

Каскадные таблицы стилей CSS

Зачем это?

CSS предназначен для форматирования веб-страниц



Структура кода

```
Селектор {  
Свойство: Значение;  
}
```

Способы применения CSS кода к html страницам

1) In-line

при помощи атрибута style

```
<p style="color: #ff0000;" >
```

Предназначен для оформления одного элемента на одной странице

Способы применения CSS кода к HTML страницам

2) Внутренний при помощи тэга style

```
<head>  
<style>  
p {  
color: #ff0000;  
}  
</style>  
</head >
```

Предназначен для оформления
нескольких элементов на одной странице

Способы применения css кода к html страницам

3) Внешний

при помощи внешнего файла *.css (style.css)

html файл:

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

css файл:

```
p {  
  color: #ffffff;  
}
```

Предназначен для оформления нескольких элементов на нескольких страницах

1. ΦΟΗ

Background

1) `background-color` – цвет фона

Значения: 16-ый код цвета (`#ff0000`) или английское название цвета (`red`).

Пример:

```
p {  
  background-color: #ff0000;  
}
```

Background

2) `background-image` – фоновое изображение

Значение: url-адрес изображения форматов
.jpg, .gif, .png - `url(“img/bgd.jpg”);`

Пример:

```
body {  
    background-image: url(“flame.jpg”);  
}
```


Background

3) `background-repeat` — повторение фонового изображения

Значения: `repeat` — повторять по обеим осям;
`repeat-x` — повторять по оси x;
`repeat-y` — повторять по оси y;
`no-repeat` — не повторять.

Пример:

```
body {  
    background-image: url("flame.jpg");  
    background-repeat: no-repeat;  
}
```

Background

4) `background-attachment` — фиксация фонового изображения

Значения: `scroll` — прокручивать вместе с содержимым;

`fixed` — зафиксировать.

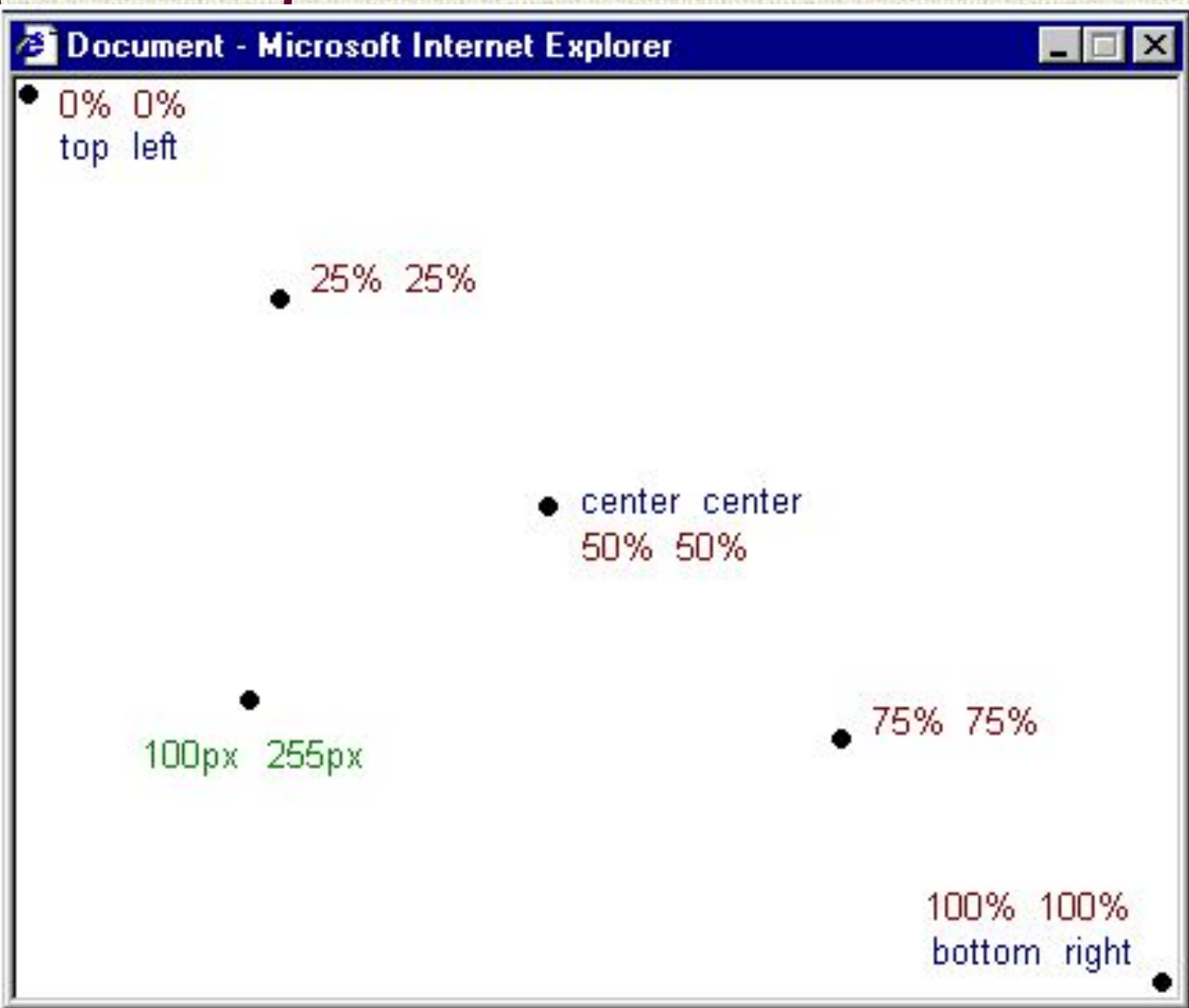
Пример:

```
body {  
    background-image: url("flame.jpg");  
    background-attachment: fixed;  
}
```

Васи

5) bas
угла ф
Значе
- В С
- КЛ
center

Пр



ХНЕГО

%:

right,

}

Background

```
body {  
    background-color: #ff0000;  
    background-image: url("flame.jpg");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: 100px 200px;  
}
```

```
body {  
    background: #ff0000 url("flame.jpg") no-repeat fixed  
100px 200px;  
}
```

2. Шрифт

Font

1) `font-style` – курсивный шрифт

Значения: `italic` – курсив;

`normal` – обычный.

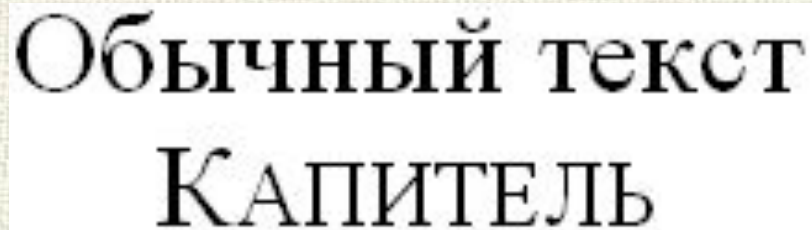
Пример:

```
h2 {  
    font-style: italic;  
}
```

Font

2) `font-variant` – начертание строчных букв

Значения: `normal` – обычный;
`small-caps` – капитель.



Обычный текст
КАПИТЕЛЬ

Пример:

```
h2 {  
  font-variant: small-caps;  
}
```

Font

3) `font-weight` – насыщенность букв

Значения: `normal` – обычный;
`bold` – насыщенный.

Пример:

```
h2 {  
    font-weight: bold;  
}
```


Font

4) `font-size` – размер букв

Значения: `pt` – пункты;
`px` – пиксели;
`%` – проценты.

Пример:

```
p {  
  font-size: 14pt;  
}
```

Font

5) font-family – тип и семейство шрифта

Times New Roman
Garamond
Georgia

Эти три шрифта принадлежат к общему семейству serif. У них имеются т. н. "засечки".

Trebuchet
Arial
Verdana

Эти три шрифта принадлежат к общему семейству sans-serif. У них нет "засечек".

Courier
Courier New
Andale Mono

Эти три шрифта принадлежат к общему семейству monospace. Символы этих шрифтов имеют одинаковую ширину (т. н. "моноширинные шрифты").

Пример:

p {

font-family: arial, verdana, sans-serif;

}

Font

```
h1 {  
    font: italic normal bold 30px arial,  
    sans-serif;  
}
```

3. Текст

Текст

1) `text-indent` – красная строка

Значения: `pt` – пункты;

`px` – пиксели;

`%` – проценты.

Пример:

```
p {  
  text-indent: 30px;  
}
```

Текст

1) `text-align` – выравнивание текста по ширине

Значения: `left`, `right`, `center`, `justify`

Пример:

```
p {  
    text-align: justify;  
}
```

Текст

1) `text-decoration` — оформление подчеркивания

Значения: `none` — отменяет все эффекты

`underline` — подчеркнутый

`overline` — надчеркнутый

`line-through` — ~~зачеркнутый~~

Пример:

```
a {  
  text-decoration: none;  
}
```

Текст

1) `text-transform` – управление регистром

Значения:

`none` – не меняет регистр

`lowercase` – нижний регистр

`uppercase` – ВЕРХНИЙ РЕГИСТР

`capitalize` – Начинать С Прописных

Пример:

```
b {  
  text-transform: uppercase;  
}
```


4. Типы селекторов

Типы селекторов

- 1) Теги
- 2) Классы
- 3) Идентификаторы



Классы

Предназначены для оформления групп сходных элементов. К одному тегу может применяться несколько классов. Для создания в html-коде используют атрибут class.

html:

```
<p class="one"> Адын </p>
```

```
<p class="two" class="one"> Дыва </p>
```

css:

```
p.one { color: #ff0000; }
```

```
.two { font-size: 20pt; }
```

Идентификаторы

Предназначены для оформления уникальных элементов. К одному тегу может применяться только 1 идентификатор. Для создания в html-коде используют атрибут id.

html:

```
<p id="one"> Адын </p>
```

```
<p id="two" > Дыва </p>
```

CSS:

```
#one { color: #ff0000; }
```

```
#two { font-size: 20pt; }
```

5. Псевдоклассы. Ссылки.

Что это?

Псевдоклассы определяют динамическое состояние элементов, которое изменяется с помощью действий пользователя, а также положение в дереве документа.

Селектор: Псевдокласс { Описание правил стиля }



Псевдоклассы

1) :active

Происходит при активации пользователем элемента. Используется преимущественно для ссылок.

Пример:

```
a:active {  
  text-transform: uppercase;  
}
```

Псевдоклассы

2) :link

Применяется к непосещенным ссылкам.

Запись `A {...}` и `A:link {...}` по своему результату равноценна, поскольку в браузере даёт один и тот же эффект, поэтому псевдокласс `:link` можно не указывать.

Исключением являются якоря, на них действие `:link` не распространяется.

Пример:

```
a:link {  
  text-transform: uppercase;  
}
```


Псевдоклассы

3) :hover

Активизируется, при наведении курсора мыши на элемент, но щелчка по нему не происходит.

Пример:

```
a:hover {  
  text-decoration : underline;  
}
```

Псевдоклассы

4) :visited

Применяется к посещенным ссылкам.

Обычно такая ссылка меняет свой цвет по умолчанию на фиолетовый, но с помощью стилей цвет и другие параметры можно задать самостоятельно

Пример:

```
a:visited {  
  color: #ffff00;  
}
```

Псевдоклассы

5) :first-child

Применяется к первому дочернему элементу селектора, который расположен в дереве элементов документа.

Пример:

```
p:first-child {  
  color: red;  
}
```

Псевдоэлемент

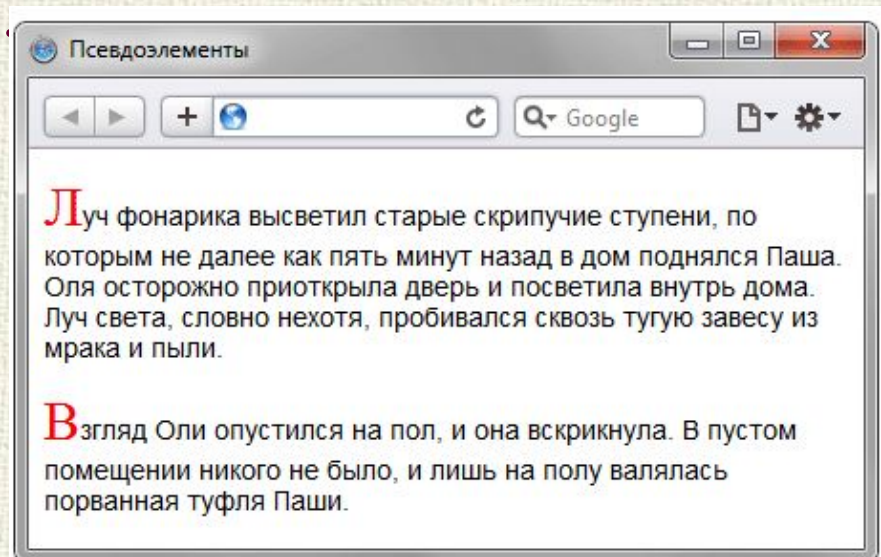
Псевдоэлементы позволяют задать стиль элементов не определённых в дереве элементов документа, а также генерировать содержимое, которого нет в исходном коде текста.

6) :first-letter

Определяет стиль первого символа в тексте элемента.

Это позволяет создавать в тексте буквицу и выступающий инициал.

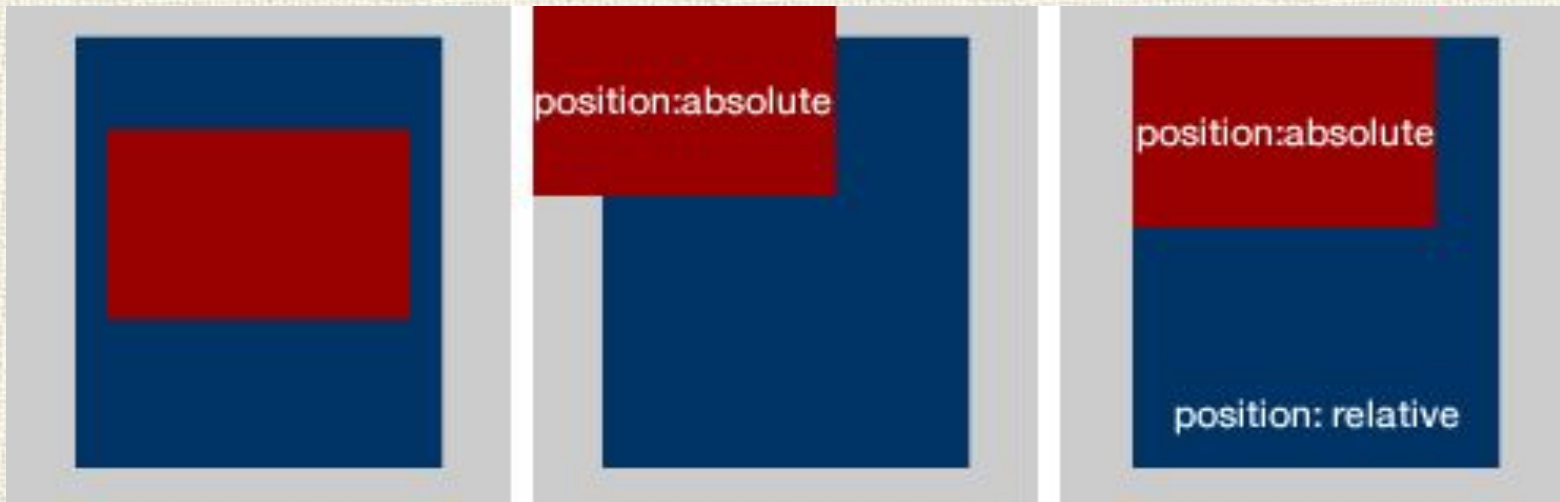
Буквица представляет собой увеличенную первую букву, базовая линия которой ниже на одну или несколько строк базовой линии основного текста. Выступающий инициал — увеличенная прописная буква, базовая линия которой совпадает с базовой линией основного текста.



6. **Позиционирование элементов**

Как делать?

Позиционирование (расположение) элементов задается свойствами: `position` (тип позиционирования), а также свойствами `top` (`bottom`) и `left` (`right`), задающими отступы.



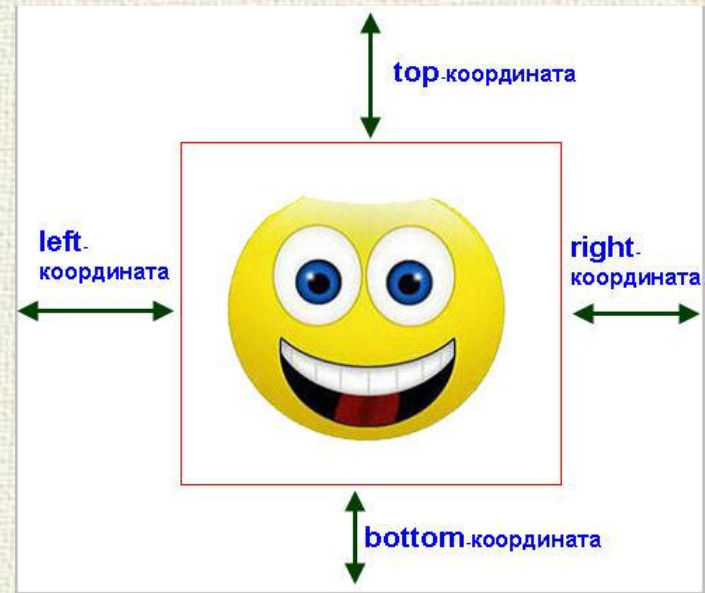
Виды позиционирования

1) position: absolute;

Абсолютное позиционирование. Точно располагает элемент внутри окна браузера или родительского элемента с заданным позиционированием (fixed, relative, absolute).

Пример!

```
#banner {  
    position: absolute;  
    top: 15px;  
    left: 15px;  
}
```



Виды позиционирования

2) position: relative;

Относительное позиционирование. Изменяет расположение элемента относительно его прежнего положения.

Пример:

```
img:hover {  
  position: relative;  
  top: 15px;  
  left: 15px;  
}
```



Виды позиционирования

3) position: fixed;

Фиксированное позиционирование. Привязывает элемент к точке в окне браузера и не меняет его положения при прокрутке страницы.

Пример:

```
#logo {  
  position: fixed;  
  top: 15px;  
  left: 15px;  
}
```



Виды позиционирования

4) `position: static;`

Значение по умолчанию.

Элементы не имеющие позиционирования (или родительских элементов с позиционированием) располагаются на странице игнорируя положение позиционированных элементов.

**7. Блочные и
строчные элементы.
Создание блоков.
Верстка.**

Сколько нужно шлакоблоков, чтоб дворец построить в срок?

Типы элементов:

- 1) Строчные – являются частью строки, не обособляются отступами, переносятся на другую строку при необходимости (примеры, ``, ``, `<a>`, ``, ``).
- 2) Блочные – размечается в виде прямоугольника, занимает всю доступную ширину, высота определяется содержимым, всегда начинается с новой строки (`<div>`, `<form>`, `<h1>`, ..., `<h6>`, `<hr>`, `<p>`, `<table>`, ``).

Запрещено вставлять блочный элемент внутрь строчного. Например, `<h1> <a> Заголовок </h1>`

Типы элементов

display

Задаёт тип элемента. Значения:

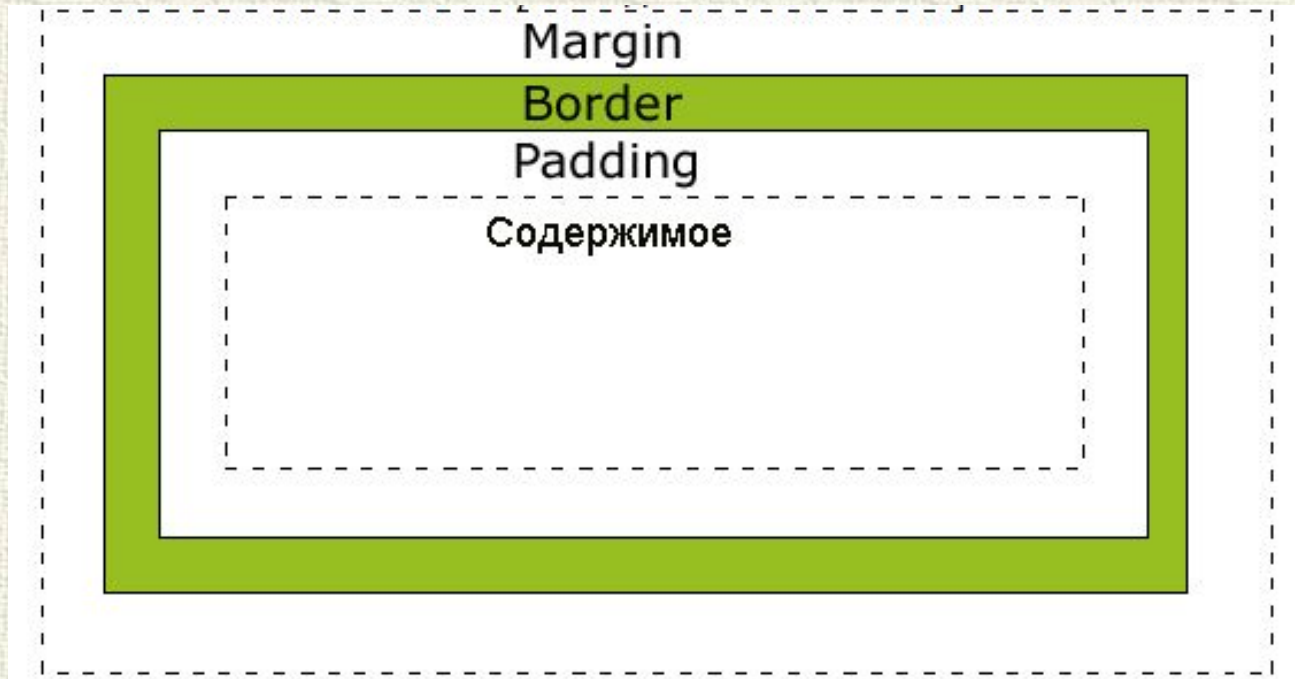
- 1) block – блочный элемент
- 2) in-line – строчный элемент
- 3) none – не отображается

Пример:

```
a {  
  display: block;  
}
```

Боксовая модель

Описывает контейнеры html-элементов



Свойства групп `margin` и `padding` задают отступы, а группа `border` – рамки элемента. Для задания ширины и высоты элементов используются свойства `width` (`min-width`, `max-width`) и `height` (`min-height`, `max-height`) соответственно.

Разделение контента

Для качественного восприятия контента его разделяют в разметке при помощи блоков. Оформление и позиционирование блоков происходит в CSS.

html

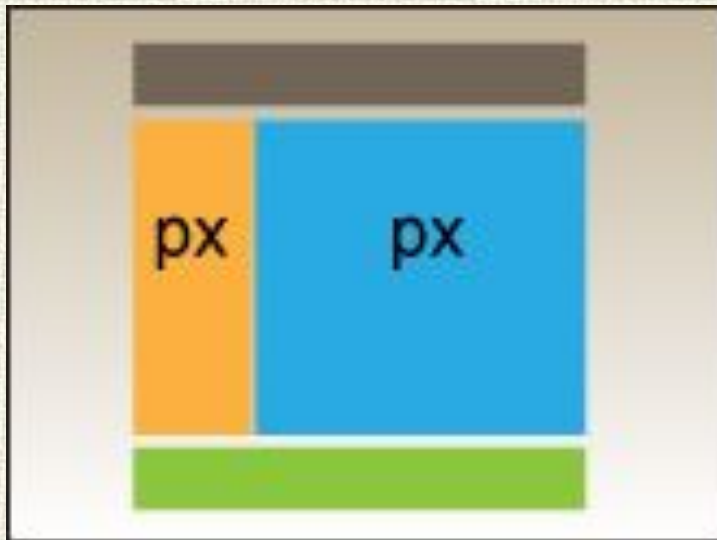
```
<body>  
  <div id="header">Шапка </header>  
  <div id="nav">Меню </nav>  
  <div id="article">  
    Основная часть </div>  
  <div id="footer">Подвал </footer>  
</body>
```

CSS

```
#header {  
  position: ...  
  left: ...  
  top: ...  
}  
  
#nav {...}  
#article {...}  
#footer {...}
```

Типы макетов

- 1) Фиксированный. Ширина макета задана в px.
 - + Упрощает дизайн и работу с контентом, корректнее отображается браузерами.
 - Плохо смотрится на мониторах с высоким и низким разрешением, неэффективно используя свободное место.

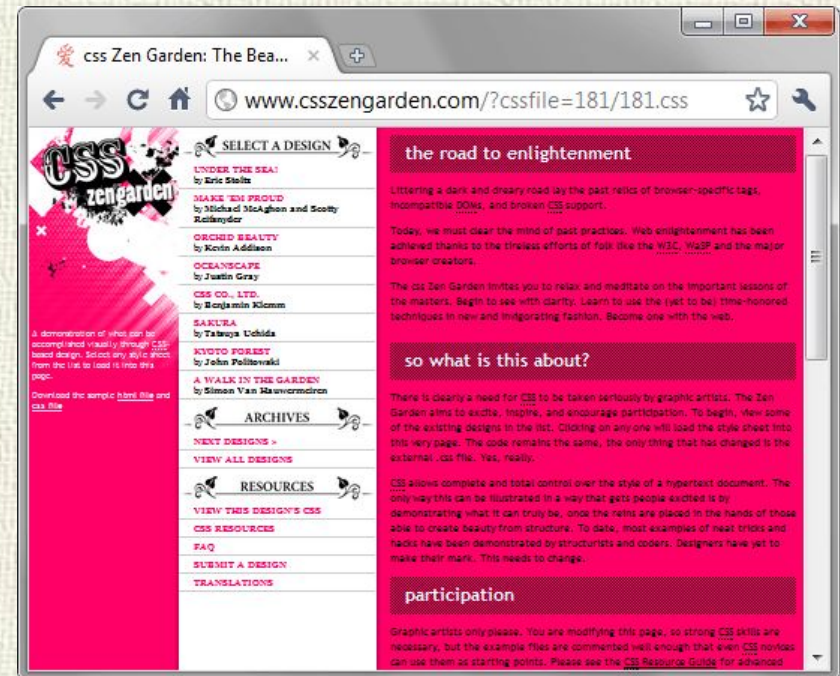


Типы макетов

2) Резиновый. Ширина колонок задаётся в процентах или сочетаются проценты и пиксели таким образом, что макет занимает всю свободную ширину окна браузера.

+ Эффективно использует пространство.

- На мониторах с высоким разрешением сайт плохо читается из-за чрезмерного удлинения строк текста. Макеты сложнее верстать и отлаживать.

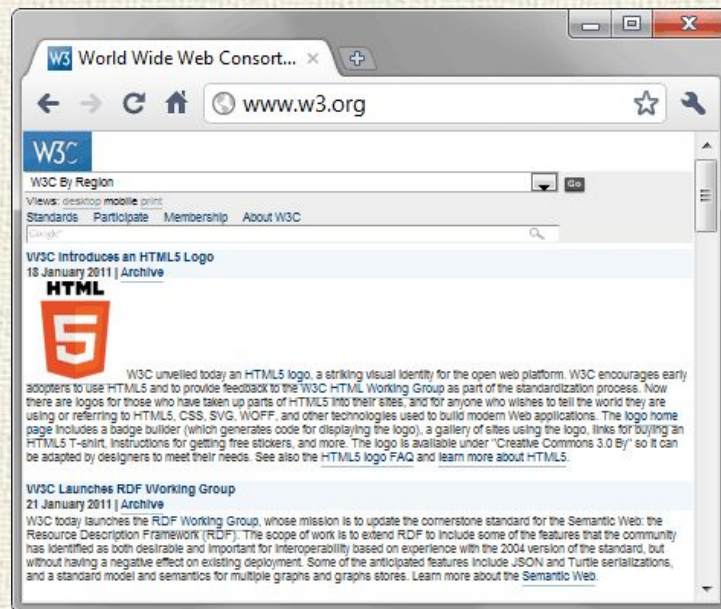


Типы макетов

3) Адаптивный. Подстраивается под разрешение монитора и окна браузера, меняя ширину макета, число колонок, размеры изображений и текста.

+ Наиболее удобен для пользователя, можно смотреть сайт на любом устройстве.

- Самый сложный макет.



Типы макетов

4) Комбинированный.

Предполагает использование разной ширины для отдельных частей страницы, например, шапку и подвал делают резиновыми, а контент фиксированным.

Наследует все плюсы и минусы фиксированного и резинового макета.

