

# Форматирование при помощи CSS.

ПОДГОТОВИЛА: ДАККА В.С.

# ПЛАН

- ▶ **Способы подключения CSS**
- ▶ Атрибуты `class` и `id` в `html`. Селекторы в `css`.
- ▶ Понятие блочных и строчных элементов. Примеры блочных и строчных тегов
- ▶ CSS Свойство для текста
- ▶ CSS Свойство для шрифтов
- ▶ Решение проблемы отступов между `inline-block` элементами

# CSS – каскадные таблицы стилей.

- (Cascading Style Sheets) каскадные таблицы стилей, описывают правила форматирования отдельного элемента веб-страницы.

```
10px @import
h1 {font-size: 2em;}
<link> 100%
Cascading Style
Sheets
CSS
```

. Правила записи CSS свойств.



## Способы добавления CSS

1. Встраиваемые таблицы стилей `<style>...</style>` в `<head>`.
2. Внутри текстовые таблицы стилей.
3. Правило `@import`
4. Тег `<link>`

# Пример : Встраиваемые стили.

```
1 <head>
2 <style type="text/css">
3 h1, h2 {
4 color: red;
5 font-family: "Times New Roman", Georgia,
6 Serif;
7 line-height: 1.3em;
8 }
9 </style>
10 </head>
11 <body>
12 ...
   </body>
```

## Пример : Внутри тега стили.

```
<p style="font-family: 'Times New Roman',  
Georgia, Serif; color: #70d7700;">  
Обратите внимание на этот текст.</p>
```

## Пример :Тег <link>

```
<head>  
<link rel="stylesheet" type="text/css"  
href="style/style.css">  
</head>
```

## Пример : <@import>

```
<style type="text/css">  
@import url(mobile.css);  
p {  
font-size: 0.9em;  
color: grey;  
}  
</style>
```

# Атрибуты class и id

```
<p class="name"> Text</p>
```

```
<p id="name"> Text</p>
```

Универсальный  
селектор :  
\* {margin: 0;}

# Селекторы в CSS



```
p {color:red;}
```

*Селектор*

*Свойство*

*Значение*

```
p.sermal {свойство:значение;}
```

```
.sermal {свойство:значение;}
```

```
#redtop {свойство:значение;}
```

# ПРИМЕР

Запись селектора	Пример записи в CSS	Примечание к примеру	Пример использования в HTML
Селектор тега	тег {...}	Стили применяются для элементов, у которых задан тег	<code>&lt;тег&gt;Текст&lt;/тег&gt;</code>
Класс	.класс {...}	Стили применяются для элементов, у которых задан .класс	<code>&lt;p class="класс"&gt;Текст&lt;/p&gt;</code>
Идентификатор	#иден {...}	Стили применяются для элементов, у которых задан #иден	<code>&lt;p id="иден"&gt;Текст&lt;/p&gt;</code>
Селектор атрибута	[атрибут] {...}	Стили применяются к тегам, имеющим атрибут	<code>&lt;div атрибут=""&gt;Текст&lt;/div&gt;</code>
Атрибут со значением	[атрибут="знач"] {...}	Стили применяются, если атрибут содержит "знач"	<code>&lt;a href="знач"&gt;Текст&lt;/a&gt;</code>
Один из значений атрибута	[атрибут~="знач2"] {...}	Стили применяются, если атрибут имеет "знач2"	<code>&lt;p class="red знач2"&gt;Текст&lt;/p&gt;</code>
Запись через запятую	.класс1, .класс2 {...}	Стили применяются для элементов, у которых задан .класс1 или .класс2	<code>&lt;p class="класс2"&gt;Текст&lt;/p&gt;</code>
Запись одновременно (вместе)	.класс1.класс2 {...}	Стили применяются для элементов, у которых .класс1 и .класс2 заданы одновременно	<code>&lt;p class="класс1 класс2"&gt;Текст&lt;/p&gt;</code>
Запись через пробел	#иден .класс {...}	Стили применяются к элементам имеющим .класс, который располагается внутри #иден	<code>&lt;div id="иден"&gt;&lt;p class="класс"&gt;Текст&lt;/p&gt;&lt;/div&gt;</code>
Запись через +	тег1 + тег2 {...}	Стили применяются к тегу2, если тег1 и тег2 являются соседними	<code>&lt;тег1&gt;Книга&lt;/тег1&gt; &lt;тег2&gt;Статья&lt;/тег2&gt;</code>
Запись через >	тег1 > тег2 {...}	Стили применяются к тегу2, если он является прямым дочерним элементом для тега1	<code>&lt;тег1&gt;&lt;тег2&gt;Текст&lt;/тег2&gt;&lt;/тег1&gt;</code>

# Примеры

Код HTML:

```
<h1 class="headline">Инструкция  
</h1>
```

Код CSS:

```
.headline {  
text-transform: uppercase;  
color: lightblue;}
```

```
h1 {font-family:  
Lobster, cursive;}
```

```
#sidebar {  
text-transform:  
uppercase;  
color: blue;}
```

# Горизонтальное выравнивание

Line-height

TEXT-DECORATION

**text-shadow**

text-indent

TEXT-TRANSFORM

**text-align**

CSS

Значения:

`left`

Выравнивание по левому краю элемента.

**Синтаксис:** `p {text-align: left;}`

`right`

Выравнивание по правому краю элемента.

**Синтаксис:** `p {text-align: right;}`

`center`

Выравнивание по центру элемента, управляет выравниванием содержимого, а не самих элементов. Центрирует каждую строку текста элемента.

**Синтаксис:** `p {text-align: center;}`

`justify`

Выравнивание по ширине элемента.

# Отступ

## text-indent

Смещает первую строку на заданную величину. Можно задавать как положительные, так и отрицательные значения (см. пример ниже).

**Синтаксис:** `p {text-indent: 5px;}`

Величина смещения задаётся в процентах.

**Синтаксис:** `p {text-indent: 2%;}`



## Высота строк

# ПРИМЕР:

Межстрочный интервал позволяет управлять расстоянием между строками текста. В данном примере при размере шрифта `{font-size: 1em}` задан межстрочный интервал `{line-height: 0.8em}`.

Для данного абзаца при размере шрифта `{font-size: 1em}` задан межстрочный интервал `{line-height: 1}`.

В данном абзаце при размере шрифта `{font-size: 1em}` установлен межстрочный интервал `{line-height: normal}`.

Оптимальным значением межстрочного интервала при размере шрифта `{font-size: 1em}` будет `{line-height: 1.4em}`.

# Вертикальное выравнивание

## `vertical-align:`

```
img {vertical-align:
baseline;}
img {vertical-align: sub;}
img {vertical-align: super;}
img {vertical-align: top;}
img {vertical-align: text-top;}
img {vertical-align: middle;}
img {vertical-align: bottom;}
img {vertical-align:
text-bottom;}
img {vertical-align: 5px;}
```

Изображение  `{vertical-align: baseline}` не имеет базовой линии, поэтому оно выравнивается по базовой линии родителя (в данном случае, текста).

Этот абзац содержит **надстрочный** `{vertical-align: super}` и **подстрочный** `{vertical-align: sub}` текст, который имеет такой же размер, как и основной текст.

Изображение  `{vertical-align: bottom}` выравнивается по низу контейнера строки, смещаясь ниже базовой линии текста.

Изображение  `{vertical-align: text-bottom}` выравнивается относительно нижней линии текста строки.

Изображение  `{vertical-align: top}` выравнивается по верху контейнера строки, при этом учитывается размер шрифта родительского элемента.

В данном примере изображение  `{vertical-align: text-top}` выравнивается относительно верхней линии текста строки.

Значение  `{vertical-align: middle}` выравнивает середину изображения по точке, которая находится на расстоянии, равном половине размера шрифта родительского элемента, над базовой линией текста.

# Немного практики



# Расстояние между словами

word-spacing

white-space

tab-size

word-spacing

letter-spacing

CSS

Расстояние между

буквами

Обработка пробелов

letter-spacing

white-space

white-space : normal; | nowrap; | pre; |  
pre-wrap; (делает переносы там, где нужно)  
pre-line; (удаляет лишние пробелы, кроме <br/>)

Преобразование текста

text-transform : none; | capitalize; |

uppercase;

lowercase;

Направление написания текста direction: ltr;

rtl;

# Декорирование текста

## `text-decoration:`

`none ; | underline;| overline;|  
line-through;`

### **Выделение первой буквы**

`:first-letter`

### **и первой строки**

`:first-line`

# Форматирование шрифта с помощью CSS-свойств

Семейство шрифтов `font-family`

Значения: `h1 {font-family: Times, 'New Century Schoolbook', serif;}`

`family-name` Название (имя) семейства шрифтов. Рекомендуется указывать вместе с базовым семейством.

`generic-family` Базовое семейство. CSS определяет пять базовых семейств шрифтов:  
Шрифты с засечками — Serif (Times New Roman, Times, Garamond, Georgia)  
Рубленые шрифты — Sans-serif (Helvetica, Geneva, Arial, Verdana, Trebuchet, Univers)  
Моноширинные шрифты — Monospace (Courier, Courier New, Andale Mono)  
Рукописные шрифты — Cursive (Comic Sans, Gabriola, Monotype Corsiva, Author, Zapf Chancery)  
Аллегорические шрифты (Western, Woodblock, Kluge)

**Стиль шрифта `font-style: normal;`  
`italic;`  
`oblique;`**

*устанавливает для текста обычный шрифт.*

*выделяет текст курсивом.*

*} устанавливает наклонное начертание шрифта.*

# Насыщенность шрифта font-weight

Варианты значений:

normal(400);

Bold (700);

Bolder(больше , чем у предка);

Lighter (меньше);

**Размер шрифта font-size:14 px | pt | cm| mm in| em| %;**

`h1 {font-weight: normal;}` устанавливает обычный шрифт. Эквивалентно значению насыщенности, равной 400.

`h1 {font-weight: bold;}` делает шрифт текста полужирным. Эквивалентно значению насыщенности, равной 700.

Значение `h1 {font-weight: 100;}` соответствует самому легкому варианту начертания шрифта.

**Значение `h1 {font-weight: 900;}` соответствует самому плотному варианту начертания шрифта.**

```
p {font: italic bold 14px/1.5 Times, 'New Century Schoolbook' serif;}
```

# ПРАКТИКА

1. Изменяя стиль у каждого абзаца (цвет и размер шрифта) создайте такую страницу.

этот абзац 14 пунктов, зеленый  
серый, 14 пикселей

Это просто размер номер 5, без стиля  
серебряный, 150% от текущего 5

Это просто размер номер 2, без стиля  
серебряный, 150% от текущего 2

красный, 1

дюйм

синий, 1.5 сантиметра

2. Задайте внутри элемента HEAD стиль для тега H2 – синий цвет, и тега P – красный цвет. Далее создайте такую страницу.

Это заголовок второго уровня

это написано в абзаце

это написано вне абзаца

это написано в абзаце, где дополнительно применен стиль - размер шрифта 10 пунктов

а это заголовок второго уровня, где

дополнительно применен стиль - размер шрифта 13 пунктов

# Домашнее задание