

CSS3 Адаптивный веб дизайн

Проектирование

Адаптивный веб дизайн - делает вашу веб страницу, чтобы она хорошо выглядела на всех устройствах.

Веб страницы можно просматривать с различных устройств: настольных компьютеров, планшетов и телефонов.

Ваша веб страница должна хорошо выглядеть и быть простой в использовании, независимо от устройства.

Веб страницы не должны пропускать информацию для небольших устройств, и адаптировать ее содержание под любое устройство:



Область просмотра

Область просмотра - это видимая область веб страницы пользователя.

Область просмотра зависит от устройства и будет меньше на мобильном телефоне, чем на экране компьютера.

HTML5 ввел метод, позволяющий веб дизайнерам управлять видовым экраном через тег `<meta>`.

Необходимо включить следующий элемент `<meta>`, установить во все ваших веб страницах:

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

Тег `<meta>` содержит инструкцию `viewport` браузера для управления размерами и масштабированием страницы.

Часть `width=device-width` задает ширину страницы следить за шириной экран - устройства (которая будет меняться в зависимости от устройства).

Часть `initial-scale=1.0` задает начальный уровень масштабирования при первой загрузке страницы браузером.

Размер содержимого окна просмотра

Пользователи используют вертикальную прокрутку веб сайтов на настольных и мобильных устройствах, а не горизонтальную!

Таким образом, если пользователь вынужден прокручивать по горизонтали или уменьшить масштаб, чтобы увидеть всю веб страницу приводит к плохому пользовательскому опыту.

Некоторые дополнительные правила:

- 1. Не используйте большие элементы фиксированной ширины** - , например, если изображение отображается по ширине шире, чем область просмотра, видимый экран может вызвать горизонтальную прокрутку. Не забудьте подогнать содержимое под ширину окна просмотра.
- 2. Не позволяйте содержимому полагаться на определенную ширину окна просмотра, чтобы визуализация была хорошей** - размеры и ширина экрана CSS в пикселях различаются шириной между устройствами, содержание не должно полагаться на определенную ширину видового экрана для хорошей визуализации.
- 3. Используйте медиа запросы CSS для применения различных стилей больших экранов** - установка большой абсолютной ширины CSS для элементов страницы вызывающий элемент должен быть слишком широким для просмотра на мобильном устройстве. Вместо этого рекомендуется использовать относительные величины, например `width: 100%`. Также быть осторожным в использование больших абсолютных расположений значений. Он может причинить элементу к выводу за пределы области просмотра на небольших устройствах.

Сетка

Многие веб страницы основаны на "вид сетки", означает, что страница разделена на столбцы:



Отзывчивый вид сетки имеет 12 столбцов, общая ширина 100 пикселей, и она будет сокращаться и расширяться по мере изменения размера окна браузера.

Сетка

Сначала убедитесь, что все HTML-элементы имеют свойство `box-sizing` установлено значение `border-box`. Это гарантирует, что отступы и границы включены в общую ширину и высоту элементов.

Мы хотим использовать отзывчивый вид сетки с 12 столбцами, чтобы иметь больше контроля над веб страницей.

Сначала мы должны вычислить процент для одного столбца: $100 \text{ процентов} / 12 \text{ столбцов} = 8.33 \text{ процентов}$.

Тогда мы сделаем отдельный класс для каждого из 12 столбцов, `class="col-"` и номер определит количество столбцов в разделе:

```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```

Сетка

Колонки должны быть обернуты в `<div class="row">` который является флекс-контейнером, или он должен иметь псевдоэлементы `:before` и `:after` при построении сетки на плавающих блоках:

```
.row:after, .row:before{
  content: "";
  clear: both;
  display: table;
}
```

Количество колонок внутри строки всегда должна быть до 12:

```
<div class="row">
  <div class="col-3">...</div> <!-- 25% -->
  <div class="col-9">...</div> <!-- 75% -->
</div>
```

Все эти столбцы иметь отступ по бокам к примеру 15 пикселей, и должны плавать влево если используем второй вариант сетки.

Медиа запросы

Медиа запрос - это CSS исполнение, представленный в CSS3.

Используется правило `@media` для включения блока свойств CSS, только если определено условие true.

Если окно браузера 600 пикселей или меньше, цвет фона будет светло-голубым:

```
@media only screen and (max-width: 600px) {  
    body {  
        background-color: lightblue;  
    }  
}
```


Медиа запросы

При помощи сеток вы сделали веб страницу со строками и столбцами, отзывчивым, но на маленьком экране он выглядел плохо.

В этом могут помочь медиа запросы. Мы можем добавить точку прерывания, где определенные части дизайна будут вести себя по-разному на каждой сетке точки прерывания.

Использовать медиа-запрос, чтобы добавить точку прерывания на 768 пикселей:

```
@media only screen and (max-width: 768px) {  
    /* Для мобильных телефонов: */  
    [class*="col-"] {  
        width: 100%;  
    }  
}
```

Медиа запросы

Всегда первый мобильный дизайн.

Мобильный в первую очередь означает дизайн для мобильных устройств до настольного компьютера или любых других устройств (это ускорит отображение страницы на небольших устройствах).

Это означает, что мы должны внести некоторые изменения в CSS.

Вместо изменения стилей, когда ширина становится *меньше*, чем 992 пикселей, мы должны изменить конструкцию когда ширина получает *больше* чем 992 пикселей. Это сделает наш мобильный дизайн первым.

Можно добавить любое количество точек прерывания.

Медиа запросы

Мы также вставим точку прерывания между планшетами и мобильными телефонами.

```
@media only screen and (min-width: 768px) {  
    /* Для планшетов: */  
    .col-s-1 {width: 8.33%;}  
    .col-s-2 {width: 16.66%;}  
    .col-s-3 {width: 25%;}  
    .col-s-4 {width: 33.33%;}  
    .col-s-5 {width: 41.66%;}  
    .col-s-6 {width: 50%;}  
    .col-s-7 {width: 58.33%;}  
    .col-s-8 {width: 66.66%;}  
    .col-s-9 {width: 75%;}  
    .col-s-10 {width: 83.33%;}  
    .col-s-11 {width: 91.66%;}  
    .col-s-12 {width: 100%;}  
}
```

Медиа запросы

Может показаться странным, что у нас есть два набора одинаковых классов, но это дает нам перспективу в *HTML*, чтобы решить, что будет происходить со столбцами в каждой точка прерывания:

Для компьютеров:

Первый и третий раздел будет охватывать каждые 3 столбца. Средняя часть будет охватывать 6 столбцов.

Для планшетов:

Первый раздел будет охватывать 3 столбца, второй - 9, и третий раздел будет отображаться ниже первых двух разделов, и он будет охватывать 12 столбцов:

```
<div class="row">
  <div class="col-3 col-s-3">...</div>
  <div class="col-6 col-s-9">...</div>
  <div class="col-3 col-s-12">...</div>
</div>
```

Типичные устройства точка прерывания

Есть много экранов и устройств с разной высотой и шириной, поэтому трудно создать точную точку прерывания для каждого устройства. Для простоты можно ориентироваться на пять групп:

```
/* Очень маленькие устройства (телефоны, 600px и ниже) */  
@media only screen and (max-width: 600px) {...}
```

```
/* Маленькие устройства (портретные планшеты и большие телефоны, 600px и выше) */  
@media only screen and (min-width: 600px) {...}
```

```
/* Средние устройства (альбомные планшеты, 768px и выше) */  
@media only screen and (min-width: 768px) {...}
```

```
/* Большие устройства (ноутбуки / настольные компьютеры, 992px и выше) */  
@media only screen and (min-width: 992px) {...}
```

```
/* Очень большие устройства (большие ноутбуки и настольные компьютеры, 1200px и выше) */  
@media only screen and (min-width: 1200px) {...}
```

Медиа запросы

Медиа запросы также могут быть использованы для изменения формата страницы в зависимости от ориентации браузера.

Вы можете иметь набор свойств CSS, которые будут применяться, когда окно браузера шире, чем его высота, так называемый "Альбомная" ориентация:

Веб страница будет иметь светло-голубой фон, если ориентация в альбомном режиме:

```
@media only screen and (orientation: landscape){  
  body {  
    background-color: lightblue;  
  }  
}
```

Медиа запросы

Другим распространенным использованием медиа запросов является скрывание элементов на экранах разного размера:

```
@media only screen and (max-width: 600px) {  
  .example {  
    display: none;  
  }  
}
```

Вы также можете использовать медиа запросы для изменения размера шрифта элемента на различных размерах экранов:

```
@media only screen and (min-width: 601px) {  
  div.example {  
    font-size: 80px;  
  }  
}  
  
/* Если размер экрана 600px или менее, установите размер шрифта <div> в 30px */  
@media only screen and (max-width: 600px) {  
  div.example {  
    font-size: 30px;  
  }  
}
```

Изображения

если свойство `max-width` имеет значение 100 процентов, изображение будет уменьшено, если необходимо, но никогда не масштабируйте больше чем свой первоначальный размер:

```
img {  
    max-width: 100%;  
    height: auto;  
}
```

Для фоновых изображений стоит использовать `background-size` со значением "cover", фоновое изображение будет масштабироваться чтобы охватить всю область содержимого. Заметьте что значение "cover" сохраняет аспект соотношение, и некоторая часть фонового изображения может быть обрезана.

Изображения

Большое изображение может быть идеальным и на большом компьютере экрана, но бесполезный на маленьком устройстве. Зачем загружать большие изображения, если тебе все равно нужно уменьшить масштаб? Для уменьшения нагрузки или по другим причинам можно использовать медиа запросы для отображения различных изображений на разных устройствах.

Здесь есть одно большое изображение и изображение, которое будет отображаться на различных устройствах:

```
/* Для ширины меньше 400px: */

body {
    background-image: url('img_smallflower.jpg');
}

/* Для ширины 400px и больше: */
@media only screen and (min-width: 400px) {
    body {
        background-image: url('img_flowers.jpg');
    }
}
```

Изображения

HTML5 предлагает элемент `<picture>` элемент, который позволяет определить более одного изображение.

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 400px)" ">
  <source srcset="img_flowers.jpg">
  
</picture>
```

Атрибут `srcset` является обязательным и определяет источник изображения.

Атрибут `media` является необязательным и принимает медиа запросы.

Видео

Чтобы сделать адаптивный блок с видео-роликом, который корректно будет отображаться на всех устройствах, воспользуемся следующим приёмом:

```
.thumb-wrap {  
  position: relative;  
  padding-bottom: 56.25%; /* задаёт высоту контейнера для 16:9 (если 4:3 — поставьте 75%) */  
  height: 0;  
  overflow: hidden;  
}
```

```
.thumb-wrap iframe {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  border-width: 0;  
  outline-width: 0;  
}
```