



## Урок 7

# Адаптивная верстка, часть 1

Медиазапросы в CSS3. Тип и характеристики носителя.  
Логические операторы

[Адаптивный дизайн](#)

[Медиазапросы в CSS3](#)

[Структура медиазапросов](#)

[Тип носителя](#)

[Характеристики носителя](#)

[Логические операторы](#)

[Оператор and](#)

[Оператор not](#)

[Оператор only](#)

[Оператор «запятая»](#)

[Основные экраны](#)

[Пример адаптивной верстки](#)

[Практическое задание](#)

[Используемая литература](#)

# Адаптивный дизайн

Адаптивная и резиновая верстка – не одно и то же. Резиновая верстка изменяет размеры блоков в зависимости от ширины экрана, адаптивная верстка не сжимается или растягивается, а именно адаптируется под ширину.

Для адаптивной верстки используются медиазапросы в CSS.

Медиазапросы позволяют менять верстку при изменении ширины экрана. Например, можно сделать так, чтобы если ширина окна больше 800px, то отступы на странице в body были 10px, а при ширине 800px и меньше – 3px. Это и есть пример адаптивной верстки.

Адаптивная верстка – это отдельные стили для разных случаев (различных разрешений экрана).

## Медиазапросы в CSS3

Медиазапросы проверяют такие свойства, как ширина и высота окна браузера, ширина и высота экрана устройства, ориентация планшета или телефона (альбомная или портретная) и много другое, но чаще всего проверяются ширина области просмотра и ширина экрана устройства.

## Структура медиазапросов

Медиазапрос состоит из правила @media, за которым следует указание типа носителя и по крайней мере одна проверяемая характеристика носителя – условие медиазапроса:

```
@media screen and (max-width: 600px)
{
  body {font-size: 80%;}
}
```

Стили данного медиазапроса, заключенные в фигурные скобки, будут применяться в тех случаях, когда ширина области просмотра не превышает 600px. Если ширина будет больше, то браузер проигнорирует стили этого медиазапроса.

Правила @media прописываются в конце файла .css.

## Тип носителя

Тип носителя = тип устройства:

Значение	Описание
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.

Параметр	Описание
width	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. min-width применяет правило, если ширина области просмотра больше значения, указанного в запросе, max-width – если ширина области просмотра меньше значения, указанного в запросе
height	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. min-height применяет правило, если высота области просмотра больше значения, указанного в запросе, max-height – если высота области просмотра меньше значения, указанного в запросе
aspect-ratio	Проверяет соотношение ширины к высоте области просмотра. Широкоэкранный дисплей с соотношением сторон 16:9 может быть помечен как (aspect-ratio: 16/9). min-aspect-ratio проверяет минимальное соотношение, max-aspect-ratio – максимальное соотношение ширины области просмотра к высоте
orientation	Проверяет ориентацию области просмотра. Принимает два значения: (orientation: portrait) и (orientation: landscape)
resolution	Проверяет разрешение экрана (количество пикселей). Значения также могут проверять количество точек на дюйм (dpi) или количество точек на сантиметр (dpcm), например (resolution: 300dpi). min-resolution проверяет минимальное разрешение экрана, max-resolution – максимальное
color	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. min-color проверяет минимальное количество бит, max-color – максимальное количество бит
color-index	Проверяет количество записей в таблице подстановки цветов. В качестве значения указывается положительное число, например (color-index: 256). min-color-index проверяет минимальное количество записей, max-color-index – максимальное количество записей
monochrome	Проверяет количество бит на пиксель монохромного устройства. Значение задается целым положительным числом, например (min-monochrome: 8). min-monochrome проверяет минимальное количество бит, max-monochrome – максимальное количество бит
-webkit-device-pixel-ratio	Задаёт количество физических пикселей устройства на каждый CSS-пиксель

Физические пиксели (device pixel или physical pixel) – привычные нам пиксели: самые маленькие элементы любого дисплея, каждый из которых имеет свой цвет и яркость.

Плотность экрана (Screen density) – это количество физических пикселей дисплея. Обычно оно измеряется в пикселях на дюйм (PPI: pixels per inch). Apple, разработав Retina-экраны с двойной плотностью пикселей, утверждает, что человеческий глаз не способен различить бо́льшую плотность.

CSS-пиксели (CSS pixels) – абстрактная величина, используемая браузерами для точного отображения контента на страницах вне зависимости от экрана (DIPs: device-independent pixels). Пример:

```
<div height="200" width="300"></div>
```

Такой блок на обычных экранах будет занимать область 200×300 пикселей, а на Retina-экранах тот же блок будет иметь размеры 400×600 пикселей. Таким образом, на Retina-экранах плотность пикселей в 4 раза больше, чем на обычных.

## Логические операторы

С помощью логических операторов можно создавать комбинированные медиазапросы, в которых будет проверяться соответствие нескольким условиям.

### Оператор and

Оператор and связывает друг с другом разные условия:

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

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-СТИЛИ */  
}
```

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

Правило @media all and (max-width: 600px) {...} равнозначно правилу @media (max-width: 600px) {...}.

### Оператор not

Оператор not позволяет медиазапросу сработать в случае, если условия противоположны указанным. Ключевое слово not добавляется в начало медиазапроса и применяется ко всему запросу целиком:

```
@media not all and (monochrome) {...}
```

Если медиазапрос составлен с использованием оператора «запятая», то отрицание будет распространяться только на ту часть, которая идет до запятой:

```
@media not screen and (color), print and (color)
```

## Оператор only

Оператор only позволяет подключать стили для браузеров, которые не поддерживают медиазапросы:

```
@media only screen and (color) {  
  /* CSS-СТИЛИ */;  
}
```

## Оператор «запятая»

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

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

## Основные экраны

Чтобы адаптировать дизайн сайта под разные устройства, необходимо задать разные стили для разных разрешений экранов, используя контрольные точки:

```
@media only screen and (min-width: 1224px) {  
  /* СТИЛИ */  
}  
@media only screen and (min-width: 1824px) {  
  /* СТИЛИ */  
}  
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){  
  /* СТИЛИ */  
}  
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){  
  /* СТИЛИ */  
}  
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 3){  
  /* СТИЛИ */  
}  
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 3){  
  /* СТИЛИ */  
}  
@media only screen and (min-width: 360px) and (max-height: 640px) and
```

```
(orientation: landscape) and (-webkit-device-pixel-ratio: 3){  
/* стили */  
}  
@media only screen and (min-width: 360px) and (max-height: 640px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 3){  
/* стили */  
}
```

## Пример адаптивной верстки

Так как урок является практическим, копировать в методические материалы весь код, который разбирается на уроке, нет смысла. К уроку должны быть прикреплены материалы, сверстанные преподавателем или подготовленные программой.

## Практическое задание

1. Создать адаптивный дизайн нашего макета.
2. Добавить медиазапросы для разных разрешений (рекомендуемые планшетные и мобильные устройства, можно добавить и множество других, но на начальном этапе это не рекомендуется).
3. Применить на практике полученные знания.

## Используемая литература

1. <http://www.wisdomweb.ru/>.
2. Гоше Х. HTML5. Для профессионалов. СПб.: Питер, 2013. – 496 с.
3. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. СПб.: Питер, 2012.
4. Макфарланд Д. Большая книга CSS3. СПб.: Питер, 2014.