



# HyperText Markup Language

# HTML&CSS

Lesson 3

Таблицы и списки

# HyperText Markup Language

## Создание таблицы, тег Table

Тег <Table> является основным контейнером, в который помещается таблица. Именно в нем указываются все атрибуты для таблицы и размещается вся разметка таблицы.

```
<table border="1">  
  <tr>  
    <td>  
      100  
    </td>  
  </tr>  
</table>
```

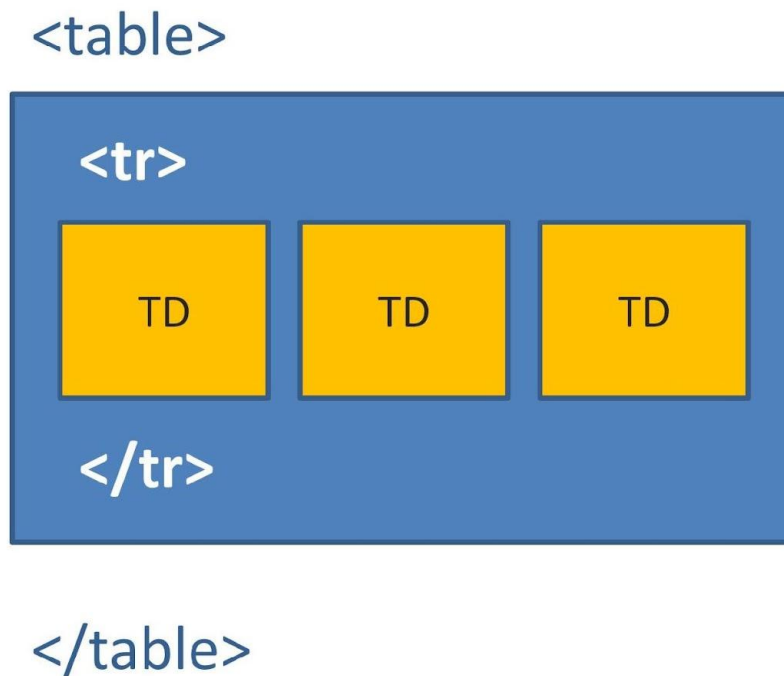
Таблица часто используется для визуализации табличных данных, а также применяется для реализации верстки определенных страниц. Иногда весь каркас страницы строится с помощью таблиц, но в последнее время эта техника применяется редко.

# HyperText Markup Language

## Понятие строки и ячейки, теги <tr>, <td>

Таблица состоит из строк (тег <tr> ... </tr> ), которые в свою очередь состоят из ячеек (тег <td> ... </td> ).

```
<table border="1">
  <tr>
    <td>
      100
    </td>
    <td>
      200
    </td>
    <td>
      300
    </td>
  </tr>
</table>
```



# HyperText Markup Language

## Заглавные ячейки <th>

<th></th> - тег, который описывает специальный вид ячеек, заглавных ячеек.

При этом текст внутри этих ячеек считается более важным, чем простых, и приобретает особые свойства – текст центрируется и получает полужирное начертание.

```
<table border="1">
  <tr>
    <th>Имя</th>
    <th>Телефон</th>
    <th>E-mail</th>
  </tr>
</table>
```

# HyperText Markup Language

## Название таблицы, тег <caption>

Тег `<caption></caption>` - название таблицы, которое размещается сверху над таблицей и выравнивается по центру.

Сам тег следует разместить сразу после открывающего тега `<table>`.

### Данные пользователей

Имя	Фамилия	Телефон	E-mail

# HyperText Markup Language

## Пустые ячейки

Не следует оставлять пустые ячейки, так как в некоторых браузерах они будут отображены некорректно и впоследствии повлияют на визуализацию всей таблицы.

Если Вам действительно необходимо оставить их пустыми, разместите там пробел, указав его код `&nbsp;`.

Some text	Some text
	Some text

Пустая ячейка

Some text	Some text
	Some text

Заполненная пробелом

# HyperText Markup Language

## Атрибуты тега <table>

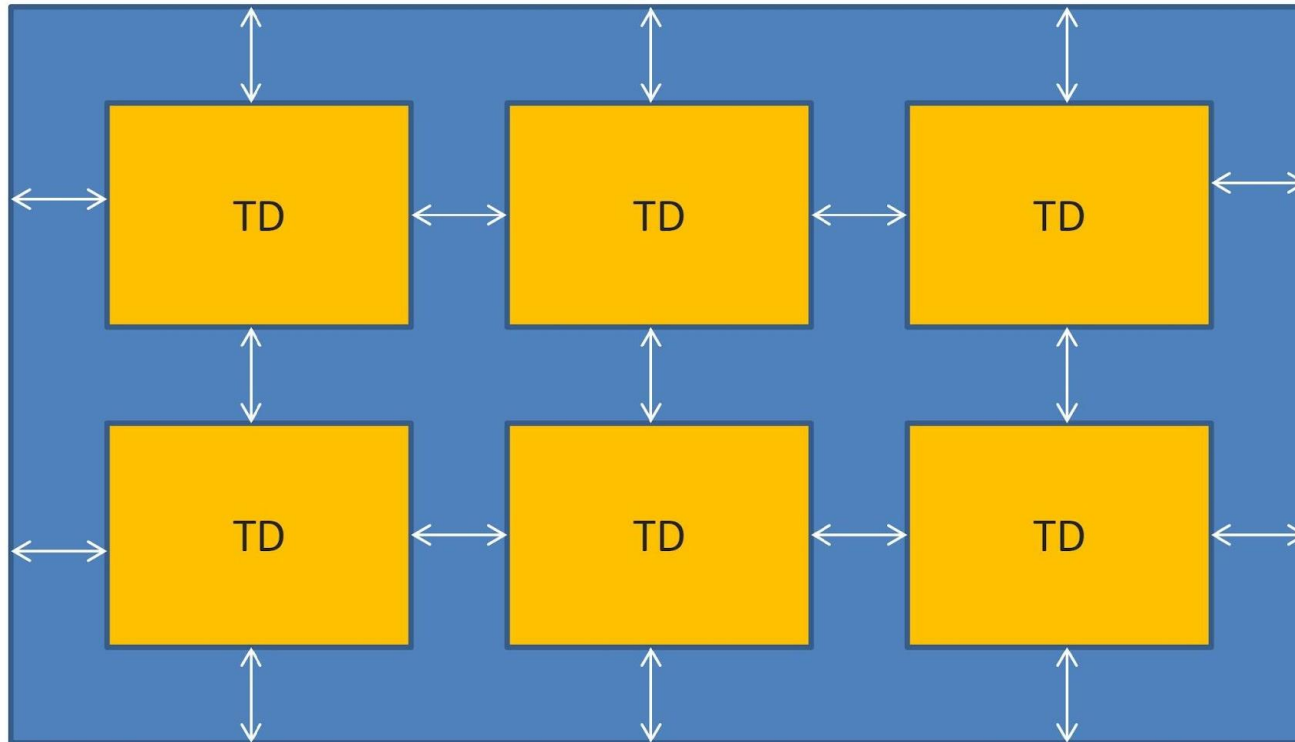
Тег `<table>`, как и другие теги, имеет свои особенные атрибуты:

1. **Border** – рамка таблицы, как вокруг нее, так и внутри между ячеек.
2. **Cellpadding** – отступы внутри ячеек.
3. **Cellspacing** – отступы внутри таблицы между ячейками и до границ таблицы.
4. **Bgcolor** – цвет фона таблицы.
5. **Background** – изображение, которое будет использовано в качестве фона таблицы.

# HyperText Markup Language

## Атрибут cellpadding

Атрибут таблицы **cellpadding** говорит об отступах между ячейками таблицы и отступах от границ ячеек до границ самой таблицы.

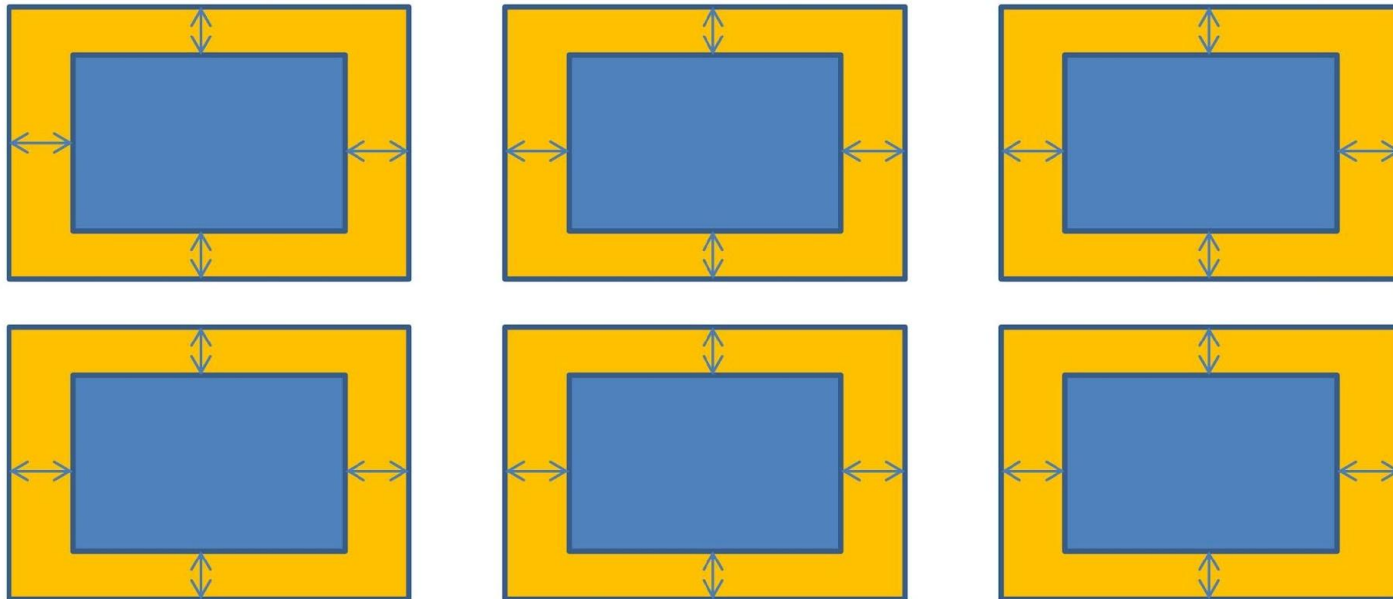




# HyperText Markup Language

## Атрибут cellpadding

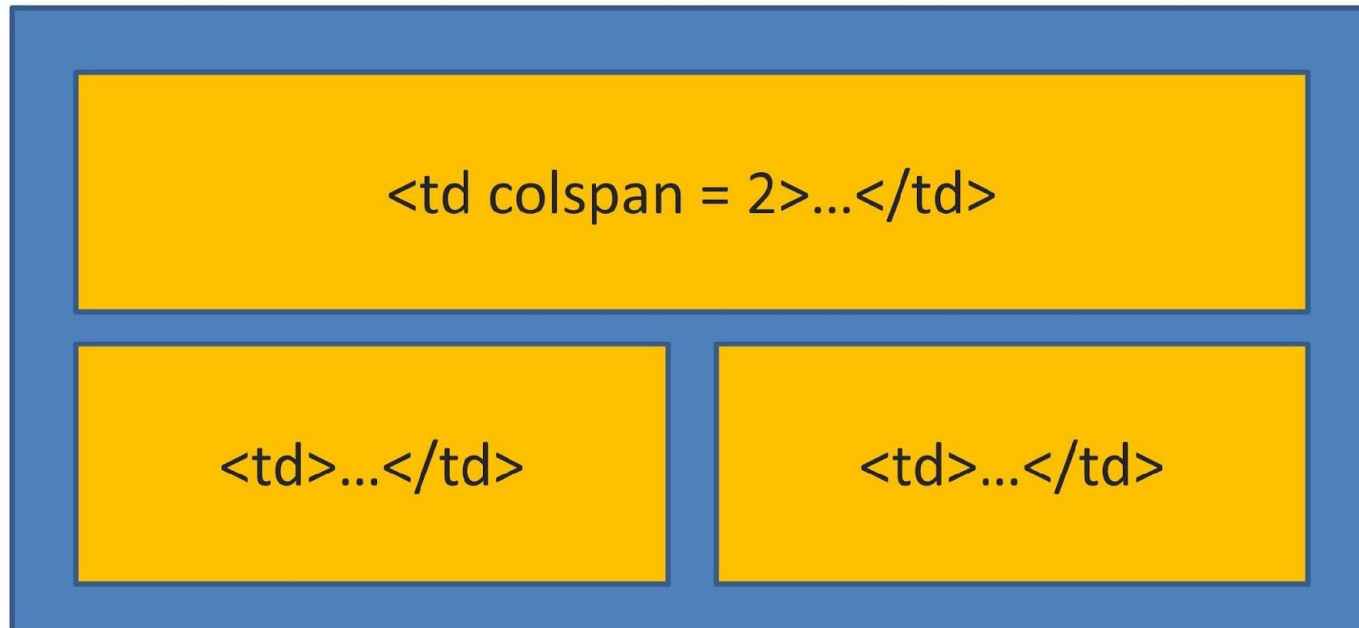
Атрибут таблицы **cellpadding** говорит об отступах внутри ячеек, то есть отступы от границ ячеек до содержимого.



# HyperText Markup Language

## Атрибут colspan

Атрибут **colspan** устанавливает число ячеек, которые должны быть объединены по горизонтали. Этот атрибут имеет смысл для таблиц, состоящих из нескольких строк.



# HyperText Markup Language

## Атрибут rowspan

Атрибут **rowspan** устанавливает число ячеек, которые должны быть объединены по вертикали. Этот атрибут имеет смысл для таблиц, состоящих из нескольких строк.

```
<td rowspan = "2"></td>
```

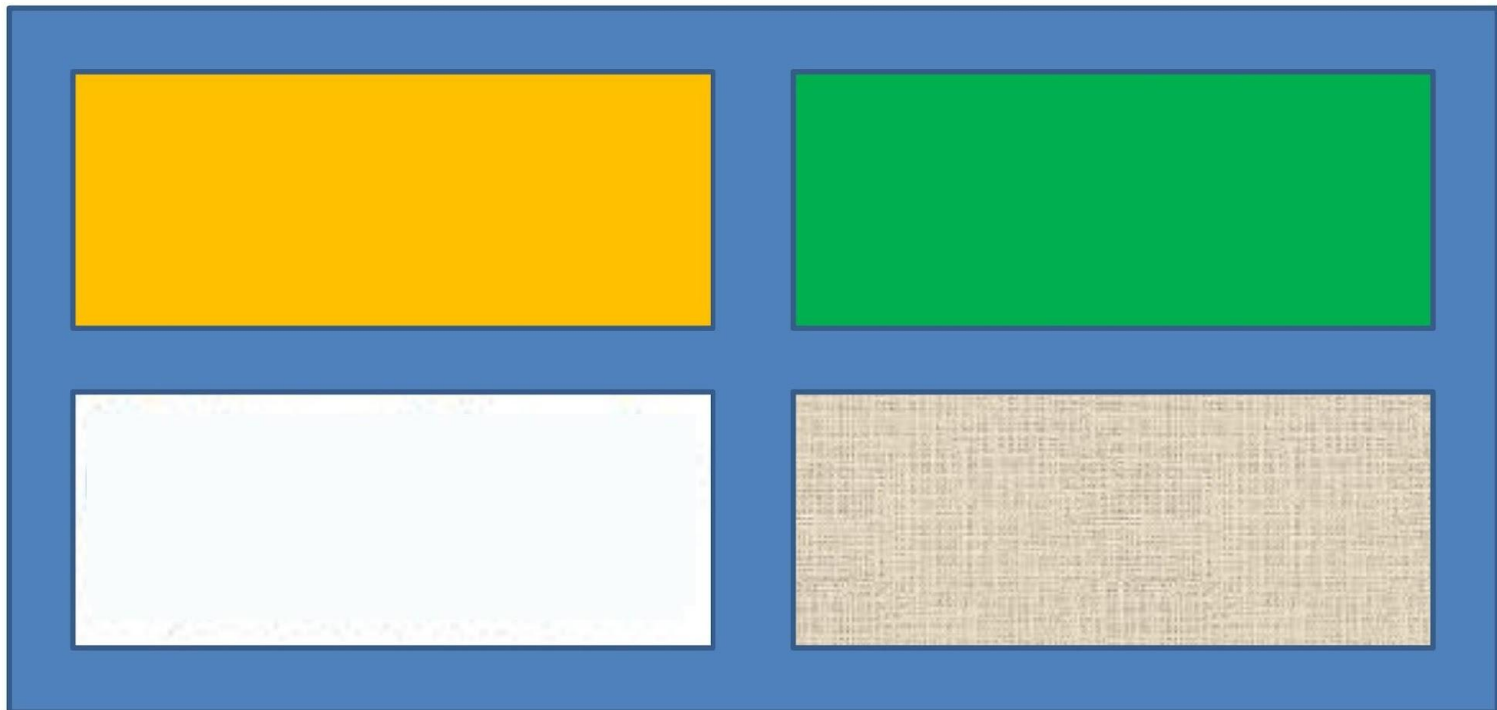
```
<td>...</td>
```

```
<td>...</td>
```

# HyperText Markup Language

## Использование фона в таблицах

С помощью атрибутов **bgcolor**, **background** можно установить фоновый цвет либо фоновый рисунок как таблицы, так и конкретной ячейки.



## Списки Lists

# HyperText Markup Language

## Упорядоченные списки

Тег `<ol></ol>` является контейнером для упорядоченных списков, так и расшифровывается - ordered list.

Все элементы списка заключаются внутри тегов `<li>...</li>`, которые в свою очередь помещаются в общий контейнер `<ol></ol>`.

`<ol>`

`<li>Пункт №1</li>`

`<li>Пункт №2</li>`

`<li>Пункт №3</li>`

`<li>Пункт №4</li>`

1. Пункт №1

2. Пункт №2

3. Пункт №3

4. Пункт №4

`</ol>`

# HyperText Markup Language

## Атрибут `type` в упорядоченных списках

У всех списков есть атрибут `type`, который указывает тип маркера. У упорядоченных списков он может принимать следующие значения:

1. **A** – указывает на то, что пункты списка будут нумероваться большими латинскими буквами.
2. **a** – указывает на то, что пункты списка будут нумероваться маленькими латинскими буквами.
3. **I** – указывает на то, что пункты списка будут нумероваться большими римскими цифрами.
4. **i** – указывает на то, что пункты списка будут нумероваться маленькими римскими цифрами.

# HyperText Markup Language

## Атрибут type в неупорядоченных списках

- A. Пункт №1
- B. Пункт №2
- C. Пункт №3

- a) Пункт №1
- b) Пункт №2
- c) Пункт №3

- I. Пункт №1
- II. Пункт №2
- III. Пункт №3

`<ol type="A">...</ol>`

`<ol type="a">...</ol>`

`<ol type="I">...</ol>`

- i. Пункт №1
- ii. Пункт №2
- iii. Пункт №3

По умолчанию список будет нумероваться арабскими цифрами:

`<ul type="i">...</ul>`

- 1. Пункт №1
- 2. Пункт №2
- 3. Пункт №3



# HyperText Markup Language

## Неупорядоченные списки

Тег `<ul></ul>` является контейнером для неупорядоченных списков, так и расшифровывается - unordered list.

Все элементы списка заключаются внутри тегов `<li>...</li>`, которые в свою очередь помещаются в общий контейнер `<ul></ul>`.

`<ul>`

`<li>Пункт №1</li>`

•Пункт №1

`<li>Пункт №2</li>`

•Пункт №2

`<li>Пункт №3</li>`

•Пункт №3

`<li>Пункт №4</li>`

•Пункт №4

`</ul>`

# HyperText Markup Language

## Атрибут `type` в неупорядоченных списках

У всех списков есть атрибут `type`, который указывает тип маркера. У неупорядоченных списков он может принимать следующие значения:

1. **Disc** – указывает маркер в форме черного закрашенного круга.
2. **Circle** – указывает маркер в форме пустого круга.
3. **Square** – указывает маркер в форме черного закрашенного квадрата.
4. **None** – указывает, что список необходимо реализовать без маркера.
5. Свой маркер можно установить указав в атрибуте `style` свойство `list-style-image:url()`, указав в нем адрес изображения, которое будет использовано в качестве маркера.

# HyperText Markup Language

## Атрибут type в неупорядоченных списках

- Пункт №1
- Пункт №2
- Пункт №3

- Пункт №1
- Пункт №2
- Пункт №3

- Пункт №1
- Пункт №2
- Пункт №3

`<ul type="disc">...</ul>`

`<ul type="circle">...</ul>`

`<ul type="square">...</ul>`

- Пункт №1
- Пункт №2
- Пункт №3

`<ul style="list-style-image:url(Image1.jpg)">...</ul>`

# HyperText Markup Language

## Вложенные списки

```
<ul>  
  <li>Coffee</li>  
  <li>Tea  
    <ul>  
      <li>Black tea</li>  
      <li>Green tea</li>  
    </ul>  
  </li>  
  <li>Milk</li>  
</ul>
```

В элементы списка можно вкладывать не только текстовое содержимое, но и другие списки.

Таким образом будут реализованы вложенные списки, а точнее подсписки.

При этом по умолчанию во вложенных списках маркер будет отличным от маркера главного списка.

# HyperText Markup Language

## Списки определений

Есть специальные списки для определений, для этого следует использовать контейнер `<dl>...</dl>`.

Элементы таких списков состоят из двух составляющих: первая – это сам термин, заключенный в тег `<dt>...</dt>`, второй – само определение, заключенное в тег `<dd>...</dd>`.

```
<dl>  
  <dt>Кофе</dt>  
  <dd>Черный горячий напиток</dd>  
  <dt>Молоко</dt>  
  <dd>Белый холодный напиток</dd>  
</dl>
```

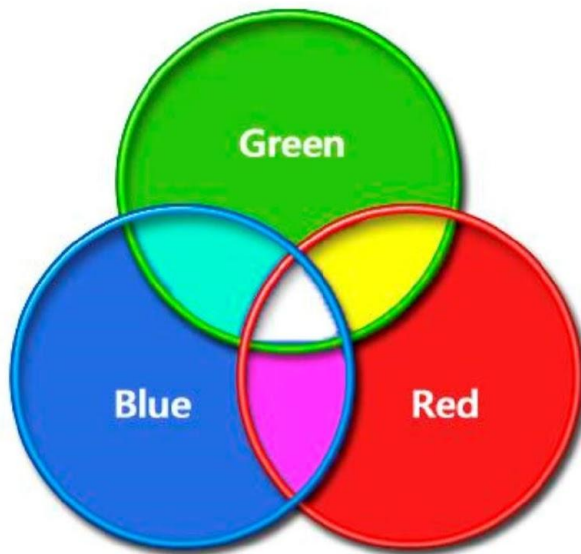
# HyperText Markup Language





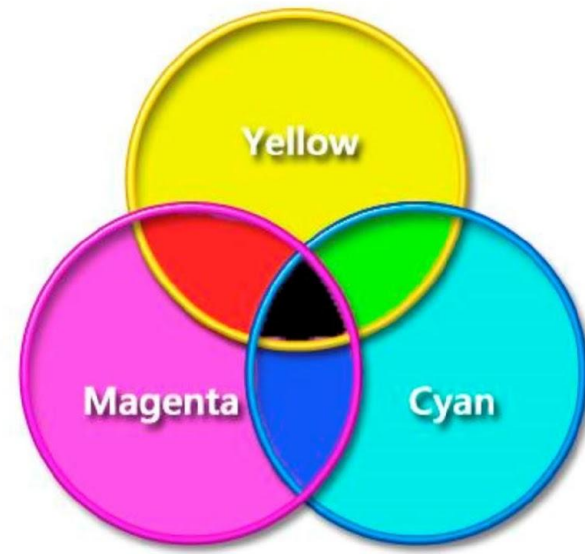
# HyperText Markup Language

## Цветовые модели



**RGB**

Red Green Blue



**CMYK**

Cyan Magenta Yellow Key color (black)

# HyperText Markup Language

## Задание цвета в разметке

Задать цвет в разметке (будь то цвет фона, рамки или текста) можно тремя способами:

1. Указав цвет соответствующей семантической единицей:

```
<body style="background-color: Blue">
```

2. Указав цвет через его RGB значения насыщенности:

```
<body style="background-color: rgb(123, 12, 0)">
```


3. Указав шестнадцатеричный код цвета:

```
<body style="background-color: #cccccc">
```



# HyperText Markup Language

## Цвет - семантическая единица

Color Name	Color HEX	Color
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
Coral	#FF7F50	
CornflowerBlue	#6495ED	
Cornsilk	#FFF8DC	
Crimson	#DC143C	
Cyan	#00FFFF	
DarkBlue	#00008B	
DarkCyan	#008B8B	
DarkGoldenRod	#B8860B	
DarkGray	#A9A9A9	
DarkGreen	#006400	
DarkKhaki	#BDB76B	

Многие цвета на практике часто используются и потому имеют конкретные имена.

Но запомнить их все человеку сложно, поэтому лучше всего пользоваться только теми семантическими единицами, которые Вы уже использовали ранее, и точно знаете как они будут отображены.

Либо Вы можете пользоваться специальными таблицами, в которых указаны имена цветов.

# HyperText Markup Language

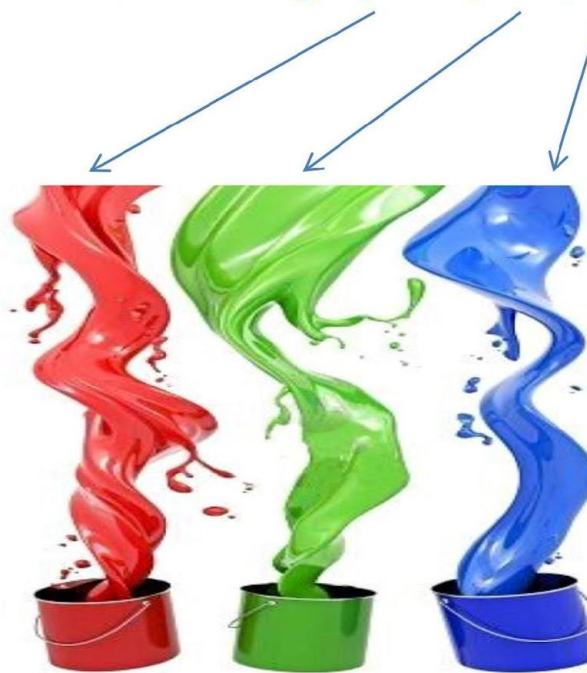
## Цвет, заданный через rgb()

Цвет в RGB формируется в результате смешения этих трех цветов, и Вы, как верстальщик, можете попробовать составить свой цвет, указав явно, какую насыщенность какого цвета Вы хотели бы использовать:

```
<body style="background-color: rgb(123, 12, 0)">
```

При этом насыщенность следует указывать в размерности от 0 до 255.

В результате Вы получите максимально точно подобранный цвет.



# HyperText Markup Language

## Цвет – шестнадцатеричный код

Цвет также можно задать с помощью шестнадцатеричного кода. Обычно он указывается следующим образом:

```
<body style="background-color: #cccccc">
```

Рассмотрим подробнее сам код:

**#E3 CC 25**

**R G B**

Шестнадцатеричный код является все тем же указанием насыщенности каждого цвета, но уже в шестнадцатеричном формате, где каждый цвет задается комбинацией двух символов от 0 до F.

В результате Вы получаете все те же от 0 до 255 значений насыщенности каждого цвета, но запись получается короче.