

Профессиональная вёрстка

Урок 6

CSS3 медиазапросы.

Адаптивная верстка

Понятие и применение адаптивного дизайна. Основы синтаксиса и примеры.

[Адаптивный дизайн](#)

[CSS3 медиазапросы](#)

[Структура медиазапросов](#)

[Тип носителя](#)

[Характеристики носителя](#)

[Логические операторы](#)

[Оператор and](#)

[Оператор not](#)

[Оператор only](#)

[Оператор запятая](#)

[Основные экраны](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

Адаптивный дизайн

Адаптивная верстка – не то же, что «резиновая». «Резиновая» верстка изменяет размеры блоков в зависимости от ширины экрана, адаптивная же не сжимается и не 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, например, 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).

С чего начинается создание адаптивного сайта?

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 метатега 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 = 1 пиксель CSS.

CSS3 медиазапросы

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 истину или ложь.

Медиазапросы проверяют такие свойства, как ширина и высота окна браузера, ширина и высота экрана устройства, ориентация планшета или телефона (альбомная или портретная) и многое другое, но чаще всего проверяется ширина области просмотра и 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, браузер игнорирует стили данного медиазапроса.

Правила @media прописываются в конце файла CSS-стилей.

Тип носителя

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

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 ширины к высоте области просмотра, max-aspect-ratio – максимальное

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 битов на пиксель монохромного устройства. Значение задается целым положительным числом, например (min-monochrome: 8). min-monochrome проверяет минимальное количество битов, max-monochrome – максимальное количество битов
-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) – количество физических пикселей дисплея. Обычно измеряется в пикселях-на-дюйм (PPI: pixels per inch). Apple, разработав Retina-экраны с двойной плотностью пикселей, утверждает, что человеческий глаз неспособен 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 200x300 пикселей, а на Retina-экранах тот же блок получит 400x600 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 противоположном случае. 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 */  
}
```

Практическое задание

1. Доделать то, что не получилось на предыдущих уроках.
2. Проверить сайт на кроссбраузерность.
3. Проверить сайт на валидность.

Дополнительные материалы

1. @media – <https://developer.mozilla.org/ru/docs/Web/CSS/@media>.
2. 14 инструментов для проверки кроссбраузерности сайта – <http://ruseller.com/lessons.php?rub=28&id=1209>.
3. CSS3 Generator – <http://css3generator.com/CSS3>.
4. Animation Generator – <http://webcodetools.com/css-generator/animation>.

5. Кривые Безье – <https://matthewlein.com/ceaser/>.

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://www.wisdomweb.ru/>.
2. <http://html5book.ru/css3-transform/>.
3. <https://html5book.ru/css3-mediazaprosy/>
4. Гоше Х. HTML5. Для профессионалов. – СПб.: Питер, 2013. – 496 с.: ил.
5. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. – СПб.: Питер, 2012. – 272 с.
6. Макфарланд Д. Большая книга CSS3. – СПб.: Питер, 2016. – 608 с.