

Введение в стили

Для чего нужен CSS?

CSS (англ. Cascading Style Sheets) - каскадные таблицы стилей. Используются для стилизации и компоновки веб-страниц - например, для изменения шрифта, цвета, размера и интервала содержимого, разделения его на несколько столбцов или добавления анимации.

Страница без CSS-стилей

Настройки браузера по умолчанию

Браузер будет стилизовать HTML-документ с помощью внутренних таблиц стилей. Благодаря этому заголовки будут больше обычного текста, ссылки будут выделяться и структуры, такие как списки и таблицы, будут понятны.

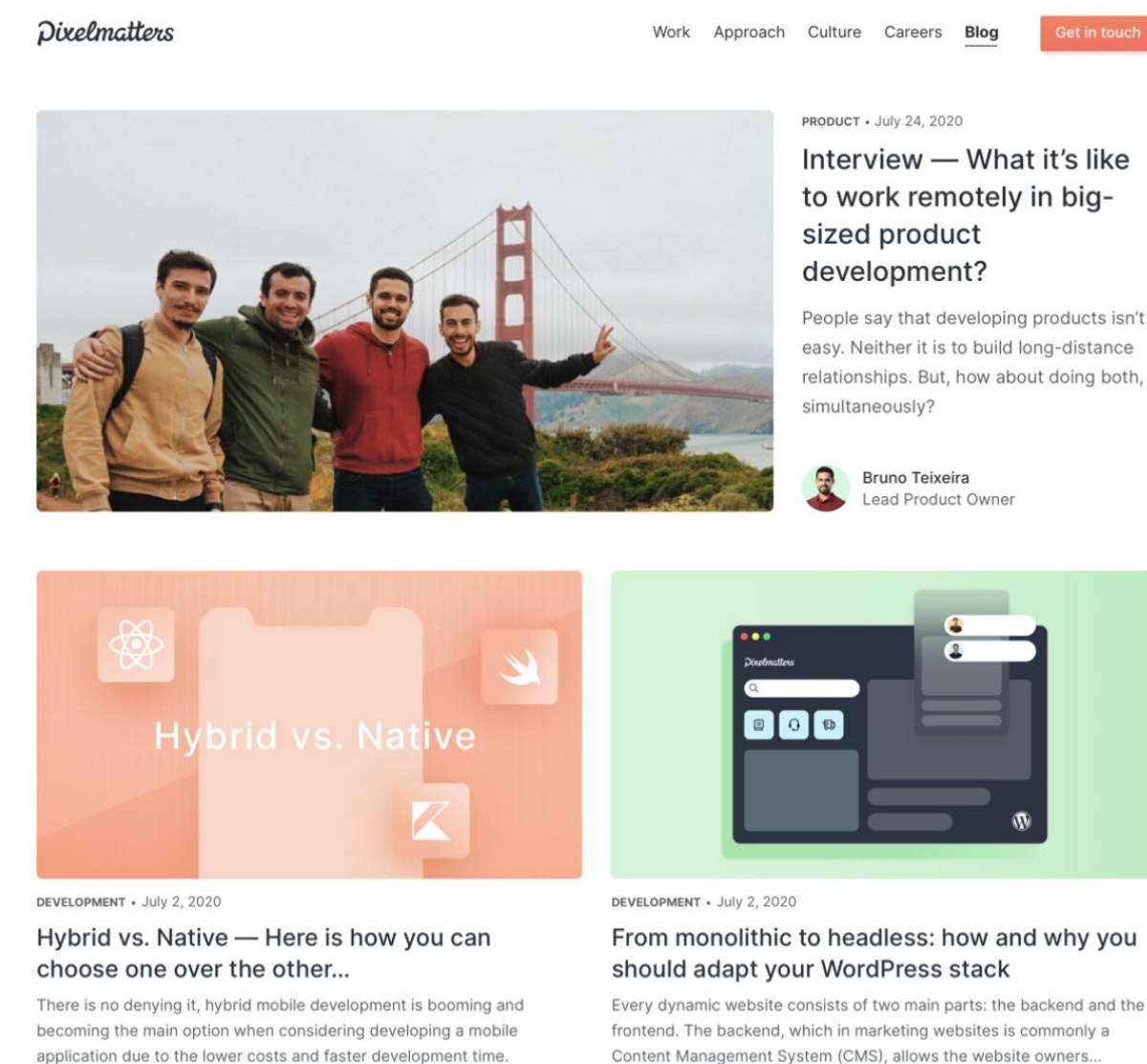
Абзацы отделяются друг от друга. У элементов списка есть маркер или номер, а ссылки выделены и подчеркнуты.

- Элемент один
- Элемент два

Подзаголовок

Всё это Вы можете изменить с помощью CSS.

Страница с применением CSS-стилей



Подключение стилей на страницу

- Встроенные стили (внутри HTML-кода страницы)

Стили прописываются внутри специального элемента `<style>`:

```
<head>  
  <style>  
    CSS-код  
  </style>  
</head>
```

- Внешние стили (из внешнего файла с расширением .css)

Используется элемент `<link>` с атрибутами `href` (адрес файла) и `rel="stylesheet"`

```
<head>  
  <link href="external.css" rel="stylesheet">  
</head>
```

Подключение стилей на страницу

- Встроенные стили, влияющие только на один элемент (содержатся в атрибуте style элемента)

```
<h1 style="color: blue">Hello</h1>
```

Не рекомендуется использовать

Как работает CSS?

1. Браузер получает HTML-страницу
2. Браузер преобразует HTML в DOM
3. Браузер забирает все ресурсы и описания, связанные с HTML-документом, например:
встроенные картинки, видео и стили CSS
4. Браузер анализирует полученный CSS код, сортирует описанные там правила в зависимости от их селекторов. Основываясь на найденных селекторах браузер понимает какие правила относятся к определённым «узлам» в DOM-дереве и применяет их по мере необходимости (формированием дерева представления)
5. Дерево представления формируется в том порядке, в каком оно затем должно будет отображаться, когда все правила будут применены.
6. Затем происходит визуальное отображение контента на странице (отрисовка)

Основные селекторы

Весь CSS-код состоит из повторяющихся блоков следующего вида:

```
selector {  
  property1: value;  
  property2: value;  
}
```

Селектор — это шаблон, который используется для выбора одного или нескольких элементов HTML и применения к ним параметров форматирования.

Основные селекторы

Виды селекторов:

- Селекторы по именам тегов (имя тега без < и >)

```
p {  
  color: blue;  
}
```

- Селекторы по классу (по значению атрибута class)

Класс элемента задается с помощью атрибута class, который содержит имя класса или имена классов через пробел:

```
<p class="info"></p>
```

Селектор с использованием класса задаётся как .classname

```
.info {  
}
```


Основные селекторы

- Селекторы по идентификатору (по значению атрибута id)

```
#footer {  
  height: 50%;  
}
```

- Универсальный селектор (выбор всех элементов на странице)

```
* {  
  margin: 0;  
  padding: 0;  
}
```


Основные селекторы

- Селекторы по атрибуту (выбор элементов по имени атрибута и (или) его значению)

Типы селекторов по атрибуту:

- [attr] - по имени атрибута
- [attr=value] - по имени и значению атрибута
- [attr^=value] - по имени и значению, с которого оно должно начинаться
- [attr|=value] - по имени атрибута и его значению, которое равно value или начинается с value
- [attr\$=value] - по имени атрибута и значению, на которое оно должно заканчиваться
- [attr*=value] - по указанному атрибуту и значению, которое должно содержать value
- [attr~=value] - по имени атрибута и значению, которое содержит value отделённое от других с помощью пробела

```
[class|= "test"] {  
    background-color: orange;  
}
```

Псевдоклассы

Псевдоклассы предназначены для более точного выбора элементов в зависимости от их динамического состояния или расположения.

Псевдокласс добавляется к селектору с помощью символа : , т.е. селектор:псевдокласс

- Псевдоклассы для выбора элементов в зависимости от их состояния
 - :link - предназначен для выбора не посещённых ссылок
 - :visited - предназначен для выбора посещённых ссылок
 - :hover - предназначен для выбора элементов при наведении на них курсора
 - :active - предназначен для выбора элементов в момент когда они активируются пользователем
 - :focus - предназначен для выбора элемента, который в данный момент находится в фокусе

Псевдоклассы

- Псевдоклассы по расположению среди соседей

:first-child - для выбора элементов, которые являются первыми дочерними элементами своего родителя

:last-child - для выбора элементов, которые являются последними дочерними элементами своего родителя

:only-child - для выбора элементов, если они являются единственными дочерними элементами внутри родительского контейнера

:nth-child(выражение) - для выбора элементов по их порядковому номеру в родителе (выражение - число, формулу или ключевые слова odd или even)

:nth-last-child(выражение) - то же самое, что и :nth-child, только отсчет элементов в родителе ведется не с начала, а с конца

Псевдоклассы

- Псевдоклассы по расположению среди соседей с учетом типа элемента

:first-of-type — выбирает первый элемент своего родителя с учётом его типа

:last-of-type - для выбора элементов, которые являются последними дочерними элементами

данного типа своего родителя

:nth-of-type(выражение) - для выбора элементов по их порядковому номеру в родителе с учетом

их типа

:nth-last-of-type(выражение) - аналогичен :nth-of-type с разницей в том, что отсчёт дочерних

элементов ведётся с конца

Псевдоклассы

- Остальные псевдоклассы:

- :not() - предназначен для выбора элементов, которые не содержат указанный селектор

- :empty - предназначен для выбора пустых элементов (элементов, у которых нет дочерних узлов, в том числе текстовых)

- :root - предназначен для выбора корневого элемента документа (в HTML корневым элементом является элемент <html>)

- :target - предназначен для выбора элемента, идентификатор которого соответствует хэшу в URL-адресе

Псевдоэлементы

Псевдоэлементы – это ключевые слова, которые определяют область элементов, которая изначально отсутствует в дереве документа. Эта область создается искусственно с помощью CSS

Псевдоэлемент добавляется к селектору с помощью `::` , т.е. селектор::псевдоэлемент

- Псевдоэлементы

`::after` - используется для вывода желаемого контента после содержимого элемента, к которому он добавляется

`::before` - используется для отображения контента до содержимого элемента, к которому он добавляется

Псевдоэлементы

- Псевдоэлементы

`::first-letter` - определяет стиль первого символа в тексте элемента, к которому добавляется

`::first-line` -задаёт стиль первой строки форматированного текста

`::selection` - применяет стиль к выделенному пользователем фрагменту текста

`::placeholder` - задает стилевое оформление подсказывающего текста, созданного атрибутом

`placeholder`

Наследование стилей

Концепция наследования в CSS:

Некоторые значения свойства CSS, установленные для родительских элементов наследуются их дочерними элементами, а некоторые нет.

Наследуемые свойства:

В основном свойства, определяющие параметры отображения текста: font-size, font-family, font-style, font-weight, color, text-align, text-transform; visibility, cursor и др.

Ненаследуемые свойства:

В основном параметры позиционирования, размеров, отступов, фона, рамок: background, border, padding, margin, width, height, position и др.

Наследование стилей

Значения свойств, контролирующие наследование:

- `inherit` - устанавливает значение свойства, применённого к элементу, таким же, как у его родительского элемента
- `initial` - устанавливает значение свойства, применённого к выбранному элементу, равным `initial value` этого свойства
- `unset` - возвращает свойству его естественное значение, что означает, что если свойство наследуется естественным образом, оно действует как `inherit`, иначе оно действует как `initial`
- `revert` - свойство принимает такое значение, которое было бы, если бы не было никаких стилей в текущем источнике стилей (ограниченная поддержка браузерами)

Каскадирование и специфичность

CSS (Cascading Style Sheets) - Каскадные Таблицы Стилей

Порядок следования правил в CSS имеет значение.

Приоритеты браузеров (чем выше в списке - тем ниже приоритет):

1. Стиль браузера
2. Стиль автора
3. Стиль пользователя
4. Стиль автора с добавлением !important
5. Стиль пользователя с добавлением !important

Ключевое слово !important - повышение приоритета (важности) стиля

Каскадирование и специфичность

Специфичность определяет, как браузер решает, какое именно правило применяется в случае, когда несколько правил имеют разные селекторы, но, тем не менее, могут быть применены к одному и тому же элементу

Если два селектора имеют одинаковую специфичность, то применяться будет тот стиль, что указан в коде ниже.

Каскадирование и специфичность

Расчет специфичности браузером:

Тысячи: объявление стиля находится внутри атрибута style (встроенные стили). Такие объявления не имеют селекторов, поэтому их специфичность всегда просто 1000.

Сотни: каждый селектор ID, содержащийся в общем селекторе

Десятки: каждый селектор класса, селектор атрибута или псевдокласс, содержащийся в общем селекторе

Единицы: каждый селектор элемента или псевдоэлемент, содержащийся в общем селекторе

Каскадирование и специфичность

Пример расчета специфичности:

Селектор	Тысячи	Сотни	Десятки	Единицы	Специфичность
h1	0	0	0	1	1
#id	0	1	0	0	100
h1 + p::first-letter	0	0	0	3	3
Без селектора, с правилом внутри атрибута style элемента	1	0	0	0	1000

Каскадирование и специфичность

Расчет специфичности:

- Самый высокий приоритет имеет атрибут style
- Второе место - присутствие ID в селекторе
- Третье место - атрибуты(в том числе и атрибут class) и псевдоклассы в селекторе
- Последнее место - селекторы с именами элементов и псевдоэлементами

!important делает конкретное свойство и значение самыми специфичными, таким образом

переопределяя нормальные правила каскада (рекомендуется использовать только в крайнем случае)