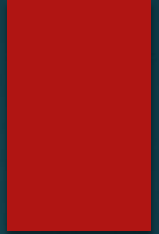


ВЕРСТКА

САЙТОВ

(HTML5+ CSS3)



Цель курса

Теоретические знания:

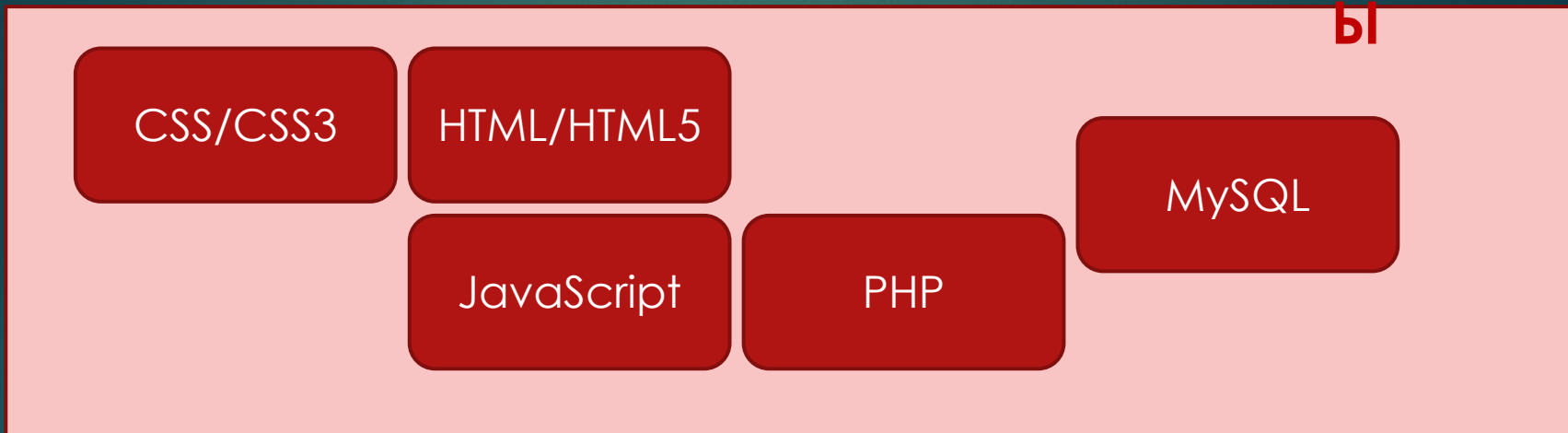
- ✓ основы языка разметки гипертекста HTML/HTML5;
- ✓ каскадные таблицы стилей CSS/CSS3;
- ✓ отзывчивый и интуитивно понятный веб-дизайн.

Практические навыки:

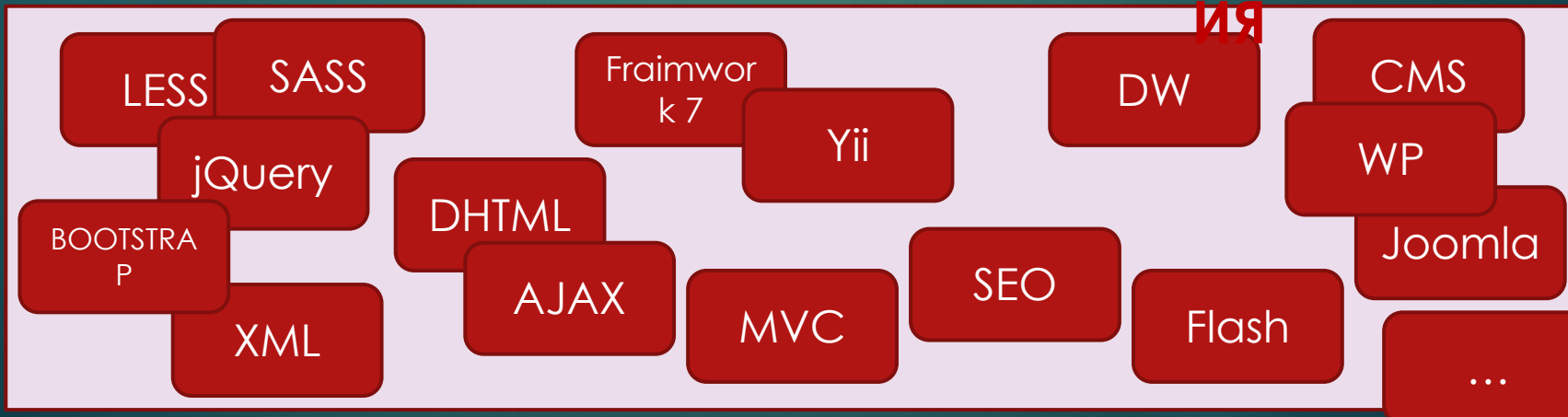
- ✓ создание web-страниц;
- ✓ написание правильного кода;
- ✓ использовании популярного редактора Adobe Dreamweaver.

WEB-ТЕХНОЛОГИИ

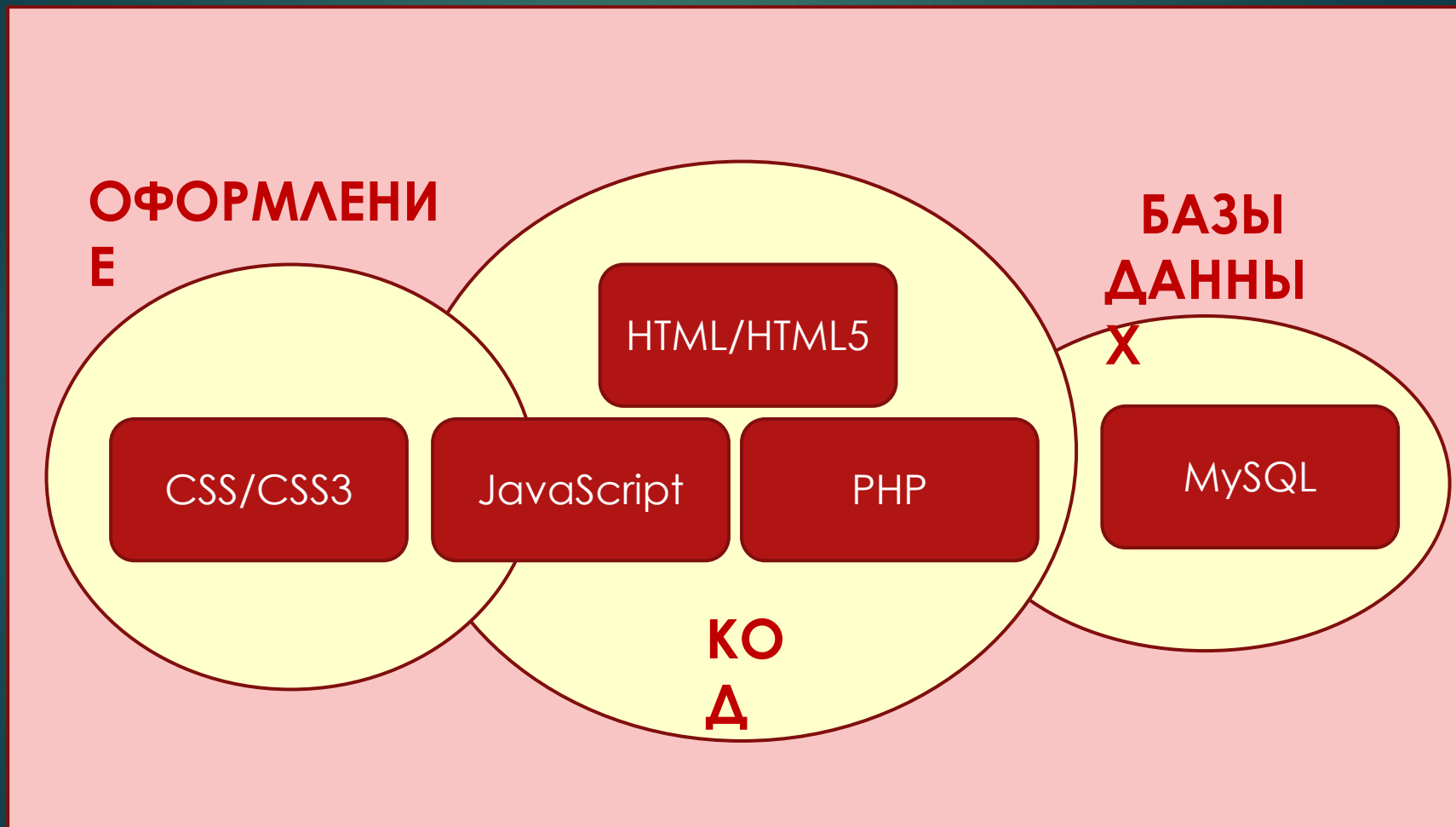
ОСНОВЫ



ДОПОЛНЕНИЯ

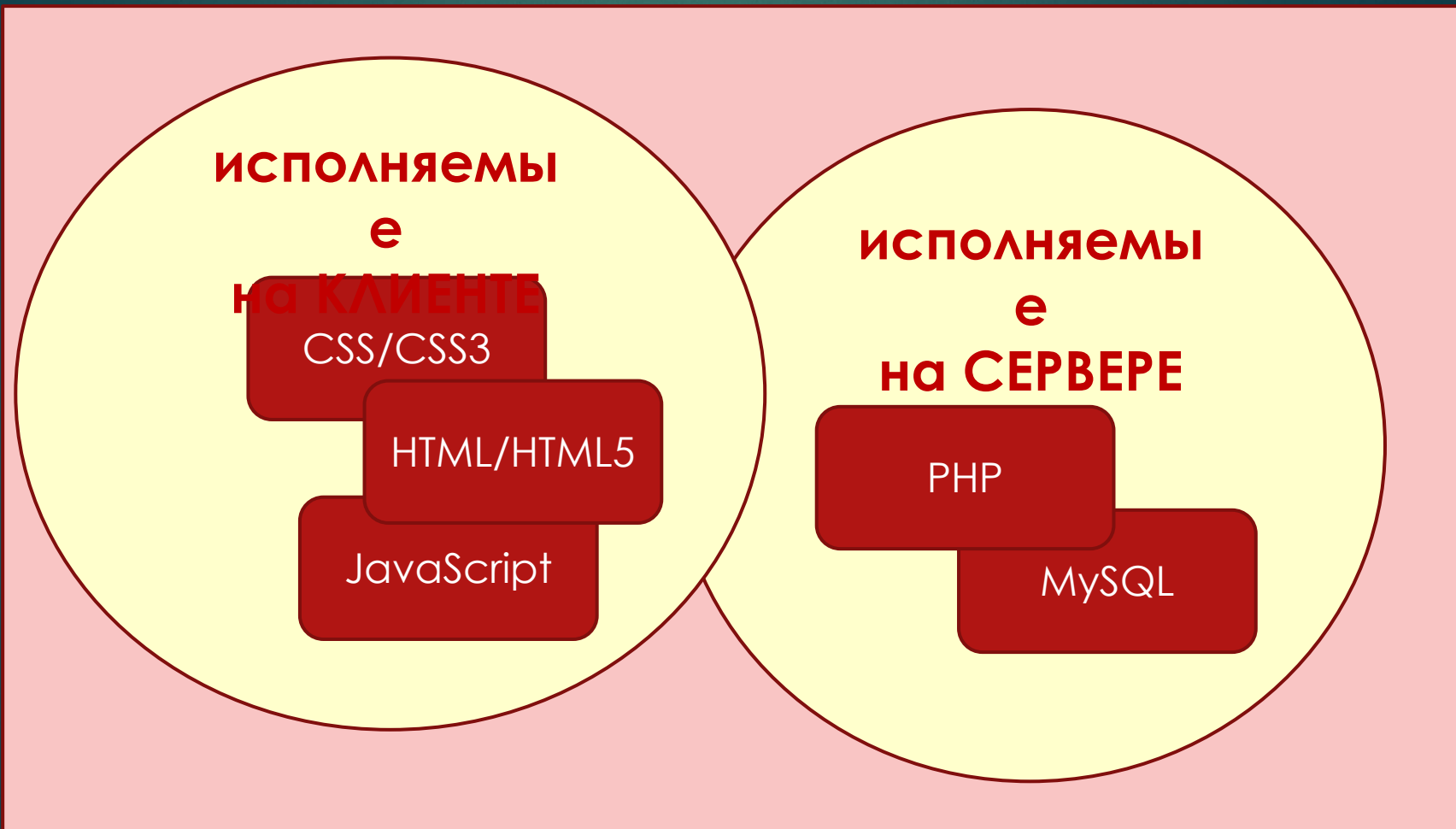


WEB - ТЕХНОЛОГИИ



WEB – ТЕХНОЛОГИИ

(ДВА ОСНОВНЫХ КЛАССА)



HTML/HTML5

- ✓ Язык HTML был разработан британским **учёным Тимом Бернерсом-Ли** приблизительно в 1986—1991 годах в стенах ЦЕРНа в Женеве в Швейцарии.
- ✓ **HTML** (от англ. **HyperText Markup Language** — «язык гипертекстовой разметки») — стандартный язык разметки документов во Всемирной паутине.
- ✓ **HTML5 (version 5)** — язык для структурирования и представления содержимого всемирной паутины.
- ✓ HTML5 — 28 октября 2014 года.
- ✓ Цель разработки HTML5 — **улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости.**

БЫЛО

```
<div id="header">
```

```
<div id="nav">
```

```
<div class="article">
```

```
<div class="section">
```

```
<div id="sidebar">
```

```
<div id="footer">
```

ΣΤΑΛΟ



`<header>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`

CSS/CSS3

- ✓ **CSS** (англ. **Cascading Style Sheets** — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.
- ✓ Термин «каскадные таблицы стилей» был предложен **Хокон Виум Ли** в 1994 году. Совместно с он стал развивать CSS.
- ✓ **Бертом Босом**
- ✓ **CSS3 (каскадные таблицы стилей третьего поколения)** — активно разрабатываемая спецификация CSS. Представляет собой формальный язык, реализованный с помощью языка разметки. Самая масштабная редакция по сравнению с CSS1, CSS2 и CSS2.1. Главной особенностью CSS3 является **возможность создавать анимированные элементы** без использования JS, **поддержка линейных и радиальных градиентов, теней, сглаживания** и многое другое.

Консорциум Всемирной паутины W3C

World Wide Web Consortium	
	
Тип	Консорциум
Основание	октябрь 1994
Основатели	Тим Бернерс-Ли
Расположение	MIT/CSAIL - США ERCIM - Франция Keio University - Япония НИУ ВШЭ - Россия и др.
Сайт	w3.org 

Консорциум Всемирной паутины (англ. **World Wide Web Consortium, W3C**) — организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины. Консорциум возглавляет сэр Тимоти Джон Бернерс-Ли, автор множества разработок в области информационных технологий.

Проверка кода



CSS Validation Service

Проверка каскадных таблиц стилей (CSS) и документов (X)HTML с таблицами стилей

Проверить URI

Проверить загруженный файл

Проверить набранный текст

Проверка по URI

Введите URI документа (HTML с CSS или только CSS) для проверки:

Адрес:

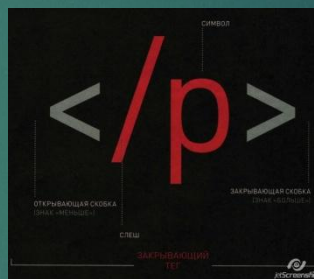
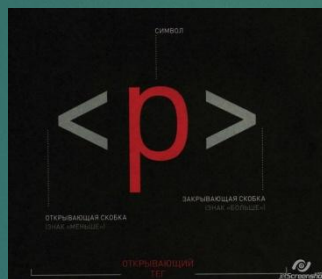
▶ [Дополнительные возможности](#)

Проверить

ЧТО ТАКОЕ ТЕГ?

- ✓ **HTML теги** являются основой языка HTML.
- ✓ Все содержимое web-страницы задается с помощью тегов.
- ✓ Структура на писания:

Открывающийся
тег



Закрывающийся
тег

- ✓ Могут быть не парные теги, например, ****
- ✓ Теги могут вкладываться друг в друга
<p><i>Текст</i></p>

Классификация HTML ТЕГОВ

Текст

`<h1>` `<p>` `<div>`
`...`

Ссылки

`<a`

Списки

``

Изображен
ие

`...`

`<img`

Таблицы

`<table>`

...

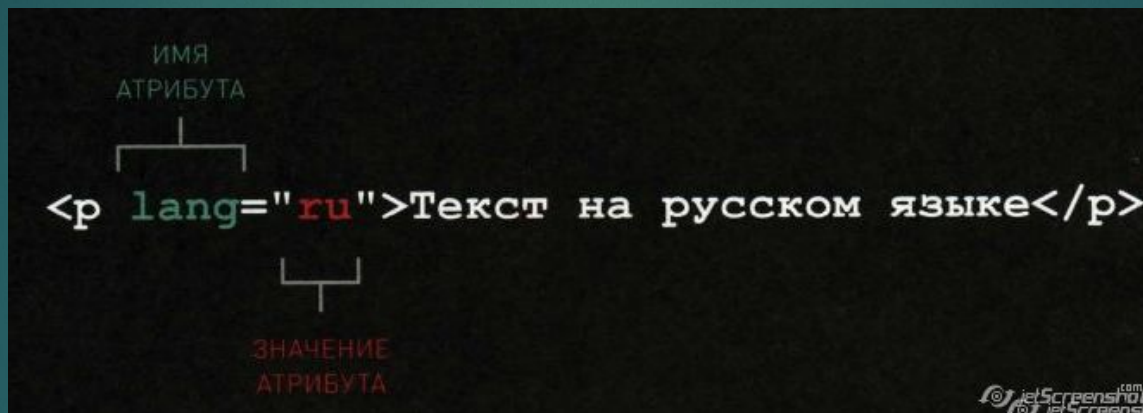
`<form>`

Формы

...

ЧТО ТАКОЕ АТТРИБУТ?

- ✓ У **HTML тегов** могут быть определены **HTML атрибуты**.
- ✓ **HTML атрибуты** сообщают браузеру, каким образом должен отображаться тот или иной элемент страницы.
- ✓ Названия и значения атрибутов не чувствительны к регистру, но, тем не менее, рекомендуется набирать их в нижнем регистре.



- ✓ Атрибуты позволяют изменять свойства элемента - **параметр="значение"**.
- ✓ Множественные значения атрибутов записываются через пробел, например - **class="nav top"**

Список атрибутов

- ✓ **class** - Определяет имя класса для элемента (используется для определения класса в таблице стилей). Принимаемые значения: имя класса.
- ✓ **id** - Определяет уникальный идентификатор элемента. Принимаемые значения: id – идентификатор элемента.
- ✓ **lang** - Определяет код языка содержимого (контента) в элементе. Принимаемые значения: код языка.
- ✓ **style** - Указывает на код CSS, применяемую для оформления элемента. Принимаемые значения: код CSS.
- ✓ **tabindex** - Определяет порядок перехода к элементу при помощи клавиши TAB. Принимаемые значения: порядковый номер.
- ✓ **title** - Определяет дополнительную информацию об элементе, задавая всплывающую подсказку для страницы. Принимаемые значения: текст.

Теги для HTML текста

1. Теги заголовков: **<h1>...<h6>**

2. Абзацы, средства переноса текста

<p> разбивает текст на отдельные абзацы

**
** переносит текст на следующую строку

<hr> используется для разделения контента на веб-странице. Отображается в виде горизонтальной линии.

3. Теги для форматирования текста:

**** и **** для задания жирного начертания шрифта;

<i> и **** отображает шрифт курсивом;

<sub> используется для создания нижних индексов.

<sup> используется для создания степеней.

<pre> позволяет вывести текст на экран, сохраняя форматирование.

<blockquote> выделяет цитаты внутри документа.

HTML ССЫЛКИ

1. Структура написания:

**Название
ССЫЛКИ**

2. Абсолютный адрес: <http://mysait.ru/files/text.html>

Относительный адрес: <files/text.html>

3. **Оформление ссылки:** Текст ссылки отображается подчеркнутым, цвет шрифта – синий, при наведении на ссылку курсор меняет вид.

4. **Ссылки на разделы текущей страницы:** Атрибут href содержит имя указателя (**якорь**), а не URL-адрес. Перед именем указателя ставится знак #.

5. Атрибуты:

href - URL-адрес документа.

target – место открытия документа:

_self – страница загружается в текущее окно;

_blank – страница открывается в новом окне браузера;

_parent – страница загружается во фрейм-родитель;

Блочный и встроенный элемент

`<div>`

``

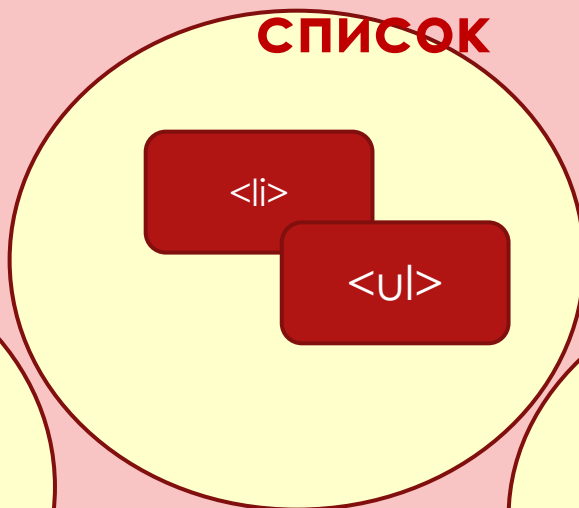
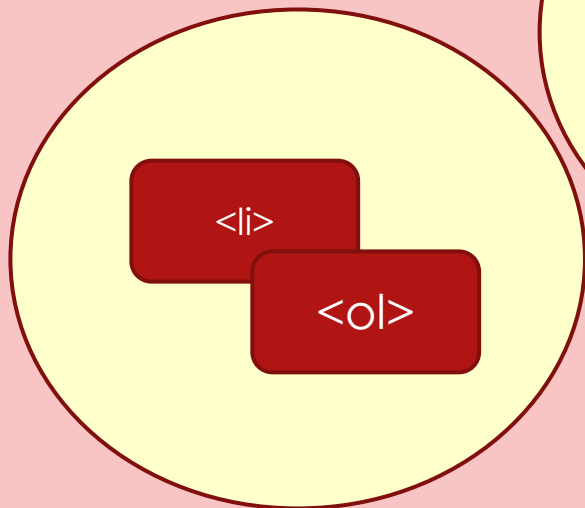
- ✓ **Тег `<div>`** - блочный элемент, для размещения текста в блоке, отделяется абзацем от остального текста.
- ✓ **Тег ``** - является встроенным элементом, не отделяется абзацем от остального текста.

СПИСКИ

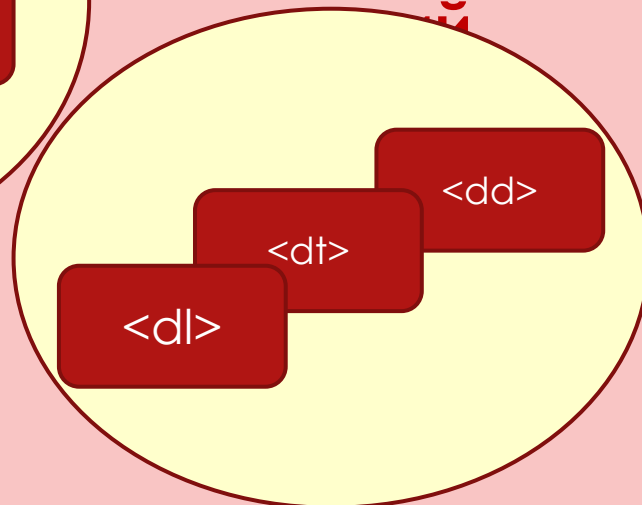
Маркированны
й

СПИСОК

Нумерованный
СПИСОК



Список
определен
й



ИЗОБРАЖЕНИЯ

1. Структура написания:

```

```

2. Атрибуты:

Alt - Атрибут alt обозначает альтернативный текст для изображения. Выводится на месте появления изображения до его загрузки или при отключенной графике, а также выводится всплывающей подсказкой при наведении курсора мыши на изображение. Принимаемые значения: текст.

Height - Атрибут height задает высоту изображения. Принимаемые значения: px/%.

Src - Атрибут src задает URL-адрес изображения. Принимаемые значения: url-адрес.

Width - Атрибут width задает ширину изображения. Принимаемые значения: px/%.

ТАБЛИЦЫ

HTML таблицы состоят из ячеек, образующихся при пересечении строк и столбцов.

```
<table>
<tr>
<th>Текст в ячейке заголовка</th>
<th>Текст в ячейке заголовка</th>
</tr>
...
</table>
```

colspan - Количество ячеек в строке для объединения по горизонтали.

`<td colspan="3">`. Возможные значения: число от 1 до 999.

rowspan - Количество ячеек в столбце для объединения по вертикали.

`<td rowspan="2">`. Возможные значения: число от 1 до 999.

span - Количество колонок, объединяемых для задания единого стиля, по умолчанию равно 1. `<col span="2">`. Принимаемые значения: любое целое положительное число.

ФОРМЫ

- ✓ **HTML формы** являются элементами управления, которые применяются для сбора информации от посетителей веб-сайта.
- ✓ Веб-формы состоят из набора текстовых полей, кнопок, списков и других элементов управления, которые активизируются щелчком мыши. Технически формы передают данные от пользователя удаленному серверу.
- ✓ Для получения и обработки данных форм используются языки веб-программирования, такие как **PHP**, **Perl**.
- ✓ С помощью форм можно использовать следующие HTML-элементы:

text

textarea

radio

checkbox

submit

select

hidden

file

HTML теги для разметки

СТРАНИЦЫ

<html>

Указывает программам просмотра html страниц начало и конец документа.

<head>

Техническая информация о документе

<meta>

Подключение мета тегов

<title>

Название страницы

<link>

Подключение файлов стилей

<script>

Подключение javascript

<style>

Подключение стилей

<body>

Информация отображаемая на странице сайта

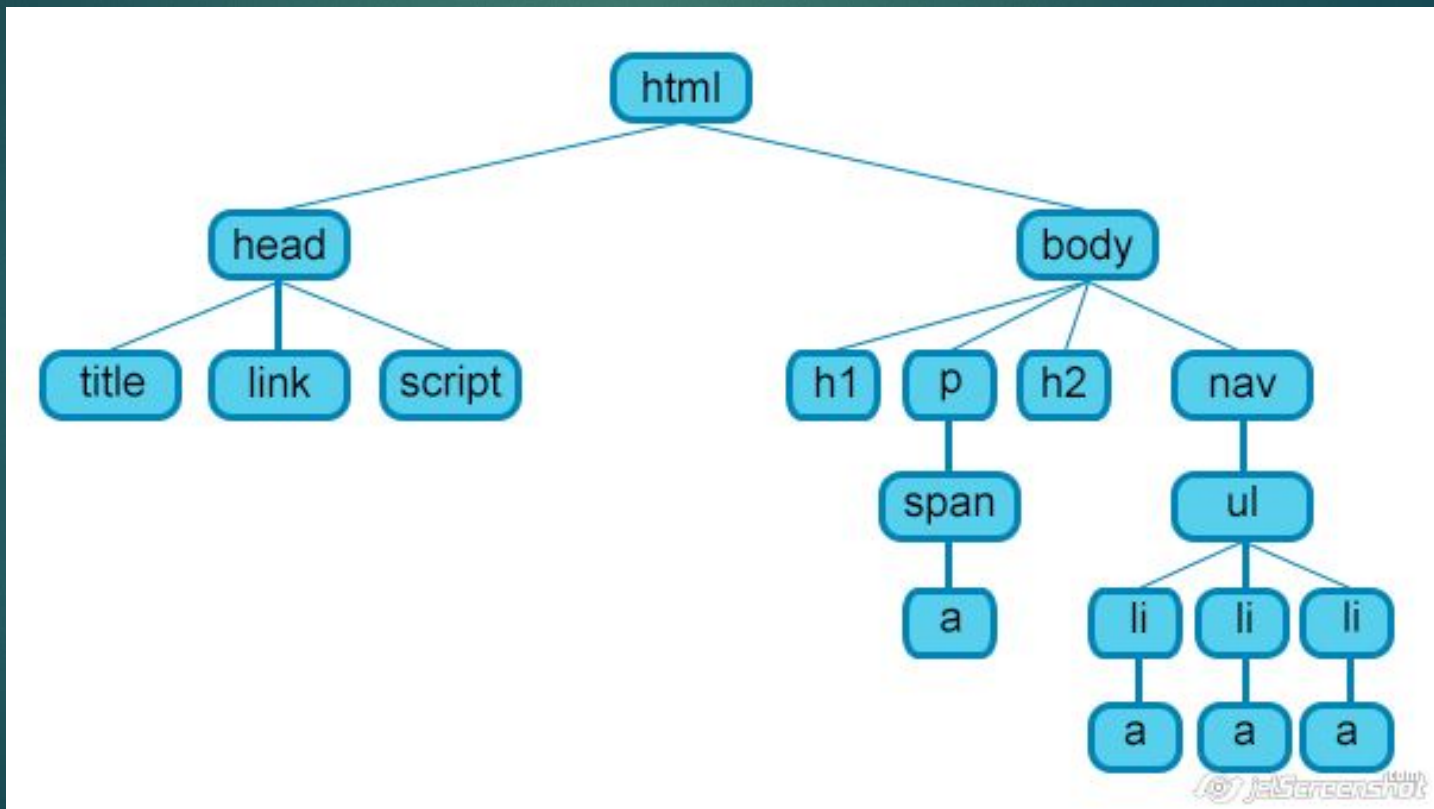
Структура страницы

```
<!DOCTYPE html> <!-- Объявление формата документа -->
<html>
<head> <!-- Техническая информация о документе -->
<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->
<title>...</title> <!-- Задаем заголовок документа -->
<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю
таблицу стилей -->
<script src="script.js"></script> <!-- Подключаем сценарии -->
</head>
<body> <!-- Основная часть документа -->
...
</body>
</html>
```


DOM

(document object model).

Элементы, находящиеся внутри тега <html>, образуют дерево документа, так называемую **объектную модель документа**.



Мета теги

Тэг `<meta>` позволяет задать метаданные для HTML документа. Мета данные не отображаются на странице, но используются поисковыми системами, браузерами, и другими веб-службами.

Кодировка символов

```
<meta charset="utf-8">
```

Информация о документе

```
<meta name="author" content="Petrov">
```

```
<meta name="copyright" content="Company Petrov">
```

Описание страницы

```
<meta name="description" content="Курс по основам создания сайтов!">
```

Description - краткое описание страницы. Данное описание используется поисковыми системами для вывода в результатах поиска, по какому либо запросу, информации о сайте и его назначении.

Ключевые слова

```
<meta name="keywords" content="обучение, сайты, html, css">
```

Keywords - ключевые слова веб-страницы, предназначены для поисковых машин.

Мета теги

Адрес

```
<meta name="Publisher-Email" content="Bau_e-mail@сервер.домен">
```

```
<meta name="Publisher-URL" content="http://www.Bau_caim/">
```

Обновление страницы

```
<meta name="revisit-after" content="15 days">
```

Команды для робота

```
<meta name="robots" content="Index, follow">
```

Данный мета тег предназначен для подачи поисковому роботу той или иной команды. Список возможных команд роботу:

Index - индексировать страницу

Noindex - не индексировать страницу

Follow - проследивать гиперссылки на странице

Nofollow - не проследивать гиперссылки на странице

All - индексировать страницу и проследивать гиперссылки на странице (по умолчанию)

None - не индексировать страницу и не проследивать гиперссылки на странице

Мета теги

Автоматический переход на другую страницу

```
<meta http-equiv="Refresh" content="10;  
URL=http://www.mysite/index.html">
```

Адаптация для ширины экрана

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Данный мета тег задает ширину страницы и начальный масштаб (в данном случае без масштабирования)

Совместимость с Internet Explorer

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
```

Данный мета тег означает, что документ будет открываться в наиболее доступном режиме браузера *IE*.

Способы добавления стилей

```
<div style="width:200px; color: red; ">
```

Информация

```
</div>
```

✓ **Внутритекстовые стили**

```
<head>
```

```
  <style>
```

```
    .box{
```

```
      width:200px;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <div class="box"> Информация </div>
```

```
</body>
```

✓ **Встраиваемые стили**

```
<head>
```

```
  <link rel="stylesheet" type="text/css"
```

```
  href="style/style.css">
```

```
</head>
```

✓ **Внешняя таблица стилей**

CSS

- ✓ **CSS (Cascading Style Sheets)**, или **каскадные таблицы стилей**, описывают правила форматирования отдельного элемента веб-страницы.
- ✓ Создав стиль один раз, его можно применять к любым элементам страницы сколько угодно раз.
- ✓ Определение стиля состоит из двух основных частей: самого элемента веб-страницы – **селектора**, и команды форматирования – **блока объявления**.
- ✓ Селектор сообщает браузеру, какой именно элемент форматировать, в блоке объявления перечисляются формирующие команды.



CSS

- ✓ **Принцип наследования** заключается в том, что свойства CSS, объявленные для элементов-предков, наследуются элементами потомками.
- ✓ **Принцип каскадирования** представляет собой процесс применения различных правил к одному и тому же элементу. Более конкретные правила имеют приоритет над более общими. Если в отношении одного и того же элемента определено несколько стилей, то в результате к нему будет применен последний из них

ПРИОРИТЕТЫ

- ✓ Наиболее низким приоритетом обладает стиль браузера.
 - ✓ Следующим по значимости является стиль, заданный пользователем браузера в его настройках.
 - ✓ И наиболее высоким приоритетом обладает стиль, заданный непосредственно автором страницы.
-
- ✓ Самым низким приоритетом обладают стили, наследуемые в документе элементом от своих предков.
 - ✓ Более высоким приоритетом обладают стили, заданные во внешних таблицах стилей, подключённых к документу.
 - ✓ Ещё более высоким приоритетом обладают стили, заданные непосредственно селекторами, содержащимися в контейнерах `style` данного документа.
 - ✓ Затем приоритетом обладают стили, объявленные непосредственно в теге данного элемента посредством атрибута `style` этого тега.
 - ✓ И наконец самым высоким приоритетом обладают стили, объявленные автором страницы или пользователем, с помощью сопроводительного слова `!important`.

Псевдоэлементы и

- ✓ **Псевдоклассы** – это селекторы, которые определяют состояние уже существующих элементов, которое может меняться при определенных условиях (например, E:hover).
- ✓ **Псевдоэлементы** – это селекторы, которые определяют область элементов, которая изначально отсутствует в дереве документа. Эта область создается искусственно с помощью CSS (например, E::first-letter).
- ✓ Ключевое отличие между ними в том, что псевдоклассы определяют именно состояние элементов, которые уже существуют на странице, а псевдоэлементы создают области (искусственные элементы), которых изначально на веб-странице не было. Но и те и другие отсутствуют в исходном коде документа.
- ✓ Разница : псевдокласс задает стиль для **элемента страницы**,

Список

ПСЕВДОЭЛЕМЕНТОВ

after - добавление контента ПОСЛЕ указанного элемента

before - добавление контента ДО указанного элемента

firstletter - стили для первой буквы в контенте элемента

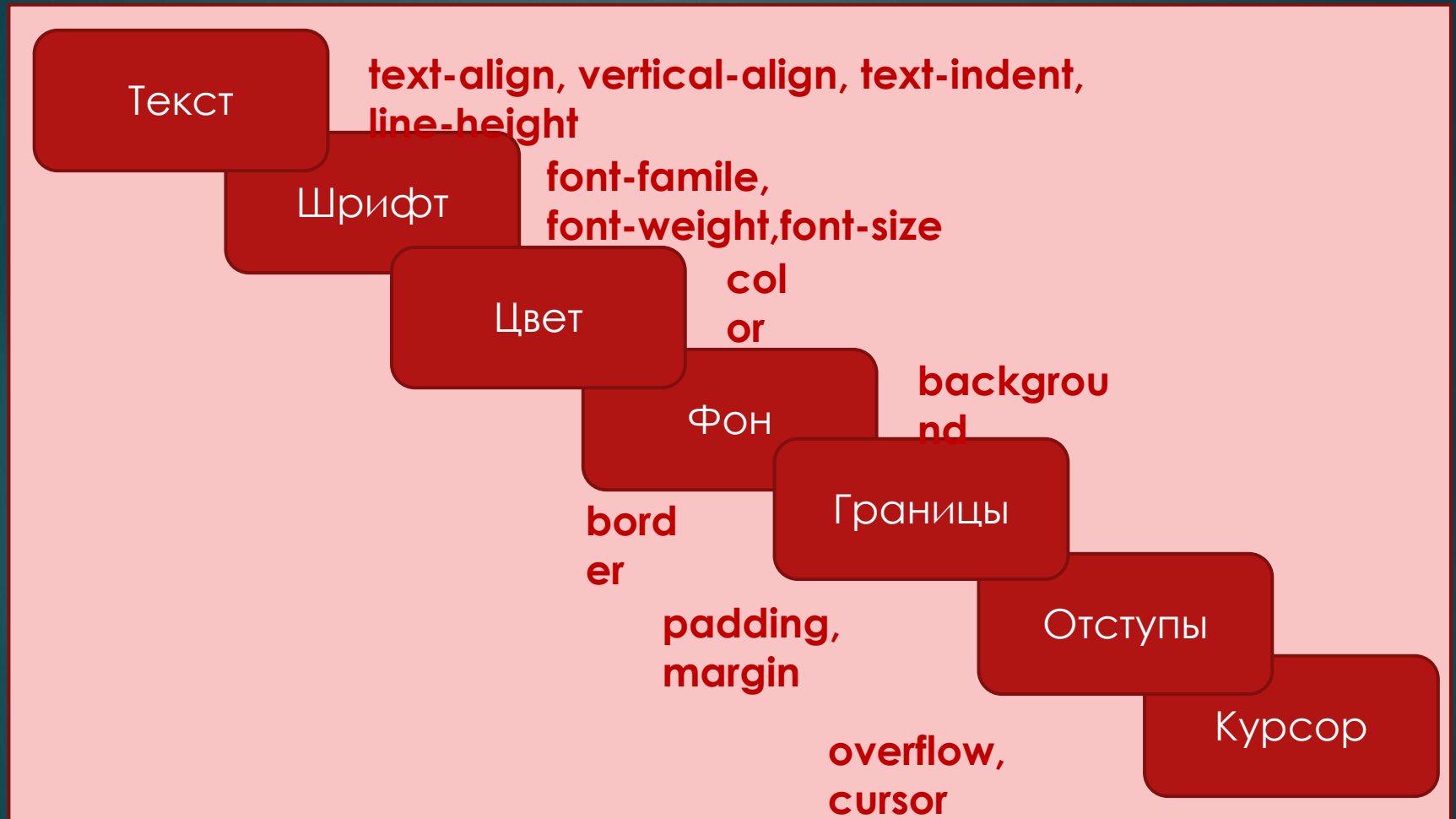
firstline - стилевое оформление первой строки текста в элементе

selection - применение стилей при выделении текста в элементе

- ✓ Одной из самых распространённых задач является добавление фразы до или после элемента.
- ✓ Псевдоэлементы **after** и **before** предназначены для "врезки" в страницу сайта контента который изначально не указан в HTML документе. Вставляется содержание перед (**:before**) или после (**:after**) какого либо элемента с помощью свойства **content**, которое собственно и определяет содержимое для вставки.

p:after {content: "Text!"; }

CSS



CSS текст

Горизонтальное выравнивание: `text-align`

Отступ: `text-indent`

Высота строк: `line-height`

Вертикальное выравнивание: `vertical-align`

Расстояние между словами: `word-spacing`

Расстояние между буквами: `letter-spacing`

Обработка пробелов: `white-space`

Настройка табуляции: `tab-size`

. Преобразование текста `text-transform`

. Направление написания текста `direction`

. Направление написания слов в тексте `unicode-bidi`

. Оформление текста `text-decoration`

. Форматирование первой буквы и первой строки абзаца `:first-letter` и `:first-line`

. Кавычки `quotes`

CSS

ТЕКСТ представляет набор CSS-стилей для форматирования текстового содержимого веб-страниц.

CSS шрифты

CSS шрифты управляют внешним видом шрифта текста веб-страниц.

Семейство шрифтов `font-family`

Стиль шрифта `font-style`

Варианты шрифтов `font-variant`

Насыщенность шрифта `font-weight`

Размер шрифта `font-size`

Цвет шрифта `color`

- ✓ Текст основного содержимого веб-страницы должен быть в первую очередь читабельным.
- ✓ Не рекомендуется использовать более двух шрифтов на странице.

Фон и фоновые изображения на веб-странице

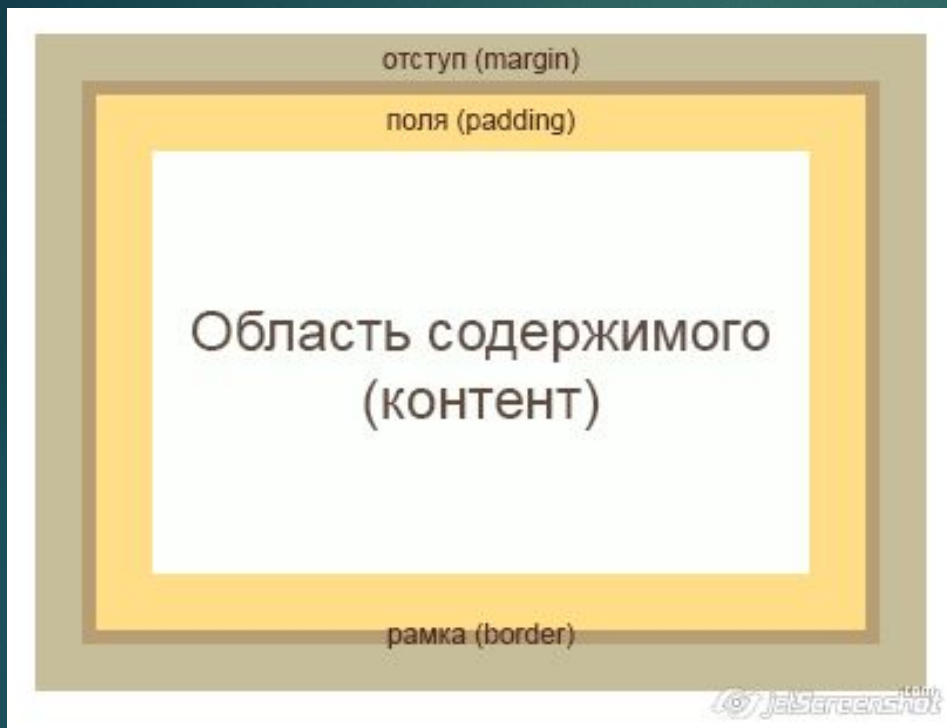
1. Цвет изображения `background-color`
2. Фоновое изображение `background-image`
3. Повтор фоновых изображений `background-repeat`
4. Позиционирование фоновых изображений `background-position`
5. Фиксация изображения на месте `background-attachment`
6. Заполнение фоном отступов и границ элемента `background-clip`
7. Положение фонового изображения относительно его родительского блока `background-origin`
8. Размер изображения `background-size`
9. Задание фона элемента одним свойством `background`

CSS границы (рамка)

1. Стиль рамки `border-style`
2. Цвет рамки `border-color`
3. Ширина рамки `border-width`
4. Задание рамки одним свойством `border`
5. Задание рамки для одной границы элемента `border-top`, `border-bottom`, `border-left`, `border-right`

- ✓ **CSS рамка** задается с помощью краткого свойства `border`
- ✓ Стиль рамки задается с помощью трех свойств: **стиль, цвет и ширина.**

CSS отступы



- ✓ **Внутренний отступ**, или поле элемента (**padding**), добавляет отступы внутри элемента, между его сторонами и содержимым, и его границей. Если для элемента задан фон, то он распространится также и на поля элемента. Внутренний отступ не может принимать отрицательных значений, в отличие от внешнего отступа.

- ✓ **Область содержимого** – это содержимое элемента.
- ✓ **Внешний отступ (margin)** добавляет отступы за границами элемента, создавая тем самым промежутки между элементами.
- ✓ Они всегда остаются прозрачными и через них виден фон родительского элемента.
- ✓ Значения padding и margin з

указываются в следующем порядке: верхнее, правое, нижнее и левое.

Курсор мыши и полоса прокрутки

- ✓ Для полосы прокрутки
- ✓ : атрибут **overflow**
 - visible** - Элемент растягивается до необходимых размеров. (по умолчанию)
 - hidden** - Содержание элемента "обрезается" видна лишь та его часть что помещается в элементе.
 - scroll** - Добавляются полосы прокрутки (всегда! даже если содержание помещается в пределах элемента).
 - auto** - Полосы прокрутки добавляются при необходимости.
- ✓ для курсора мыши: значение по умолчанию **cursor: pointer;**

Все выдохнули