси теги `head`, `body`, `html`, `doctype` с их порядком объявления в HTML-документе.

Match two lists



Totally right.

Correct answer from **54,883** learners

Total **80%** of tries are correct

1

`<!DOCTYPE>`

2

`<html>`

3

`<head>`

4

`<body>`

Небольшой тест

Тест на структуру страницы

Выбери один или несколько вариантов с верными суждениями.

Это задание на внимательность, потому что разработчик должен быть внимательным.

Select all correct options from the list

- В теге `<body>` размещается весь видимый контент страницы
- В теге `<head>` прописывается меню и шапка сайта
- Стандарт HTML5 обозначается с помощью тега `<DOCTYPE html>`
- Тег `<html>` - это контейнер, которой показывает браузеру на каком языке написан код.

Correct answer from **54,049** learners

Total **18%** of tries are correct

Небольшой тест

Тест на структуру страницы

Выбери один или несколько вариантов с верными суждениями.

Это задание на внимательность, потому что разработчик должен быть внимательным.

Select all correct options from the list

All is correct.

You just solved a difficult problem, congratulations! You can [help others](#) in the comments below.

Correct answer from **54,049** learners
Total **18%** of tries are correct

- В теге `<body>` размещается весь видимый контент страницы
- В теге `<head>` прописывается меню и шапка сайта
- Стандарт HTML5 обозначается с помощью тега `<DOCTYPE html>`
- Тег `<html>` - это контейнер, которой показывает браузеру на каком языке написан код.

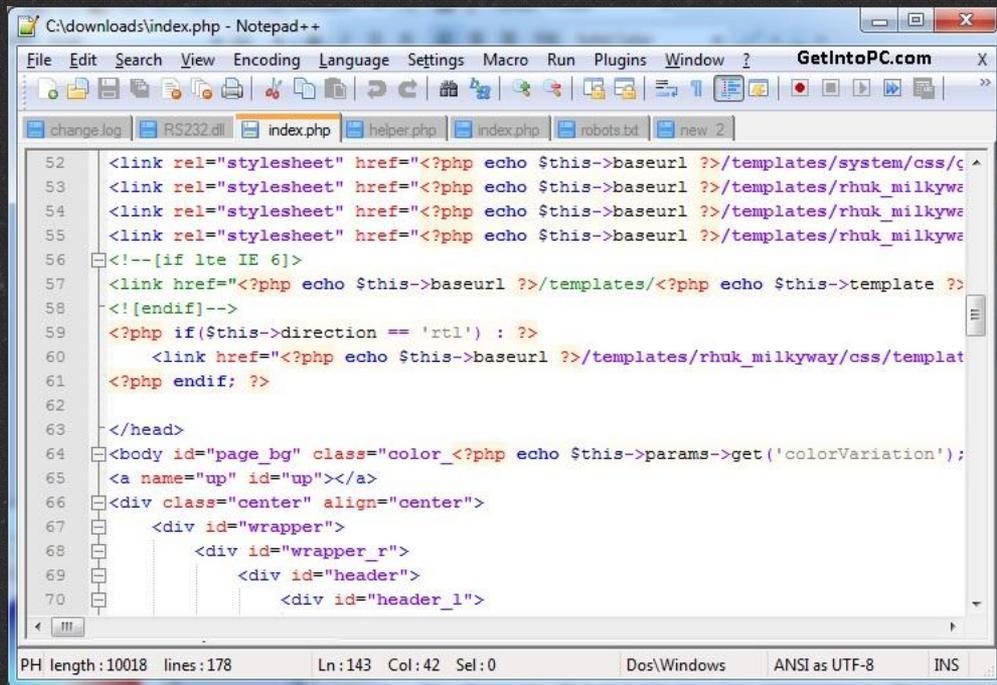


04

Редакторы кода

Редакторы кода и их характеристики

Notepad++

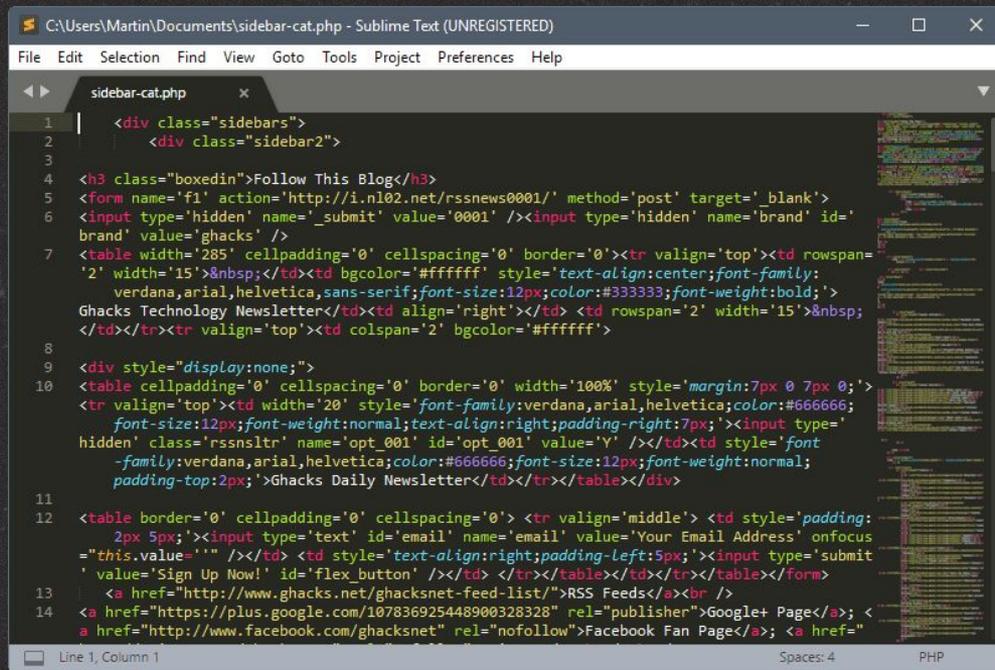


The screenshot shows the Notepad++ application window with the file 'C:\downloads\index.php' open. The code is PHP code for a web page header, featuring several `<link>` tags for stylesheets and a conditional comment for Internet Explorer 6. The code is color-coded: PHP tags are blue, HTML tags are green, and text is black. The status bar at the bottom indicates 'PH length: 10018 lines: 178' and 'Ln: 143 Col: 42 Sel: 0'.

```
52 <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/system/css/c
53 <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/rhuk_milkywa
54 <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/rhuk_milkywa
55 <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/rhuk_milkywa
56 <!--[if lte IE 6]>
57 <link href="<?php echo $this->baseurl ?>/templates/<?php echo $this->template ?>
58 <![endif]-->
59 <?php if($this->direction == 'rtl') : ?>
60 <link href="<?php echo $this->baseurl ?>/templates/rhuk_milkyway/css/templat
61 <?php endif; ?>
62
63 </head>
64 <body id="page_bg" class="color_<?php echo $this->params->get('colorVariation');
65 <a name="up" id="up"></a>
66 <div class="center" align="center">
67 <div id="wrapper">
68 <div id="wrapper_r">
69 <div id="header">
70 <div id="header_1">
```

Боевая классика. Ветеран среди редакторов кода, считался самым популярным у веб-разработчиков. Сегодня его в основном используют ностальгирующие консерваторы.

Sublime Text



```
1 <div class="sidebars">
2   <div class="sidebar2">
3
4   <h3 class="boxedin">Follow This Blog</h3>
5   <form name='f1' action='http://i.n102.net/rssnews0001/' method='post' target='_blank'>
6     <input type='hidden' name='_submit' value='0001' /><input type='hidden' name='brand' id='
7     brand' value='ghacks' />
8     <table width='285' cellpadding='0' cellspacing='0' border='0'><tr valign='top'><td rowspan=
9     '2' width='15'>&nbsp;</td><td bgcolor='#ffffff' style='text-align:center;font-family:
10     verdana,arial,Helvetica,sans-serif;font-size:12px;color:#333333;font-weight:bold;'>
11     Ghacks Technology Newsletter</td><td align='right'></td> <td rowspan='2' width='15'>&nbsp;<
12     /td></tr><tr valign='top'><td colspan='2' bgcolor='#ffffff'>
13
14   <div style="display:none;">
15   <table cellpadding='0' cellspacing='0' border='0' width='100%' style='margin:7px 0 7px 0;'>
16     <tr valign='top'><td width='20' style='font-family:verdana,arial,Helvetica;color:#666666;
17     font-size:12px;font-weight:normal;text-align:right;padding-right:7px;'><input type='
18     hidden' class='rssnsltr' name='opt_001' id='opt_001' value='Y' /></td><td style='font
19     -family:verdana,arial,Helvetica;color:#666666;font-size:12px;font-weight:normal;
20     padding-top:2px;'>Ghacks Daily Newsletter</td></tr></table></div>
21
22   <table border='0' cellpadding='0' cellspacing='0'> <tr valign='middle'> <td style='padding:
23     2px 5px;'><input type='text' id='email' name='email' value='Your Email Address' onfocus
24     ="this.value="" /></td> <td style='text-align:right;padding-left:5px;'><input type='submit
25     ' value='Sign Up Now!' id='flex_button' /></td> </tr></table></td></tr></table></form>
26
27   <a href="http://www.ghacks.net/ghacksnet-feed-list/">RSS Feeds</a><br />
28   <a href="https://plus.google.com/107836925448900328328" rel="publisher">Google+ Page</a>; <
29   a href="http://www.facebook.com/ghacksnet" rel="nofollow">Facebook Fan Page</a>; <a href="
```

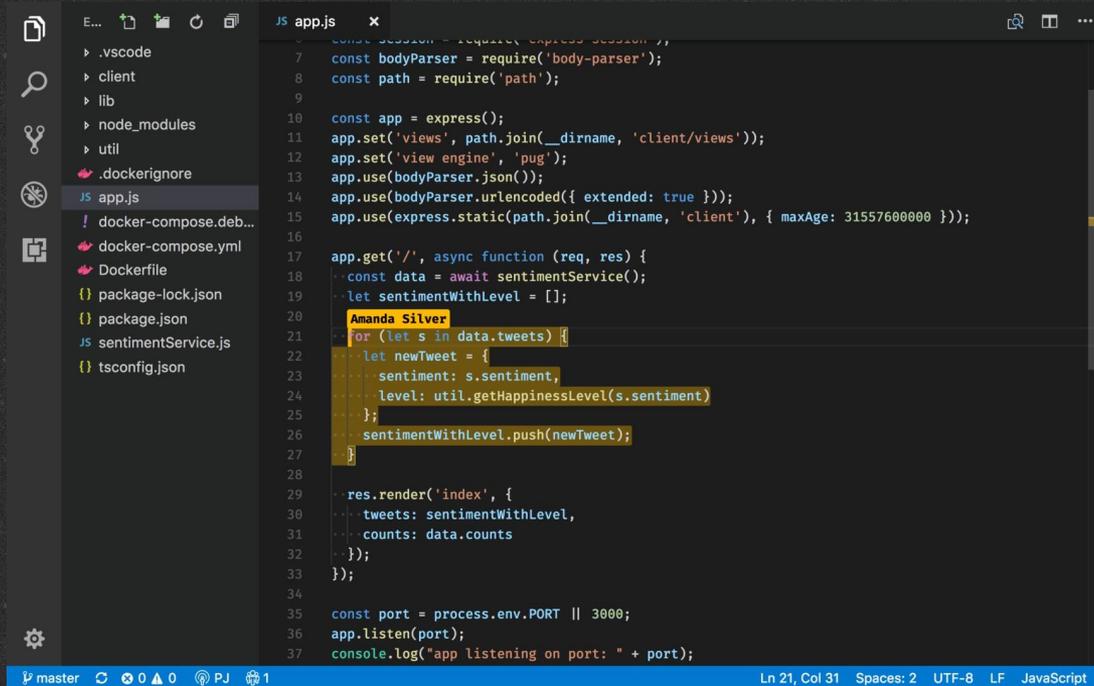
Довольно быстро работает, неплохо выглядит и кастомизируется, имеет несколько полезных плагинов. В целом неплох, но для веб-разработчика есть более подходящий софт. Рекомендуется использовать его только если у тебя слабый компьютер.

Atom

```
54 handleEvents: =>
55   @on 'mousedown', '.atom-chat-resize-handle', (e) => @resizeStarted(e)
56   @on 'keyup', '.chat-input .editor', (e) => @enterPressed(e)
57   @on 'click', '.chat-room', => @roomsClicked()
58   @on 'click', '.add-room', => @addRoomClicked()
59
60   @subscriptions.add atom.config.onDidChange 'atom-chat.showOnRightSide', ({newValue}) =>
61     @onSideToggled(newValue)
62
63   @subscriptions.add atom.config.onDidChange 'atom-chat.username', ({newValue}) =>
64     if newValue is "User"
65       @username = newValue+@uuid
66     else
67       @username = newValue
68     socket.emit 'atom:username', @username
69
70 showOnline: (online)->
71   @toolTipDisposable?.dispose()
72   if online > 0
73     @title.html('Atom Chat ('+online+')')
74     title = "#{_pluralize(online, 'user')} online"
75   else
76     title = "Nobody online"
77   @title.html('Atom Chat')
78   @toolTipDisposable = atom.tooltips.add @title, title: title
79
80 onSideToggled: (newValue) =>
81   @element.dataset.showOnRightSide = newValue
82   if @isVisible()
83     @detach()
84     @attach()
85
```

Хороший редактор кода, заточенный под веб-разработку. Много тем оформления, плагинов. Работает на веб-технологиях, поэтому если ты планируешь развиваться дальше и изучать JavaScript, то в дальнейшем сможешь писать свои расширения. Его минус - скорость работы.

Visual Studio Code



```
1 const session = require('express-session');
2 const bodyParser = require('body-parser');
3 const path = require('path');
4
5
6
7
8
9
10 const app = express();
11 app.set('views', path.join(__dirname, 'client/views'));
12 app.set('view engine', 'pug');
13 app.use(bodyParser.json());
14 app.use(bodyParser.urlencoded({ extended: true }));
15 app.use(express.static(path.join(__dirname, 'client'), { maxAge: 31557600000 }));
16
17 app.get('/', async function (req, res) {
18   const data = await sentimentService();
19   let sentimentWithLevel = [];
20
21   for (let s in data.tweets) {
22     let newTweet = {
23       sentiment: s.sentiment,
24       level: util.getHappinessLevel(s.sentiment)
25     };
26     sentimentWithLevel.push(newTweet);
27   }
28
29   res.render('index', {
30     tweets: sentimentWithLevel,
31     counts: data.counts
32   });
33 });
34
35 const port = process.env.PORT || 3000;
36 app.listen(port);
37 console.log("app listening on port: " + port);
```

Не путай с Visual Studio. Редактор кода для веба от Microsoft. По сути, это более быстрый аналог Atom. Он имеет все те же самые плюсы, что и Atom, но работает ощутимо быстрее.

WebStorm

```
1 import React, {Component, PropTypes} from 'react';
2 import {View, TextInput} from 'react-native';
3 import styles from './issue-summary.styles';
4 import MultilineInput from '../multiline-input/multiline-input';
5
6 export default class AttachmentsRow extends Component {
7
8   static propTypes = {"editable": PropTypes.bool...}
9
10  render() {
11    const {editable, showSeparator, summary, description, ...rest} = this.props;
12
13    return (
14      <View {...rest}>
15        <Te
16          TextInput      ReactNative (react-native.js, react-native)
17          CreateIssue   (create-issue.js, src/views/create-issue)
18          EnterServer   (enter-server.js, src/views/enter-server)
19          ^↓ and ^↑ will move caret down and up in the editor >>
20          editable={editable}
21          autoCapitalize="sentences"
22          multiline={true}
23          underlineColorAndroid="transparent"
24          placeholder="Description"
25          value={description}
26          onChangeText={this.props.onDescriptionChange} />
27        </View>
28      );
29    }
30  }
31
32
33
34
35
36
37
38
39
```

Единственная IDE в нашем списке. JetBrains WebStorm — интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA. WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Подходит для более опытных разработчиков, которые занимаются разработкой веб-приложений.



СПАСИБО!



Пожалуйста, ваши вопросы

dannu2001@gmail.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**



WEB-ПРОГРАММИРОВАНИЕ

HTML & CSS

3 занятие

ДОЛ «Байтик», 2021



По всем

вопросам

Зайцев Даниил Николаевич

dannu2001@gmail.com

ЧТО МЫ СЕГОДНЯ ИЗУЧИМ

01

Элементы и их виды

02

Списки

03

Изображения и видео

04

Адреса, ссылки и
якоря



01

Элементы и их виды

Элементы и их виды

Элементы - то, что создаётся тегами. Можно сказать, что теги это текстовое представление элементов. Элементы бывают двух видов: блочные и строчные

Блочные элементы

Составляют структуру страницы.

Особенности:

блоки располагаются друг под другом по вертикали
запрещено вставлять блочный элемент внутрь строчного
занимают всё допустимое пространство по ширине
высота вычисляется автоматически, исходя из содержимого

Примеры:

абзацы `<p>`

списки: маркированные (с маркером) `` и нумерованные (с числами) ``

заголовки: от первого уровня `<h1>` до шестого уровня `<h6>`

статьи `<article>`

разделы `<section>`

длинные цитаты `<blockquote>`

блоки общего назначения `<div>`

Строчные элементы

Используются для форматирования текстовых фрагментов. Обычно содержат одно или несколько слов.

Особенности:

элементы, идущие подряд, располагаются на одной строке и переносятся на другую при необходимости
внутри допустимо вставлять текст или другие строчные элементы, помещать блочные элементы – запрещено

Примеры:

ссылки `<a>`

выделенные слова ``

важные слова ``

короткие цитаты `<q>`

аббревиатуры `<abbr>`



**Если ты запомнишь только одну вещь
из этого раздела, запомни, что:**

**блочные элементы занимают всё доступное
пространство по ширине**



**строчные элементы ведут себя как текст -
выстраиваются в ряд по горизонтали и
переносятся на следующую строчку, если не
хватает места**

Небольшой тест

Выбери правильные суждения

Это задание на внимательность, потому что разработчик должен быть внимательным

Correct answer from **48,696** learners

Total **36%** of tries are correct

Select all correct options from the list

- По умолчанию `<p>`, `<div>`, `<a>` - блочные элементы
- Заголовок (`h1-h6`) по умолчанию является строчным элементом
- По умолчанию ``, ``, `<blockquote>` - строчные элементы
- Строчные элементы могут располагаться в одной строке (по горизонтали)
- Блочные элементы занимают всё доступное пространство по ширине

Небольшой тест

Выбери правильные суждения

Это задание на внимательность, потому что разработчик должен быть внимательным

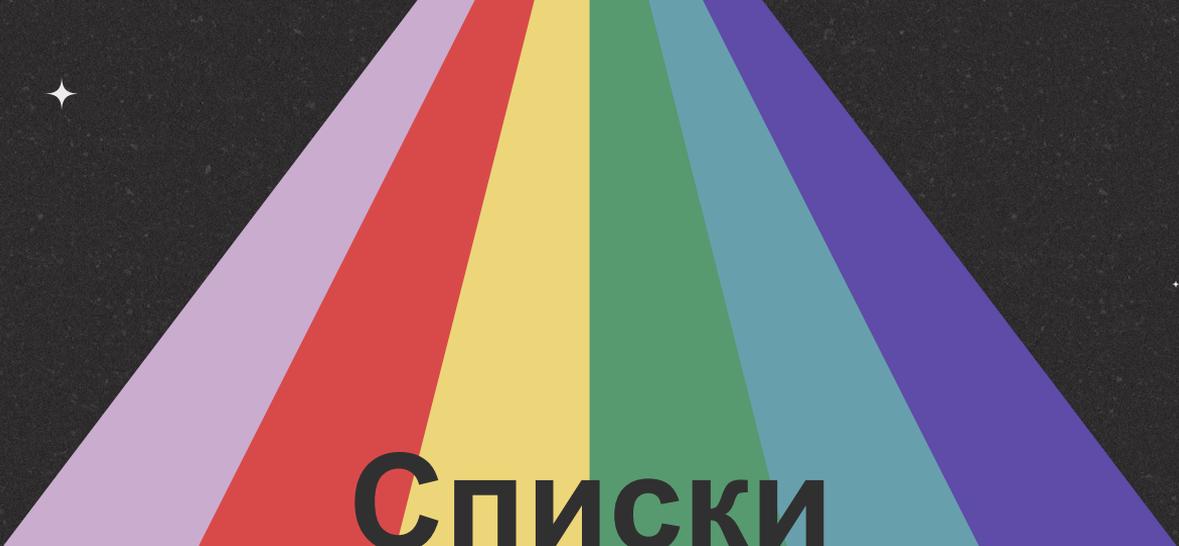
Select all correct options from the list

Well done!

Correct answer from **48,696** learners

Total **36%** of tries are correct

- По умолчанию `<p>`, `<div>`, `<a>` - блочные элементы
- Заголовок (h1-h6) по умолчанию является строчным элементом
- По умолчанию ``, ``, `<blockquote>` - строчные элементы
- Строчные элементы могут располагаться в одной строке (по горизонтали)
- Блочные элементы занимают всё доступное пространство по ширине



СПИСКИ



02

Списки

В HTML существует три вида списков:

1. Маркированный
2. Нумерованный
3. Список определений

Маркированный

Список из неупорядоченных элементов.

Состоит из двух тегов:

`` (unordered list) - тег начала и конца списка

`` (list item) - пункт списка

Пример:

Список ингредиентов:

```
<ul>
  <li>Картошка</li>
  <li>Морковка</li>
  <li>Свекла</li>
</ul>
```

Нумерованный

Упорядоченный список, каждый пункт имеет свой номер.

Состоит из двух тегов:

`` (ordered list) - тег начала и конца списка

`` (list item) - пункт списка

Пример:

Инструкция по приготовлению:

```
<ol>  
  <li>Довести воду до кипения</li>  
  <li>Засыпать ингредиенты</li>  
  <li>Варить 10 минут</li>  
</ol>
```

Список определений

Используются для создания списка терминов и их определений.
В общем случае, каждый пункт — это пара "имя/значение".

Состоит из трёх тегов:

`<dl>` (description list) - тег начала и конца списка

`<dt>` (term) - термин

`<dd>` (description) – определение

Пример:

`<dl>`

`<dt>Гаспачо</dt><dd>лёгкий холодный суп из перетёртых в пюре свежих овощей</dd>`

`<dt>Том-ям</dt><dd>кисло-острый суп на основе куриного бульона с креветками, курицей, рыбой или другими морепродуктами</dd>`

`<dt>Борщ</dt><dd>разновидность супа на основе свёклы, которая придаёт борщу характерный красный цвет</dd>`

`</dl>`

Небольшой тест

Установи соответствия

Иногда прочитанное сразу же забывается. Уверены, сейчас не тот случай.

Correct answer from **47,952** learners

Total **73%** of tries are correct

Match two lists

ol
ul
dl
Нумерованный список
Маркированный список
Список определений



Небольшой тест

Установи соответствия

Иногда прочитанное сразу же забывается. Уверены, сейчас не тот случай.

Match two lists

✓ Right.

Correct answer from **47,952** learners

Total **73%** of tries are correct

ol

Нумерованный список

ul

Маркированный список

dl

Список определений



Вложенные списки

Списки можно вкладывать друг в друга. Для этого в элемент списка вставляется ещё один список. Можно вкладывать списки любого вида друг в друга в любой последовательности.

Пример:

```
<ul>
  <li>Омлет
    <ul>
      <li>Молоко</li>
      <li>Яйца</li>
    </ul>
  </li>
  <li>Блинчики
    <ul>
      <li>Молоко</li>
      <li>Яйца</li>
      <li>Мука</li>
      <li>Сахар</li>
      <li>Соль</li>
      <li>Разрыхлитель</li>
    </ul>
  </li>
  <li>Торт</li>
  <li>Пирог</li>
</ul>
```

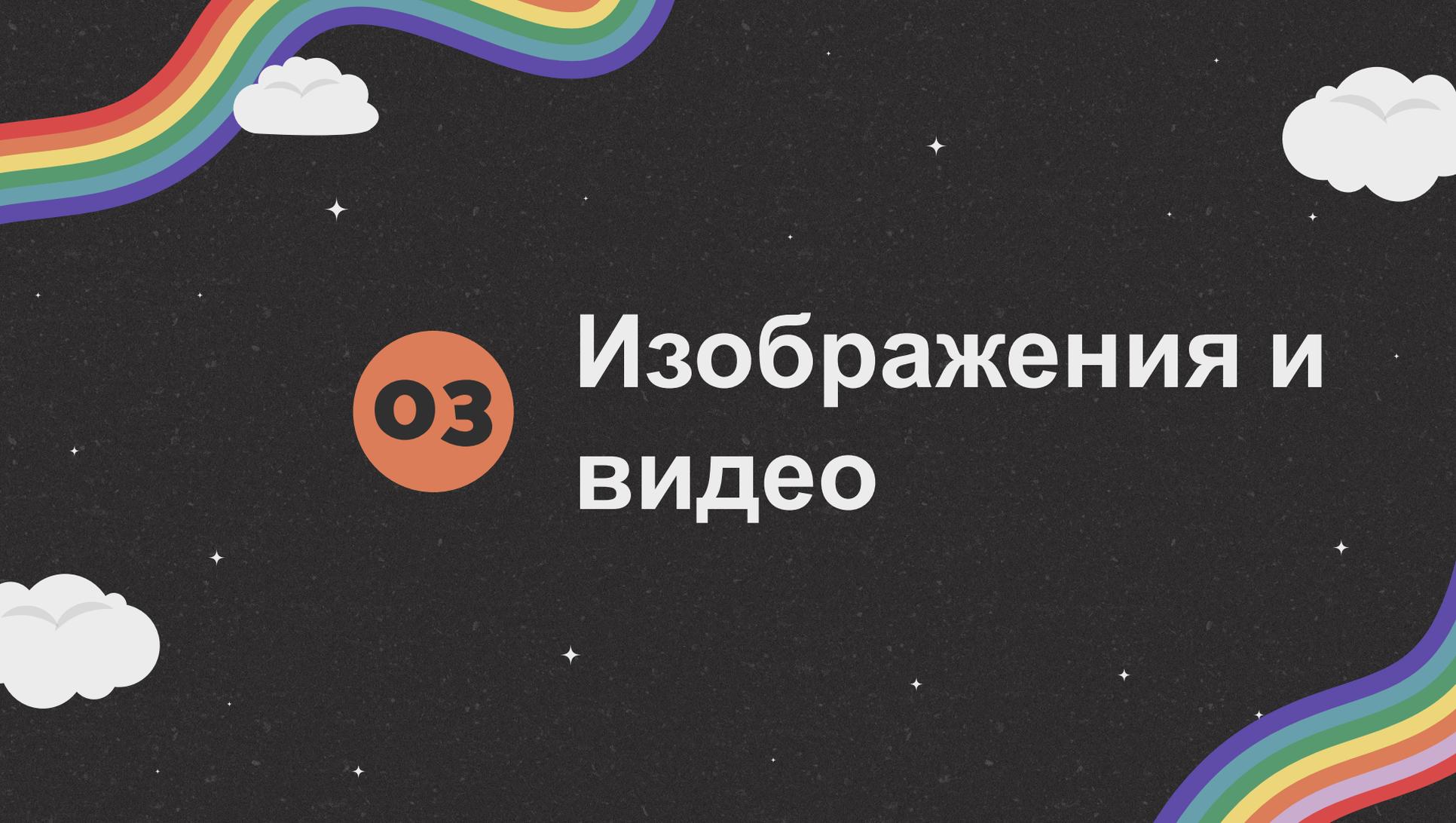
Результат:

- Омлет
 - Молоко
 - Яйца
- Блинчики
 - Молоко
 - Яйца
 - Мука
 - Сахар
 - Соль
 - Разрыхлитель
- Торт
- Пирог

ЗАДАНИЕ

Создайте список участников вашей команды и добавьте его на html-страничку





03

Изображения и видео

Изображения

Для добавления изображения используется тег ``. Это одинарный тег. Вот его основные атрибуты:

`src` - ссылка на картинку

`alt` - текст, который отображается вместо картинки, если она не загрузилась

`title` - текст, который отображается при наведении мыши на картинку

`width` - ширина картинки в пикселях

`height` - высота картинки в пикселях

Изображения

Пример:

```


# Семантические изображения с подписью в HTML 5

В HTML 5 появились теги для оформления объектов с подписями - `figure` и `figcaption`. Если твоей картинке нужна подпись - пользуйся ими. Пример кода:

```
<figure>

 <figcaption>
 Лого гугла от 2015 года
 </figcaption>
</figure>
```

# Семантические изображения с подписью в HTML 5

Результат:



# Небольшой тест

## Выбери истинные утверждения

*Воистину верные, воверную истинные.*

Select all correct options from the list

- alt - текст, который отображается при наведении мыши на картинку
- width - ширина картинки
- атрибут href нужен для указания ссылки на картинку
- title - текст, который отображается вместо картинки, если она не загрузилась

Correct answer from **47,054** learners

Total **42%** of tries are correct

# Небольшой тест

## Выбери истинные утверждения

*Воистину верные, воверную истинные.*

Select all correct options from the list

Good news for you, correct!

alt - текст, который отображается при наведении мыши на картинку

width - ширина картинки

атрибут href нужен для указания ссылки на картинку

title - текст, который отображается вместо картинки, если она не загрузилась

Correct answer from **47,054** learners

Total **42%** of tries are correct

# ЗАДАНИЕ

Добавьте 2 изображения на вашу страничку:  
с подписью и без



# Видео

Добавляет, воспроизводит и управляет настройками видеоролика на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег `<source>`.

# Видео

Основные атрибуты тега:

`autoplay` - видео начинает воспроизводиться автоматически после загрузки страницы.

`controls` - добавляет панель управления к видеоролику.

`height` - задает высоту области для воспроизведения видеоролика.

`loop` - повторяет воспроизведение видео с начала после его завершения.

`poster` - указывает адрес картинки, которая будет отображаться, пока видео не доступно или не воспроизводится.

`preload` - используется для загрузки видео вместе с загрузкой веб-страницы.

`src` - указывает путь к воспроизводимому видеоролику.

`width` - задает ширину области для воспроизведения видеоролика.

# ЗАДАНИЕ

1. Скачать видео
2. Добавить к себе в проект
3. Залить видео на свою html-страничку, добавив атрибуты height и width



**04**

# Адреса, ССЫЛКИ И ЯКОРЯ

# Ссылки и адреса

Ты уже знаком со ссылками:

```
Google
```

Повторим: для создания ссылки необходимо использовать тег `<a>`.

Атрибут `href` указывает адрес, по которому будет совершён переход.

Адреса бывают двух видов: абсолютные и относительные

# Абсолютные адреса

Абсолютный адрес, записанный в полной форме. Например,

<https://google.com/doodles>

Давай разберём этот адрес:

**https** - так называемая «схема», обычно это название протокола.

**HTTPS** - защищённая версия HTTP

**google.com** - доменное имя сайта

**/doodles** - путь (директория) внутри сайта

# Абсолютные адреса

Ещё пример:

<file:///C:/Users/admin/Desktop/Новая%20папка/image.jpg>

file - схема URI, предназначенная для того, чтобы адресовать файлы на локальном компьютере или в локальной сети

/C:/Users/admin/Desktop/Новая%20папка/image.jpg - путь до файла.  
%20 - код пробела в URI-кодировании.

# Относительные адреса

Относительный - сокращённый адрес. В таком адресе начальная часть опущена и браузер использует текущий адрес для определения полного адреса. Примеры:

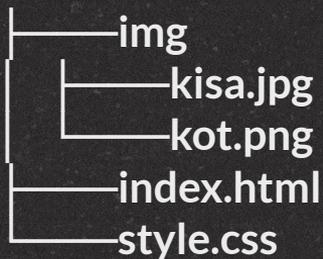
//google.com - ссылка на домен в текущем протоколе: если мы находимся по адресу, который начинается с http, то ссылка будет вести на <http://google.com>

/sheets - ссылка на путь внутри текущего домена: если мы находимся на <http://google.com>, то ссылка будет вести на <http://google.com/sheets>, а если на <http://facebook.com>, то на <http://facebook.com/sheets>.

# Пример использования относительного адреса

Файловая система:

Новая папка



Код в index.html:

```
...
<link rel="stylesheet"
href="style.css">
...


```

# Относительные адреса

При выполнении заданий с использованием файлов - картинок, шрифтов, веб-страниц, которые находятся локально (то есть у тебя на устройстве), используй относительные ссылки. Потому что при загрузке кода на сервер, ссылке вроде `file:///C:/Users/admin/Desktop/Новая%20папка/image.jpg` перестанут работать.

# Небольшой тест

## Выбери истинные утверждения

*Красна птица перьями, а человек знанием.*

Select all correct options from the list

- //google.com - относительная ссылка
- https://vk.com/reu\_its - абсолютная ссылка
- /users/21 - абсолютная ссылка
- //itc.digital/users/21#info - абсолютная ссылка

Correct answer from **43,279** learners

Total **49%** of tries are correct

# Небольшой тест

## Выбери истинные утверждения

*Красна птица перьями, а человек знанием.*

Select all correct options from the list

Great!

Correct answer from **43,279** learners

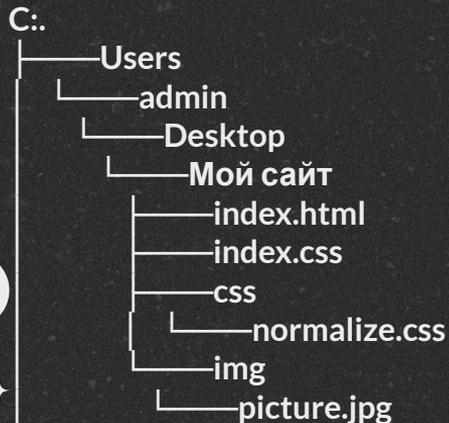
Total **49%** of tries are correct

- //google.com - относительная ссылка
- https://vk.com/reu\_itc - абсолютная ссылка
- /users/21 - абсолютная ссылка
- //itc.digital/users/21#info - абсолютная ссылка

# Используй. Относительные. Ссылки.

Замени абсолютные ссылки в HTML-коде на относительные.

Структура файловой системы:



```
<!-- index.html -->
<!DOCTYPE html>
<html lang="ru">
<head>
 <title>Мой сайт!!1</title>
 <meta charset="UTF-8">
 <link href="file:///C:/Users/admin/Desktop/Мой%20сайт/index.css"
rel="stylesheet">
 <link
href="file:///C:/Users/admin/Desktop/Мой%20сайт/css/normalize.css"
rel="stylesheet">
</head>
<body>
 <h1>Мама, я в интернете!</h1>

</body>
</html>
```

# Используй. Относительные. Ссылки.

```
<!-- index.html -->
<!DOCTYPE html>
<html lang="ru">
<head>
 <title>Мой сайт!!1</title>
 <meta charset="UTF-8">
 <link href="index.css" rel="stylesheet">
 <link href="css/normalize.css" rel="stylesheet">
</head>
<body>
 <h1>Мама, я в интернете!</h1>

</body>
</html>
```

# Якоря

Ссылки могут ссылаться не только на страницу, но ещё и на конкретное место на странице. Такие ссылки называются якорными, а места, на которые они ссылаются -- якорями.

Якорному элементу нужно прописать атрибут `id` с именем якоря, например:

```
<h1 id="anchor">Якорь</h1>
```

Теперь, чтобы сослаться на этот якорь, нужно использовать вот такую ссылку:

```
Перейти к якорю
```

При нажатии на ссылку произойдёт переход к нужному месту страницы, а в адресной строке к адресу страницы добавится адрес якоря `#anchor`

- Якоря могут использоваться как в относительных ссылках, как в примере выше, так и в абсолютных, например: `http://example.com/page#anchor`

# Задание на ссылки

Создай ссылку, которая:

1. **ссылается на `https://google.com`**
2. **будет открываться в новой вкладке**



СПАСИБО!



Пожалуйста, ваши вопросы

dannu2001@gmail.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

# EXAMPLES



## VENUS

Venus is the second planet from the Sun



## JUPITER

It's the biggest planet in the Solar System



## MARS

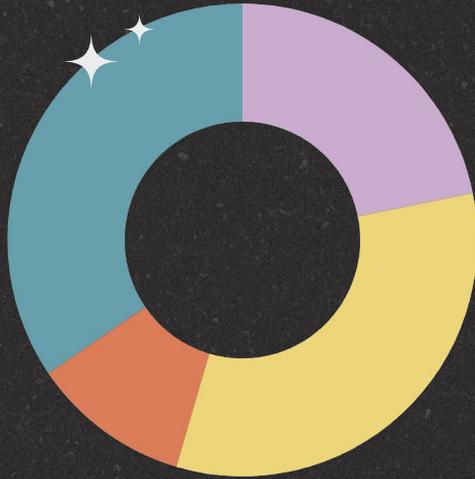
Despite being red, Mars is a cold place



## SATURN

Saturn is a gas giant and has several rings

# THIS IS A GRAPH



**21%**

## MERCURY

Mercury is the closest planet

**32%**

## VENUS

Venus is the second planet

**12%**

## JUPITER

Jupiter is the biggest planet

**35%**

## NEPTUNE

Neptune is the farthest planet

To modify this graph, click on it, follow the link, change the data and paste the new graph here, replacing this one



# THANKS YOUR TEACHERS

## DEAR TEACHER:

Mercury is the closest planet to the Sun and the smallest one in the Solar System. This planet's name has nothing to do with the liquid metal, since Mercury was named after the Roman messenger god



# THIS IS AN INFOGRAPHIC

## MERCURY

Mercury is the closest planet

1



## VENUS

Venus is the second planet

2



## JUPITER

Jupiter is the biggest planet

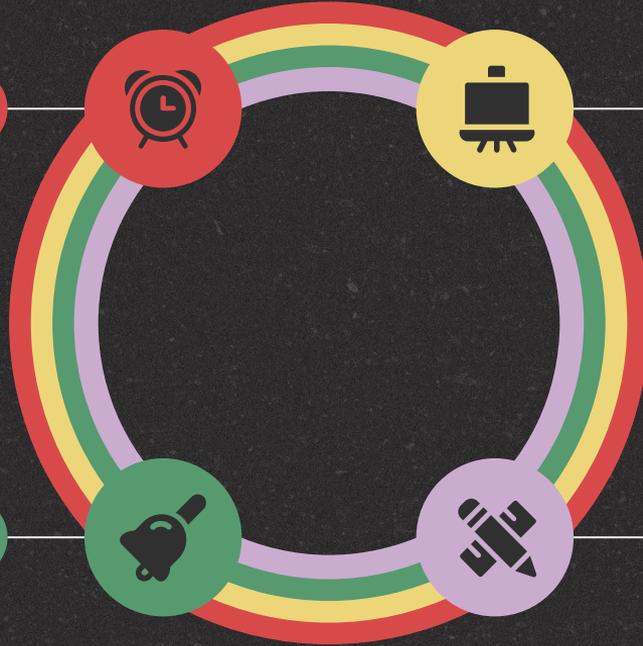
3



## NEPTUNE

Neptune is the farthest planet

4



# ORGANIZE YOUR WEEK

MON	TUE	WED	THU	FRI
Mercury is the closest planet			Neptune is the farthest	
	Venus is the second planet			Pluto is now a dwarf planet
		Mars is a cold place, not hot		
Jupiter is the biggest planet				Saturn is the ringed one

**NOTES:** Mercury is the closest planet to the Sun and the smallest one

**TO DO:** The Earth is the blue planet we live in. It's the only one that harbors life

# THIS IS A GRAPH



## MERCURY

Mercury the  
closest planet



## VENUS

Venus is the  
second planet



**60%**

Neptune is the  
farthest planet

To modify this graph, click on it, follow the link, change the data and paste the new graph here, replacing this one

# A TIMELINE



**MERCURY**  
Mercury the  
closest planet



**NEPTUNE**  
Neptune is the  
farthest planet

1

2

3

4

**VENUS**  
Venus is the  
second planet



**MARS**  
Mars is actually  
a very cold place



# THIS IS A MAP

## MERCURY

Mercury the  
closest planet



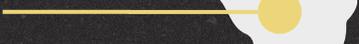
## VENUS

Venus is the  
second planet



## JUPITER

Jupiter is the  
biggest planet



## NEPTUNE

Neptune is the  
farthest planet



# CLASS SCHEDULE

hour	MON	TUE	WED	THU	FRI
9.30-10.30	Music	Science	Social	Math	Language
10.30-11.30	Art	Math	Art	Music	Math
<b>LUNCH TIME</b>					
12:00-12:30	Language	Music	Science	Art	Social
12:30-13:30	Science	Art	Math	Science	Art

# NOTES

- 
- Write your notes here
  - Write your notes here

- 
- Write your notes here
  - Write your notes here

# OUR TEACHERS



**THANKS MS. ANNE**



**THANKS MS. SARA**



**THANKS MR. JHON**



**THANKS MS. EMMA**



**150,000,000**

Big numbers catch your audience's attention



**36X5**

Mercury is the closest planet to the Sun



**27/2**

It has a beautiful name, but it's terribly hot



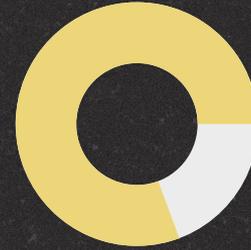
**12,56%**

Despite being red,  
Mars is a cold place



**55,67%**

Saturn is the ringed  
one and a gas giant



**87,89%**

Neptune is the  
farthest planet

# DID YOU KNOW THIS?



## MERCURY

Mercury is the closest planet to the Sun



## VENUS

It has a beautiful name, but it's terribly hot



## MARS

Despite being red, Mars is a cold place, not hot



## JUPITER

It's the biggest planet in our Solar System



## SATURN

Saturn is the ringed one and a gas giant



## NEPTUNE

Neptune is the farthest planet from the Sun



# OUR TEAM



**JENNA DOE**

Here you can talk a bit  
about this person



**TIMMY JIMMY**

Here you can talk a bit  
about this person



**SARA BONES**

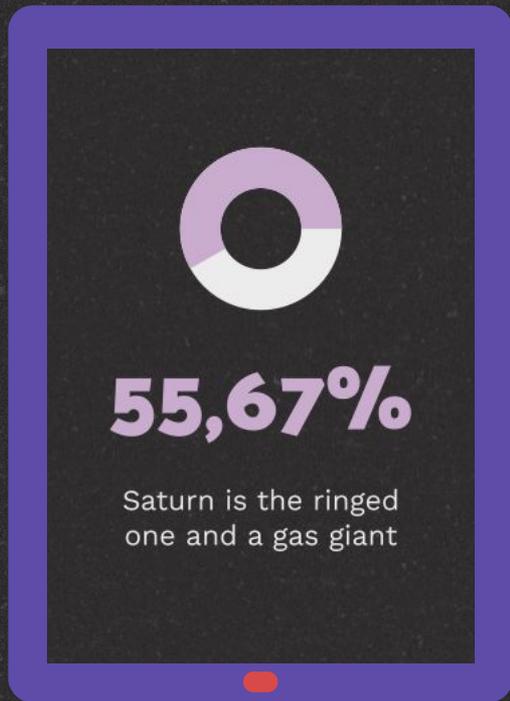
Here you can talk a bit  
about this person



# DESKTOP SOFTWARE

You can replace the image on the screen with your own work. Just right-click on it and select “Replace image”





## TABLET APP

You can replace the image on the screen with your own work. Just right-click on it and select “Replace image”

# MOBILE APP

You can replace the image on the screen with your own work. Just right-click on it and select “Replace image”



# ALTERNATIVE RESOURCES



# RESOURCES

## VECTOR

- People protesting against homophobic concept
- Pride day concept
- Flat st. patrick's day banner
- Flat rainbow around beige heart
- Rainbows collection with different shapes in flat style

## PHOTO

- School colleagues looking through books together
- Cute school girl reading her notes
- Kindergarten teacher holding a notebook
- Kids and teacher celebrating teacher's day
- Teacher helping kids in class
- Teacher and young student having a good time
- School colleagues holding each other