

CSS - адаптивность

мобильная верстка

Что такое и почему важно

Мобильная верстка и адаптивность на front-end'е относятся к способам создания веб-страниц, которые оптимизированы для просмотра на мобильных устройствах, таких как смартфоны и планшеты, а также адаптируются к разным размерам экранов.

Мобильная верстка:

- Это процесс создания веб-страниц, который учитывает ограничения и особенности мобильных устройств, такие как меньшие экраны, сенсорные интерфейсы и ограниченные вычислительные ресурсы.
- В мобильной верстке используются специальные техники и подходы, чтобы обеспечить удобство и эффективность использования веб-страниц на мобильных устройствах.

Адаптивность на front-end'е:

- Это концепция, согласно которой веб-страницы должны адаптироваться к разным размерам и ориентациям экранов, чтобы обеспечить оптимальное отображение и пользовательский опыт на любом устройстве.
- Для реализации адаптивности на front-end'е используются технологии, такие как медиазапросы в CSS, которые позволяют изменять стили в зависимости от характеристик экрана, таких как ширина, высота и разрешение.

@media - медиа запросы

Как добавить:

С помощью HTML:

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

С помощью правила @import:

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

Непосредственно в коде:

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

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.

Оператор запятая

Оператор запятая работает по аналогии с логическим оператором or.

```
@media screen, projection {
```

```
/* CSS-стили */;
```

```
}
```

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

Оператор not

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

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

будет эквивалентен запросу

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

Если медиа запрос составлен с использованием оператора запятая, то отрицание будет распространяться только на ту часть, которая идет до запятой, т.е. запрос

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

будет эквивалентен запросу

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

Оператор only (устарело)

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

```
media="only screen and (min-width: 401px) and (max-width: 600px)"
```

Эти браузеры ожидают список медиа-типов, разделённых запятыми. И, согласно спецификации, они должны отсекают каждое значение непосредственно перед первым неалфавитно-цифровым символом, который не является дефисом. Таким образом, старый браузер должен интерпретировать предыдущий пример как `media="only"`. Поскольку данного типа медиа-типа не существует, то и таблицы стилей будут игнорироваться.

Тип носителя

- **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
<code>width</code>	Проверяет ширину области просмотра. Значения задаются в единицах длины, <code>px</code> , <code>em</code> и т.д., например, <code>(width: 800px)</code> . Обычно для проверки используются минимальные и максимальные значения ширины. <code>min-width</code> применяет правило если ширина области просмотра больше значения, указанного в запросе, <code>max-width</code> — ширина области просмотра меньше значения, указанного в запросе.
<code>height</code>	Проверяет высоту области просмотра. Значения задаются в единицах длины, <code>px</code> , <code>em</code> и т.д., например, <code>(height: 500px)</code> . Обычно для проверки используются минимальные и максимальные значения высоты. <code>min-height</code> применяет правило если высота области просмотра больше значения, указанного в запросе, <code>max-height</code> — высота области просмотра которого меньше значения, указанного в запросе.
<code>aspect-ratio</code>	Проверяет соотношение ширины к высоте области просмотра. Широкоэкранный дисплей с соотношением сторон 16:9 может быть помечен как <code>(aspect-ratio: 16/9)</code> . <code>min-aspect-ratio</code> проверяет минимальное соотношение, <code>max-aspect-ratio</code> — максимальное соотношение ширины к высоте области просмотра.
<code>orientation</code>	Проверяет ориентацию области просмотра. Принимает два значения: <code>(orientation: portrait)</code> и <code>(orientation: landscape)</code> .
<code>resolution</code>	Проверяет разрешение экрана (количество пикселей). Значения также могут проверять количество точек на дюйм (<code>dpi</code>) или количество точек на сантиметр (<code>dpcm</code>), например, <code>(resolution: 300dpi)</code> . <code>min-resolution</code> проверяет минимальное разрешение экрана, <code>max-resolution</code> — максимальное.
<code>color</code>	Проверяет количество бит на каждый из цветовых компонентов устройства вывода. Например, <code>(min-color: 4)</code> означает, что экран конкретного устройства должен иметь 4-битную глубину цвета. <code>min-color</code> проверяет минимальное количество бит, <code>max-color</code> — максимальное количество бит.
<code>color-index</code>	Проверяет количество записей в таблице подстановки цветов. В качестве значения указывается положительное число, например, <code>(color-index: 256)</code> . <code>min-color-index</code> проверяет минимальное количество записей, <code>max-color-index</code> — максимальное количество записей.
<code>monochrome</code>	Проверяет количество битов на пиксель монохромного устройства. Значение задается целым положительным числом, например, <code>(min-monochrome: 8)</code> . <code>min-monochrome</code> проверяет минимальное количество битов, <code>max-monochrome</code> — максимальное количество битов.
<code>-webkit-device-pixel-ratio</code>	Задаёт количество физических пикселей устройства на каждый CSS-пиксель.

Viewport Meta Tag

`<meta name="viewport">` - это мета тег HTML, который используется для контроля масштабирования и расположения контента на мобильных устройствах. Он позволяет веб-разработчикам указать браузеру, как должна быть настроена область просмотра (viewport) страницы на мобильных устройствах.

Атрибуты мета тега viewport:

- `width`: Устанавливает ширину области просмотра в пикселях или в соотношении `device-width`, что означает ширину экрана устройства.
- `initial-scale`: Устанавливает начальный уровень масштабирования страницы.
- `minimum-scale`: Устанавливает минимальный уровень масштабирования страницы.
- `maximum-scale`: Устанавливает максимальный уровень масштабирования страницы.
- `user-scalable`: Указывает, разрешено ли пользователю масштабировать страницу (значения: `yes` - разрешено, `no` - запрещено).

Особенности

Мобильный браузер - немного кривой так как в учет 100vh берутся элементы управления браузера

На мобильном браузере нет понятия hover

На мобильном браузере можно менять ориентацию экрана - нужно помнить

Некоторые анимации и переходы на мобильном браузере плохо отрабатывают

Мобильный браузер скролл высчитывает очень близко к drag and drop, с чем также возникают проблемы

Размеры элементов в адаптивности

- fit-content
- max-content
- min-content

особые свойства:

- min-width
- max-width
- min-height
- max-height

fit-content:

- Это значение позволяет элементу автоматически изменять свою ширину или высоту, чтобы подстроиться под свое содержимое, но не превышать ширину или высоту своего родительского контейнера.
- Это особенно полезно, когда элемент содержит текст или другие блочные элементы, и вы хотите, чтобы его размеры автоматически регулировались в зависимости от контента.

```
.container {  
    width: fit-content;  
    height: fit-content;  
}
```

max-content:

- Это значение устанавливает ширину или высоту элемента таким образом, чтобы они были равны максимальному размеру его содержимого внутри блока.
- Это означает, что элемент будет расширяться до максимального размера своего содержимого в пределах своего родительского контейнера.

```
.container {  
    width: max-content;  
    height: max-content;  
}
```

min-content:

- Это значение устанавливает ширину или высоту элемента таким образом, чтобы они были равны минимальному размеру его содержимого внутри блока.
- Это означает, что элемент будет занимать минимально необходимое пространство для отображения всего своего содержимого.

```
.container {  
    width: min-content;  
    height: min-content;  
}
```

max & min width/height

Когда мы используем свойства CSS `min-width`, `max-width`, `min-height` и `max-height`, мы контролируем минимальные и максимальные размеры элементов веб-страницы. Эти свойства особенно полезны при создании адаптивных дизайнов, где нужно гибко реагировать на изменения размеров экрана или контента.

width

min-width:

Это свойство устанавливает минимальную ширину элемента.

При использовании в адаптивной верстке, `min-width` может быть полезным для предотвращения слишком сильного сжатия содержимого на маленьких экранах.

Например, вы можете установить `min-width: 320px`; для того чтобы гарантировать, что элемент не будет сжиматься меньше, чем на экранах мобильных устройств шириной 320 пикселей.

max-width:

Это свойство устанавливает максимальную ширину элемента.

При использовании в адаптивной верстке, `max-width` может быть полезным для предотвращения слишком широкого отображения содержимого на больших экранах.

Например, вы можете установить `max-width: 1200px`; для того чтобы ограничить ширину элемента на экранах с широким разрешением, таких как настольные компьютеры.

height

min-height:

Это свойство устанавливает минимальную высоту элемента.

При использовании в адаптивной верстке, `min-height` может быть полезным для гарантирования минимальной видимости содержимого на экранах с малым вертикальным пространством.

Например, вы можете установить `min-height: 100vh`; для того чтобы гарантировать, что элемент будет занимать как минимум всю высоту экрана (100% видимой области).

max-height:

Это свойство устанавливает максимальную высоту элемента.

При использовании в адаптивной верстке, `max-height` может быть полезным для предотвращения слишком большой высоты элемента на экранах с большим вертикальным пространством.

Например, вы можете установить `max-height: 500px`; для того чтобы ограничить высоту элемента на маленьких экранах, чтобы не занимать слишком много места.

Изображения и мультимедиа

В адаптивной верстке использование изображений требует особого внимания, чтобы они хорошо отображались на различных устройствах и разрешениях экранов.

Более того картинки высокого разрешения потребляют больше ресурсов для подгрузки, что в некоторых случаях не удобно

Атрибут srcset:

srcset - это атрибут элемента ``, который позволяет вам предоставить браузеру несколько вариантов изображений разного размера или разрешения для выбора наиболее подходящего изображения в зависимости от характеристик экрана устройства, таких как плотность пикселей (DPI) или размер экрана.

Когда браузер загружает веб-страницу на устройстве с различным разрешением экрана, он может выбрать наиболее подходящее изображение из набора, предоставленного в атрибуте srcset. Это позволяет улучшить производительность и оптимизировать использование данных, загружая изображения оптимизированных размеров для конкретного устройства.

srcset

```

```

В атрибуте srcset перечисляются пути к изображениям и их размеры. Размеры указываются в формате {URL} {размер}, где размер может быть задан в пикселях (например, 500w) или в устройственных пикселях (500х).

Размеры используются для того, чтобы браузер мог выбрать наиболее подходящее изображение для текущего разрешения экрана.

Альтернатива для своей реализации

На уровне JS - создать свой элемент `img` который будет принимать решения какую картинку подгружать на основе своей логики (логику описываем мы)

На уровне фреймворка/библиотеки - реализуем свой компонент который также будет принимать набор изображений разного разрешения и принимать решение об отображении (рендере) на основе своей логики (логику описываем мы)

Учитывая современный интернет возможно это излишняя оптимизация

Если на странице много картинок то лучше все же оптимизировать

Элемент <picture>

Элемент <picture> позволяет указывать несколько источников изображений для различных условий отображения, таких как разрешение экрана, ориентация и т. д.

Внутри элемента <picture> могут быть указаны несколько элементов <source>, каждый из которых представляет собой источник изображения для определенных условий отображения.

Если условия <source> не совпадают, браузер будет использовать элемент в качестве запасного варианта.

```
<picture>
  <source media="(min-width: 768px)" srcset="large.jpg">
  <source media="(min-width: 480px)" srcset="medium.jpg">
  
</picture>
```

filter - СВОЙСТВО

- `blur()` Значение задается в единицах длины, например `px`, `em`. Применяет размытие по Гауссу к исходному изображению. Чем больше значение радиуса, тем больше размытие. Если значение радиуса не задано, по умолчанию берется 0.
- `brightness()` Значение задается в % или в десятичных дробях. Изменяет яркость изображения. Чем больше значение, тем ярче изображение. Значение по умолчанию 1.
- `contrast()` Значение задается в % или в десятичных дробях. Регулирует контрастность изображения, т.е. разницу между самыми темными и самыми светлыми участками изображения/фона. Значение по умолчанию 100%. Нулевое значение скроет исходное изображение под темно-серым фоном. Значения, увеличивающиеся от 0 до 100% или от 0 до 1, будут постепенно открывать исходное изображение до оригинального отображения, а значения свыше будут увеличивать контраст между светлыми и темными участками.
- `drop-shadow()` Фильтр действует подобно свойствам `box-shadow` и `text-shadow`. Использует следующие значения: смещение по оси X смещение по оси Y размытость растяжение цвет тени. Отличительная особенность фильтра заключается в том, что тень добавляется к элементам и его содержимому с учетом их прозрачности, т.е. если элемент содержит текст внутри, то фильтр добавит тень одновременно для текста и видимых границ блока. В отличие от других фильтров, для этого фильтра обязательно задание параметров (минимальное — величина смещения).
- `grayscale()` Извлекает все цвета из картинки, делая на выходе черно-белое изображение. Значение задается в % или десятичных дробях. Чем больше значение, тем сильнее эффект.
- `hue-rotate()` Меняет цвета изображения в зависимости от заданного угла поворота в цветовом круге. Значение задается в градусах от 0deg до 360deg. 0deg — значение по умолчанию, означает отсутствие эффекта.
- `invert()` Фильтр делает негатив изображения. Значение задается в %. 0% не применяет фильтр, 100% полностью преобразует цвета.
- `opacity()` Фильтр работает аналогично со свойством `opacity`, добавляя прозрачность элементу. Отличительная особенность — браузеры обеспечивают аппаратное ускорение для фильтра, что позволяет повысить производительность. Дополнительный бонус — фильтр можно одновременно сочетать с другими фильтрами, создавая при этом интересные эффекты. Значение задается только в %, 0% делает элемент полностью прозрачным, а 100% не оказывает никакого эффекта.
- `saturate()` Управляет насыщенностью цветов, работая по принципу контрастного фильтра. Значение 0% убирает цветность, а 100% не оказывает никакого эффекта. Значения от 0% до 100% уменьшают насыщенность цвета, выше 100% — увеличивают насыщенность цвета. Значение может задаваться как в %, так и целым числом, 1 эквивалентно 100%.
- `sepia()` Эффект, имитирующий старину и «ретро». Значение 0% не изменяет внешний вид элемента, а 100% полностью воспроизводит эффект сепии.
- `url()` Функция принимает расположение внешнего XML-файла с svg-фильтром, или якорь к фильтру, находящемуся в текущем документе.
- `none` Значение по умолчанию. Означает отсутствие эффекта.
- `initial` Устанавливает это свойство в значение по умолчанию.
- `inherit` Наследует значение свойства от родительского элемента.

Ресурсы

@media - документация MDN - [ТЫК](#)

media запросы статья на русском - [ТЫК](#)

media html документация - [ТЫК](#)

фильтры filter статья на русском - [ТЫК](#)

адаптивные изображения - [ТЫК](#)

статья про адаптивные изображения - [ТЫК](#)