

Слайд 1

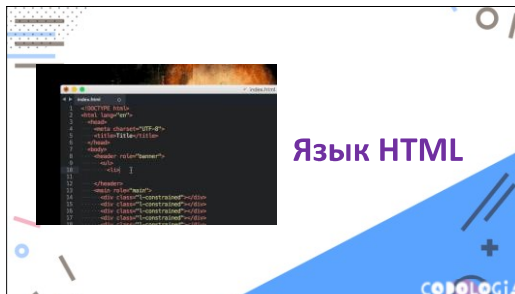


Для того, чтобы сделать сайт, нужно знать много разных веб языков.

Языки HTML и CSS предназначены для верстки сайтов (верстка - это размещение элементов сайта по нужным местам).

Язык PHP нужен для программирования сайта (с его помощью можно, к примеру, сделать регистрацию пользователей). Язык JavaScript нужен для того, чтобы 'оживить' сайт: к примеру, сделать меняющиеся картинки (слайдер).

Слайд 2

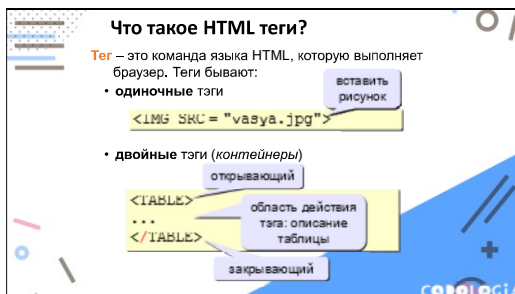


Язык HTML - это основа web сайтов, с его помощью создается каркас страницы, которую вы видите в браузере.

Если сравнивать страницу сайта и обычную бумажную книгу, то на сайте, как и в книге, есть абзацы и заголовки. В книге есть название всей книги (по сути самый главный заголовок), есть названия глав, параграфов в этих главах и так далее.

Заголовки, абзацы и другие блоки можно выделить и на странице сайта. Это делается с помощью HTML тегов.

Слайд 3



HTML теги - это специальные команды для браузера. Они говорят ему, что, к примеру, следует считать заголовком страницы, а что абзацем.

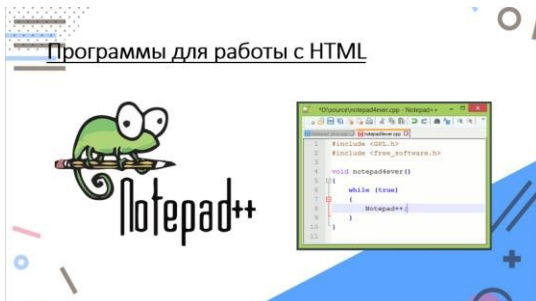
Теги строятся по такому принципу: уголок <, потом имя тега, а потом уголок >, вот так: <имя тега>. Имя тега может состоять из английских букв и цифр. Примеры тегов: <h1>, <p>, .

Теги обычно пишутся парами - открывающий тег и соответствующий ему закрывающий. Разница между открывающим и закрывающим тегами в том, что в закрывающем теге после уголка < стоит слеш /.

К примеру, <p> - так я открыл тег p, а так - </p> - я его закрыл. Все, что попадает между открывающим и закрывающим тегами, подпадает под воздействие нашего тега.

Бывают теги, которые не нужно закрывать, например,
 или .

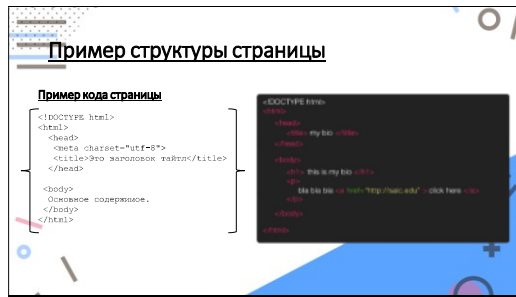
Слайд 4



Программы для работы с HTML

Notepad++ — свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов. Базируется на компоненте Scintilla, написан на C++ с использованием STL, а также Windows API и распространяется под лицензией GNU General Public License. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как интерпретаторы и преобразователи.

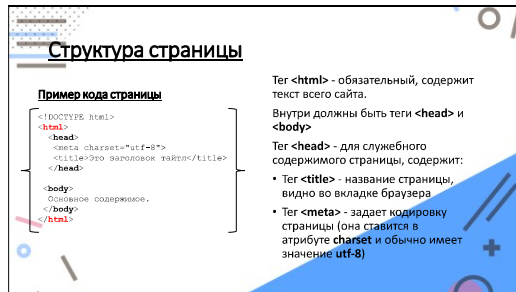
Слайд 5



1. Структура простейшей страницы

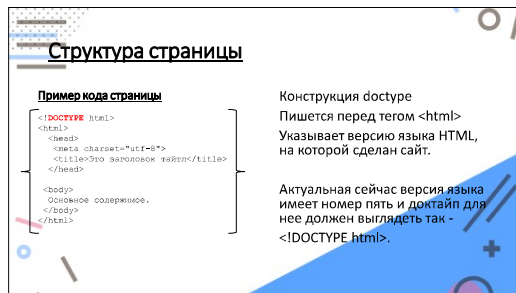
Страница сайта - это обычный текстовый файл с расширением **.html**. Внутри этого файла и хранится текст HTML страницы вместе с тегами. Этот файл обязательно должен иметь следующие теги: тег **<html>**, который должен содержать в себе текст всего сайта (все, что написано вне этого тега, браузером будет проигнорировано), а внутри него должны быть еще два тега: тег **<head>** для служебного содержимого страницы и тег **<body>** - для основного текста, который и виден на экране браузера.

Слайд 6



В служебное содержимое, которое располагается внутри тега **<head>**, входит много различных вещей, но пока нам нужны только две из них. Это тег **<title>**, задающий название страницы, которое будет видно во вкладке браузера, и тег **<meta>**, который задает кодировку страницы (она ставится в атрибуте **charset** и обычно имеет значение **utf-8**).

Слайд 7



Кроме того, перед тегом **<html>** обычно пишется конструкция **doctype**, которая указывает версию языка HTML, на которой сделан сайт. Актуальная сейчас версия языка имеет номер пять и доктайп для нее должен выглядеть так - **<!DOCTYPE html>**.

Итак, давайте посмотрим на основную структуру страницы (чтобы запустить это пример в браузере - скопируйте его в текстовый файл с расширением **.html** и откройте в браузере).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>
```

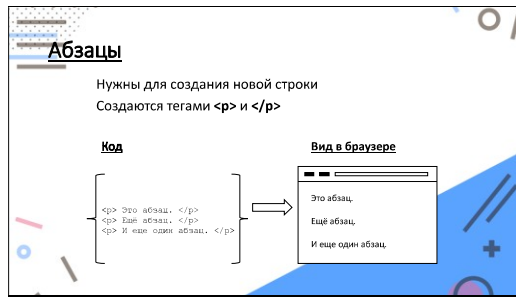
```
<body> Это основное содержимое страницы. </body>
```

```
</html>
```

Ну, а теперь, научившись создавать простейшие страницы, мы перейдем к изучению полезных тегов, которые следует использовать внутри тега **<body>**. Это будут теги для абзацев, заголовков, списков, ссылок и других полезных вещей.

Итак, приступим.

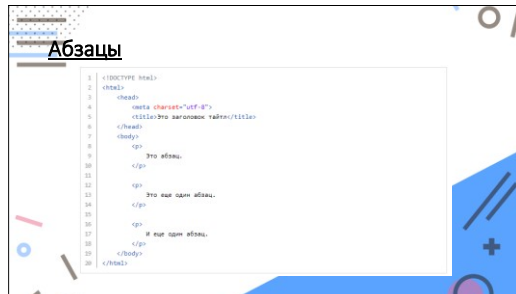
Слайд 8



2. Абзацы

Одним из основных элементов страницы являются **абзацы**. Их можно сравнить с абзацами в книге - каждый абзац начинается с новой строки и имеет так называемую красную строку (это когда первая строка текста абзаца немного отступает вправо). По умолчанию красной строки нет, но ее легко сделать.

Слайд 9



2. Абзацы

Пример

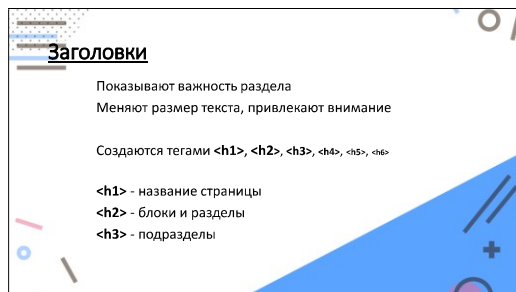
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>

  <body>
    <p>
      Это абзац.
    </p>

    <p>
      Это еще один абзац.
    </p>

    <p>
      И еще один абзац.
    </p>
  </body>
</html>
```

Слайд 10



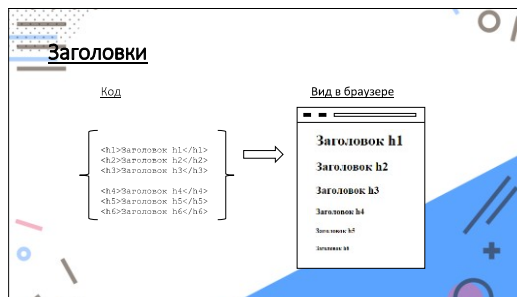
3. Заголовки h1, h2, h3, h4, h5, h6

Кроме абзацев важное значение на странице имеют **заголовки**. Их также можно сравнить с заголовками из книги - каждая глава имеет свой заголовок (название этой главы) и разбита на параграфы, которые тоже имеют свои заголовки. Ну, а основной текст страницы располагается в абзацах.

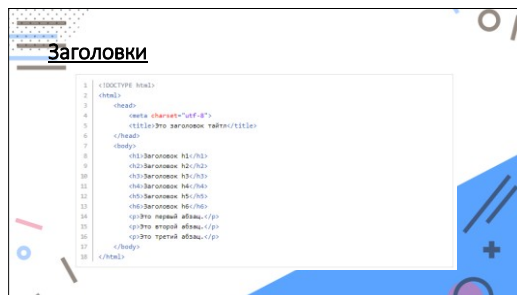
Заголовки создаются с помощью тегов **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>**, **<h6>**. Они имеют разную степень важности. В заголовке **h1** следует располагать **название всей HTML страницы**, в **h2** - название **блоков** страницы, в **h3** - название подблоков и так далее.

Все заголовки по умолчанию жирные и имеют разный размер (это можно поменять через CSS).

Слайд 11



Слайд 12



Смотрите пример:

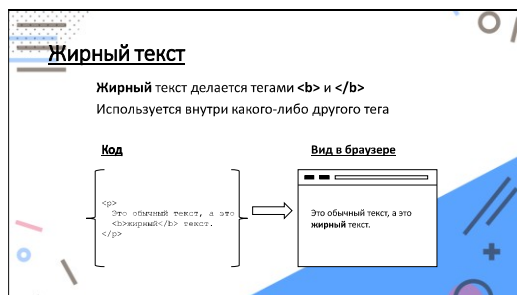
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>
```

```
<body>
  <h1>Заголовок h1</h1>
  <h2>Заголовок h2</h2>
  <h3>Заголовок h3</h3>
  <h4>Заголовок h4</h4>
  <h5>Заголовок h5</h5>
  <h6>Заголовок h6</h6>
```

```
<p>Это первый абзац.</p>
<p>Это второй абзац.</p>
<p>Это третий абзац.</p>
```

```
</body>
</html>
```

Слайд 13

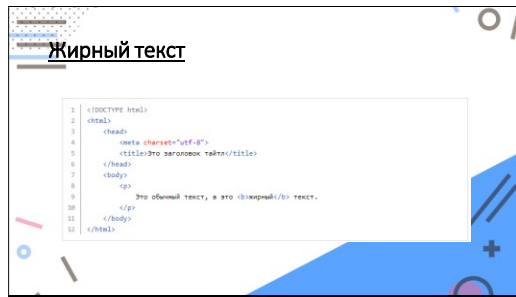


4. Жирный текст

Вы уже знаете, что заголовки по умолчанию **жирные**. Однако, можно сделать жирным и обычный текст - достаточно взять его в тег ``.

Обратите внимание на то, что тег `` следует использовать внутри какого-либо другого тега, например абзаца. В этом случае абзацы создают общую структуру страницы (абзацы и заголовки), а тег `b` делает жирным отдельные кусочки текста.

Слайд 14



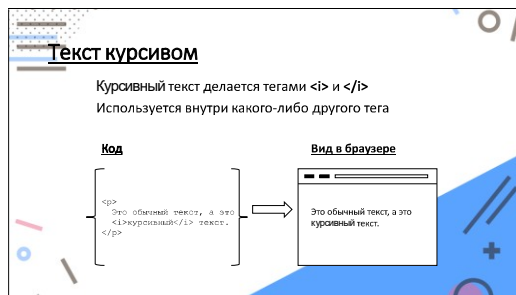
Смотрите пример:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
<title>Это заголовок тайтл</title>
</head>
<body>
  <p> Это обычный текст, а это <b>жирный</b> текст.
</p>
</body>
</html>

```

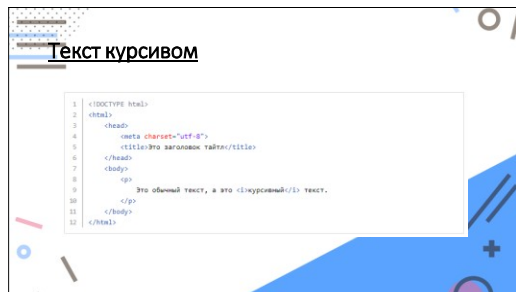
Слайд 15



5. Курсив

Кроме жирного можно сделать также и курсив с помощью тега `<i>`.

Слайд 16



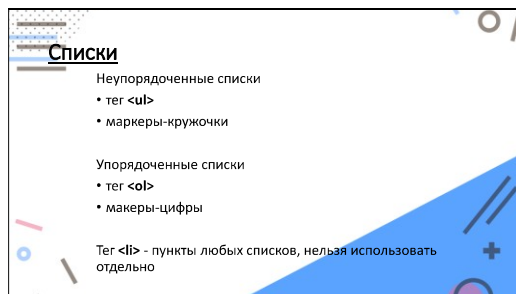
Пример:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Это заголовок тайтл</title>
</head>
<body>
  <p>
    Это обычный текст, а это <i>курсивный</i> текст.
  </p>
</body>
</html>

```

Слайд 17



6. Списки

Наряду с абзацами и заголовками существует еще один важный элемент страницы - это **списки**. Такие элементы наверняка знакомы всем пользователям интернета. Они представляют собой перечисление чего-либо (список) по пунктам. Возле каждого пункта списка обычно стоит закрашенный кружок (он называется **маркер** списка).

Списки создаются с помощью тега ``, внутри которого обязательно должны идти теги ``. Тег `ul` задает сам список, а в теги `li` следует положить пункты списка (то есть одному `li` соответствует один маркер списка). Имейте ввиду, что отдельно теги `li` использовать нельзя.

Слайд 18

Неупорядоченные списки

Создаются тегом `` и ``

Код

```
<ul>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
  <li>Третий пункт</li>
</ul>
```

Вид в браузере

- Первый пункт
- Второй пункт
- Третий пункт

Списки, созданные через тег `ul`, называются **неупорядоченными** списками. Такое название им дано потому, что существуют еще и **упорядоченные** списки, у которых вместо маркеров-точек будут маркеры-цифры. Такие списки вместо тега `ul` имеют тег `ol`, а пункты таких списков также создаются через теги `li`.

Слайд 19

Неупорядоченные списки

Создаются тегом `` и ``

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Это заголовок тайтл</title>
6   </head>
7   <body>
8     <ul>
9       <li>Первый пункт списка.</li>
10      <li>Второй пункт списка.</li>
11      <li>Третий пункт списка.</li>
12    </ul>
13  </body>
14 </html>
```

Смотрите пример:

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>
  <body>
    <ul>
      <li>Первый пункт списка.</li>
      <li>Второй пункт списка.</li>
      <li>Третий пункт списка.</li>
    </ul>
  </body>
</html>
```

Слайд 20

Упорядоченные списки

Создаются тегом `` и ``
Нумерация перестраивается при добавлении

Код

```
<ol>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
  <li>Третий пункт</li>
</ol>
```

Вид в браузере

1. Первый пункт
2. Второй пункт
3. Третий пункт

Давайте сделаем упорядоченный список, используя тег `ol`:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>
  <body>
    <ol>
      <li>Первый пункт списка.</li>
      <li>Второй пункт списка.</li>
      <li>Третий пункт списка.</li>
    </ol>
  </body>
</html>
```

Слайд 21

Упорядоченные списки

Создаются тегом `` и ``
Нумерация перестраивается при добавлении

Код

```
<ol>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
  <li>Третий пункт</li>
</ol>
```

Вид в браузере

1. Первый пункт
2. Второй пункт
3. Третий пункт

Удобство упорядоченных списков в том, что я могу вставить новый пункт списка в любое место - и нумерация сама перестроится (то есть мне не приходится за ней следить в случае каких-либо изменений, как было бы в том случае, если бы я расставлял ее вручную).

Слайд 22

Упорядоченные списки

Создаются тегом `` и ``
Нумерация перестраивается при добавлении

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Это заголовок тайтл</title>
6   </head>
7   <body>
8     <ol>
9       <li>Первый пункт списка.</li>
10      <li>Второй пункт списка.</li>
11      <li>Третий пункт списка.</li>
12    </ol>
13  </body>
14 </html>

```

Удобство упорядоченных списков в том, что я могу вставить новый пункт списка в любое место - и нумерация сама перестроится (то есть мне не приходится за ней следить в случае каких-либо изменений, как было бы в том случае, если бы я расставлял ее вручную).

Слайд 23

Ссылки

Создаются тегом `<a>` и ``
Атрибут `href` – адрес куда перейти

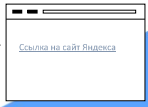
Код

```

<a href="http://ya.ru">
  Ссылка на сайт Яндекса
</a>

```

Вид в браузере



7. Ссылки

Ссылки являются теми элементами, которые делают из интернета интернет. Нажимая на ссылки, мы можем переходить с одной страницы сайта на другую. Если бы их не было - интернет был бы просто набором страниц, никак не связанных друг с другом.

Ссылка создается с помощью тега `<a>`, при этом у нее должен быть обязательный атрибут `href`, в котором хранится адрес той страницы, на которую ведет ссылка.

Смотрите пример:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Это заголовок тайтл</title>
</head>

```

```

<body>
  <a href="http://phphtml.net">Ссылка на сайт
  phphtml.net.</a>
</body>

```

```

</html>

```

Ссылки бывают **абсолютные** и **относительные**, кроме того, они могут вести как на ваш сайт, так и на чужой.

Слайд 24

Картинки

Создаются тегом ``, он НЕ закрывается
Атрибут `src` – путь к картинке


Код

```



```

Вид в браузере



8. Картинки

Давайте теперь разберемся с тем, как разместить **изображение** на странице вашего сайта. Для этого предназначен тег ``, имеющий обязательный атрибут `src`, в котором хранится путь к файлу картинки. Обратите внимание на то, что тег `` не требует закрывающего тега.

Слайд 25



Как это работает, посмотрим на следующем примере:

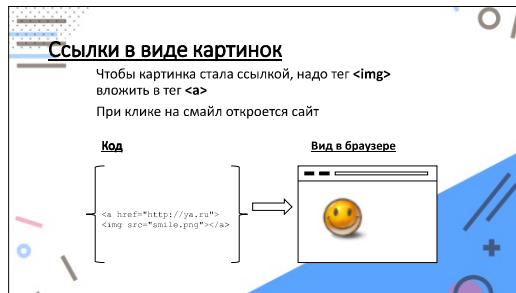
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>

  <body>
    
  </body>
</html>

```

Слайд 26



9. Ссылки в виде картинок

Ссылкой может быть не только текст, но и **картинка** - для этого достаточно тег `` вложить в тег `<a>`, как это сделано в следующем примере.

Нажмите на картинку - и вы перейдете по ссылке на сайт `phphtml.net` (чтобы потом вернуться назад на книжку - нажмите кнопку "назад" в браузере):

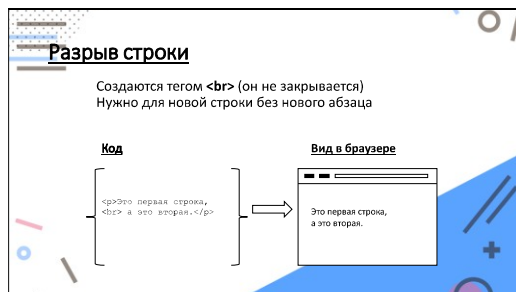
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>

  <body>
    <a href="http://ya.ru"></a>
  </body>
</html>

```

Слайд 27



10. Разрыв строки

Бывают, однако, ситуации, когда мы хотели бы, чтобы у нас был один абзац, но некоторый текст в нем начинался с новой строки. Зачем такое может понадобиться? К примеру, я хочу набрать стихотворение, но не хочу разбивать каждую его строку в отдельный абзац, так как это было бы не очень логично.

Чтобы сделать такое, в том месте, где должен быть разрыв строки, следует написать тег `
`. Учтите, что этот тег особенный и не имеет закрывающего.

Слайд 28

Разрыв строки

Создаются тегом `
` (он не закрывается)
Нужно для новой строки без нового абзаца

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Это заголовок тайтл</title>
6   </head>
7   <body>
8     <p>Это первая строка текста, <br> а это вторая.</p>
9   </body>
10 </html>

```

Давайте посмотрим на его работу на следующем примере:

```

<!DOCTYPE html> <html> <head> <meta charset="utf-8">
<title>Это заголовок тайтл</title> </head>

```

```

<body>
  <p>Это первая строка текста,
  <br> а это вторая.</p>
</body>
</html>

```

Слайд 29

Комментарии HTML

Это текст, который браузер проигнорирует
Нужны для пометок себе и другим
Можно прятать код, который не хочется удалять

Код

```

<!--
  Это комментарий HTML -
  на странице сайта он не
  отображается.
-->

```

Текст страницы

Вид в браузере

Текст страницы

11. Комментарии HTML

Практически во всех языках программирования есть такое понятие как "комментарии". **Комментарии** - это такой текст, который будет проигнорирован браузером - на экране он будет не виден, но останется в исходном коде страницы.

Для чего нужны комментарии? **Во-первых**, для того, чтобы оставлять в исходном коде пометки себе или тому человеку, который будет править ваш код после вас. **Во-вторых**, с помощью комментариев можно убирать некоторые вещи, которые вы бы сейчас хотели удалить, но потом, может быть, захотите вернуть. Их можно просто временно закоментировать, а потом откомментировать при необходимости.

Комментарии в HTML оформляются следующим образом: сначала уголок, знак ! и два дефиса - `<!--`, потом текст комментария, а потом два дефиса и уголок - `-->`.

Слайд 30

Комментарии HTML

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Это заголовок тайтл</title>
6   </head>
7   <body>
8     <!--
9       Это комментарий HTML -
10      на странице сайта он не отображается.
11    -->
12     Это основное содержимое страницы.
13   </body>
14 </html>

```

Смотрите пример:

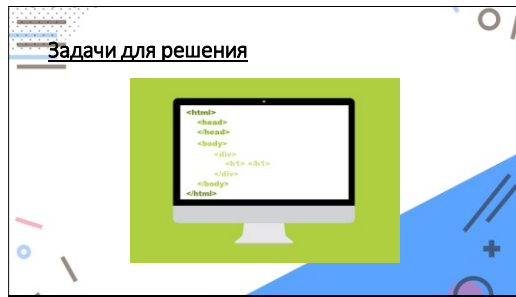
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>
  <body>
    <!--
      Это комментарий HTML -
      на странице сайта он не отображается.
    -->

    Это основное содержимое страницы.
  </body>
</html>

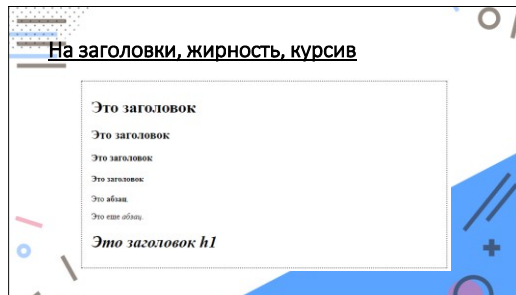
```

Слайд 31



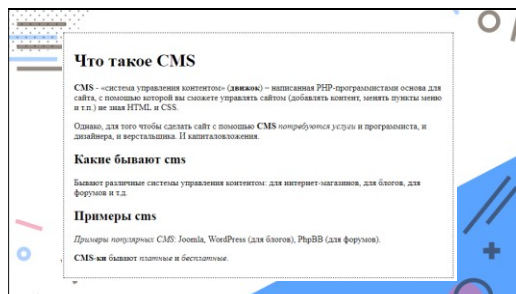
Задачи для решения

Слайд 32



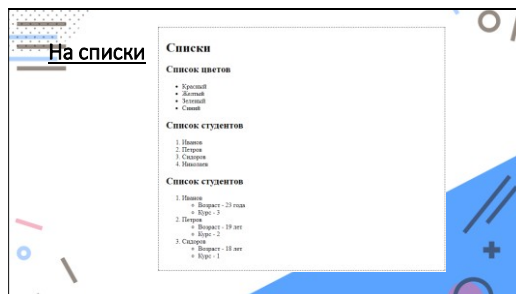
На заголовки, жирность, курсив
Повторите страницу по данному по образцу:

Слайд 33



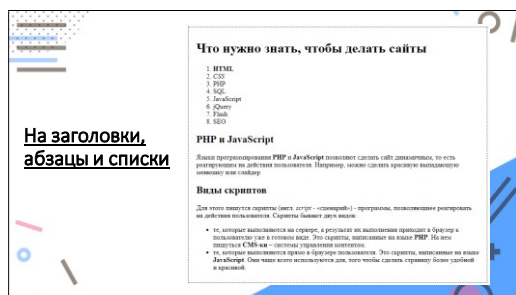
Повторите страницу по данному по образцу:

Слайд 34



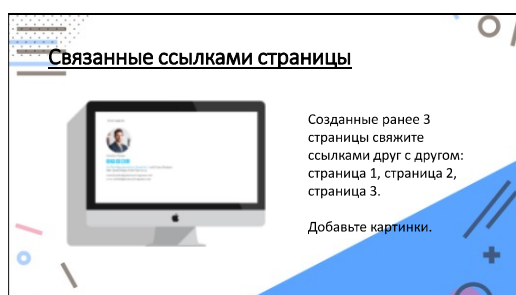
На списки
Повторите страницу по данному по образцу:

Слайд 35



На заголовки, абзацы и списки
Повторите страницу по данному по образцу:

Слайд 36



Связанные ссылками страницы
Созданные ранее 3 страницы свяжите ссылками друг с другом: страница 1, страница 2, страница 3.
И добавьте в них картинки.