

CSS

<<Media Queries>>

Введение

В 2001 году в **HTML4** и **CSS2** была введена поддержка аппаратно-зависимых таблиц стилей, позволившая создавать стили и таблицы стилей для определенных типов устройств. В качестве медиа-типов были определены следующие: **aural**, **braille**, **handheld**, **print**, **projection**, **screen**, **tty**, **tv**. Таким образом, браузер применял таблицу стилей только в случае, когда активизировался данный тип устройства.

Кроме того, было введено ключевое слово **all**, которое использовалось, чтобы указать, что таблица стилей применяется ко всем типам носителей.

В HTML4 медиа-запрос записывался следующим образом:

```
<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css">  
<link rel="stylesheet" type="text/css" media="print" href="serif.css">
```

Внутри таблицы стилей также можно было объявить, что блоки объявлений должны применяться к определенным типам носителей:

```
@media screen {  
    * {font-family: sans-serif;}  
}
```

Предусматривая возможность введения новых значений и значений с параметрами в будущем, для браузеров была реализована поддержка значений атрибута медиа-носителя, указанных следующим образом:

```
<link rel="stylesheet" media="screen, 3d-glasses, print and resolution > 90dpi" href="...">
```

Текущий синтаксис **HTML5** и **CSS3** напрямую ссылается на первую спецификацию **Media Queries**, обновляя правила для **HTML**. Также был расширен список характеристик медиа-носителей.

Добавление медиа-запросов

В общем случае **медиа-запрос** состоит из **ключевого слова**, описывающего тип устройства (необязательный параметр) и **выражения**, проверяющего характеристики данного устройства.

Из всех характеристик чаще всего проверяется ширина устройства **width**.

Медиа-запрос является **логическим выражением**, которое возвращает **true** или **false**.

```
<link rel="stylesheet" media="screen and (color)" href="example.css">
```

```
@import url(thema.css) screen and (color);
```

```
<style>
  @media (max-width: 600px) {
    #sidebar {display: none;}
  }
</style>
```

```
@media (max-width: 600px) {
  #sidebar {display: none;}
}
```

Таблица стилей, прикрепленная через тег **<link>**, будет загружаться вместе с документом, даже если её медиа-запрос вернет **false**

Логические операторы

Оператор **and** связывает друг с другом разные условия.

```
@media screen and (max-width: 600px) {  
    /* CSS-styles */;  
}
```

Стили этого запроса будут применяться только для экранных устройств с шириной области просмотра не более **600px**

```
@media (min-width: 600px) and (max-width: 800px) {  
    /* CSS-styles */;  
}
```

Стили этого запроса будут применяться для всех устройств при ширине области просмотра от **600px** до **800px** включительно

Другие операторы

Оператор **запятая** работает по аналогии с логическим оператором **or**.

```
@media screen, projection{  
    /* CSS-styles ;*/  
}
```

В данном случае CSS-стили, заключенные в фигурные скобки, сработают только для экранных или проекционных устройств.

Оператор **not** позволяет сработать медиа запросу в противоположном случае. Ключевое слово **not** добавляется в начало медиа запроса и применяется ко всему запросу целиком

```
@media not all and (monochrome) {  
    /* CSS-styles ;*/  
}
```

Оператор **only** используется, чтобы скрыть стили от старых браузеров (поддерживающих синтаксис медиа-запросов CSS2).

```
<link rel="stylesheet"  
      media="only screen and (min-width: 401px) and (max-width: 600px)"  
      href="example.css">
```

Тип носителя

Тип носителя представляет собой тип устройства, например, принтеры, экраны

all	Подходит для всех типов устройств.
print	Предназначен для страничных материалов и документов, просматриваемых на экране в режиме предварительного просмотра печати.
screen	Предназначен в первую очередь для экранов цветных компьютерных мониторов.
speech	Предназначен для синтезаторов речи.

Характеристики носителя

К характеристикам медианосителя относятся проверяемые параметры устройства. Значения, которые используются при задании характеристик, являются контрольными точками.

width	<p>Проверяет ширину области просмотра. Значения задаются в единицах длины, px, em и т.д., например, (width: 800px). Обычно для проверки используются минимальные и максимальные значения ширины.</p> <p>min-width применяет правило если ширина области просмотра больше значения, указанного в запросе, max-width — ширина области просмотра меньше значения, указанного в запросе.</p>
height	<p>Проверяет высоту области просмотра. Значения задаются в единицах длины, px, em и т.д., например, (height: 500px). Обычно для проверки используются минимальные и максимальные значения высоты.</p> <p>min-height применяет правило если высота области просмотра больше значения, указанного в запросе, max-height — высота области просмотра которого меньше значения, указанного в запросе.</p>
aspect-ratio	<p>Проверяет соотношение ширины к высоте области просмотра. Широкоэкранный дисплей с соотношением сторон 16:9 может быть помечен как (aspect-ratio: 16/9).</p> <p>min-aspect-ratio проверяет минимальное соотношение, max-aspect-ratio — максимальное соотношение ширины к высоте области просмотра.</p>
orientation	<p>Проверяет ориентацию области просмотра. Принимает два значения: (orientation: portrait) и (orientation: landscape).</p>

resolution	<p>Проверяет разрешение экрана (количество пикселей). Значения также могут проверять количество точек на дюйм (dpi) или количество точек на сантиметр (dpcm), например, (resolution: 300dpi).</p> <p>min-resolution проверяет минимальное разрешение экрана, max-resolution — максимальное.</p>
color	<p>Проверяет количество бит на каждый из цветовых компонентов устройства вывода. Например, (min-color: 4) означает, что экран конкретного устройства должен иметь 4-битную глубину цвета.</p> <p>min-color проверяет минимальное количество бит, max-color — максимальное количество бит.</p>
color-index	<p>Проверяет количество записей в таблице подстановки цветов. В качестве значения указывается положительное число, например, (color-index: 256).</p> <p>min-color-index проверяет минимальное количество записей, max-color-index — максимальное количество записей.</p>
monochrome	<p>Проверяет количество битов на пиксель монохромного устройства. Значение задается целым положительным числом, например, (min-monochrome: 8).</p> <p>min-monochrome проверяет минимальное количество битов, max-monochrome — максимальное количество битов.</p>
-webkit-device-pixel-ratio	<p>Задаёт количество физических пикселей устройства на каждый CSS-пиксель.</p>

Метатег viewport

Для управления разметкой в мобильных браузерах используется метатег **viewport**. Изначально данный тег был представлен разработчиками Apple для браузера Safari на iOS. Мобильные браузеры отображают страницы в виртуальном окне просмотра, которое обычно шире, чем экран устройства. С помощью мета тега **viewport** можно контролировать размер окна просмотра и масштаб.

Страницы, адаптированные для просмотра на разных типах устройств, должны содержать в разделе **<head>** метатег **viewport**.

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

Свойство **width** определяет виртуальную ширину окна просмотра, значение **device-width** — физическую ширину устройства. Другими словами, **width** отражает значение **document.documentElement.clientWidth**, а **device-width** — **screen.width**.

При первой загрузке страницы свойство **initial-scale** управляет начальным уровнем масштабирования, **initial-scale=1** означает, что 1 пиксель окна просмотра равен 1 пикселю CSS.