

Медиа-запросы CSS

Такое понятие, как медиа-запросы, появилось в CSS3. С их помощью можно определять внешний вид веб-страниц, опираясь на ширину и высоту окна браузера. Благодаря медиа-запросам вы можете параллельно создавать стили для небольших мобильных телефонов, планшетов, ноутбуков и мониторов с большой диагональю.

Идея адаптивного дизайна — предоставить посетителям сайт в наиболее читаемой и привлекательной форме. Дизайн создается под как минимум три целевых экрана: настольный ПК, планшет и смартфон. Но учитывая большое количество вариаций этих девайсов, недостаточно остановиться на какой-то одной ширине для каждого типа устройства. Поэтому важно использовать гибкую сетку и проводить тестирование в процессе разработки, отслеживая то, как подстраивается дизайн сайта под разную ширину браузерного окна.

Когда использовать медиа-запросы?

Как правило, для адаптации дизайна веб-страницы необходимо прописать ряд CSS-стилей для определенных элементов. Ниже мы перечислим наиболее частые сценарии, когда вам понадобится обратиться к медиа-запросам.

- **Определение количества колонок.** Трехколоночный макет сайта, который шикарно смотрится на настольных ПК, будет совсем неуместен для мобильных телефонов и планшетов. С помощью медиа-запросов можно переопределить количество колонок в макете с учетом ширины экрана устройства, сгруппировав для смартфонов весь контент в одну колонку.
- **Относительная ширина.** Когда вы устанавливаете фиксированную ширину макета, скажем, в 960 пикселей, это подойдет для настольных компьютеров, ноутбуков и некоторых особо крупных планшетов. Однако для мобильного телефона такой размер контейнера слишком большой. Используя медиа-запрос, можно специально для смартфонов создать «резиновый» макет, ширина которого будет не фиксированной, а относительной. Таким образом, контейнер шириной 100% будет легко подстраиваться под любой смартфон в любой ориентации.
- **Уменьшение отступов.** Большие расстояния между блоками и элементами создают ощущение воздушного, легкого дизайна. Однако отступы, которые хорошо смотрятся на мониторе с диагональю 21-27 дюймов, будут выглядеть абсолютно безобразно на небольшом экране смартфона, к тому же принуждая пользователей больше прокручивать страницу. Медиа-запросы позволяют задать альтернативные размеры отступов для узких дисплеев.
- **Регулировка размера шрифта.** Заголовки высотой в 60 пикселей могут красиво смотреться на десктопе, но для мобильных устройств такой размер шрифта редко когда уместен, поэтому для узких экранов будет целесообразным установить меньшее значение *font-size*.
- **Адаптивная навигация.** Нередко бывает так, что навигационное меню, которое было продумано для десктопной версии сайта, совершенно не подходит для мобильных устройств ввиду своей объемности либо манеры расположения пунктов меню. Существует несколько подходов к реализации удобной навигации для мобильной версии сайта, и медиа-запросы непременно принимают в этом участие.
- **Скрытие элементов.** Некоторые части веб-страницы могут быть бесполезными либо малозначимыми, если сайт просматривается с мобильного телефона. С помощью свойства *display: none* и медиа-запросов вы можете скрыть определенные элементы от глаз посетителя, если он зашел на сайт со смартфона. Но имейте в виду, что подобный прием не экономит трафик: скрытые элементы по-прежнему будут загружаться браузером.

Конечно, это далеко не все ситуации, когда применяются медиа-запросы.

Breakpoints (контрольные точки)

При помощи медиа-запросов вы можете создавать так называемые контрольные точки (англ. *breakpoints*) и привязывать к ним CSS-стили. К примеру, можно определить для браузера следующие правила: «Если ширина экрана составляет больше чем 767 пикселей, к веб-странице применяются эти стили, а если ширина экрана составляет больше чем 991 пиксель, применяются другие стили». Вот эти числа, обозначающие ширину экрана, и называются контрольными точками.

Какую ширину необходимо указывать в контрольных точках? Наиболее простой ответ: ту, на которой верстка веб-страницы начинает ломаться. Представим, что у вас есть сайт с контейнером шириной 1180 пикселей. Тогда в окне шириной 1200 пикселей (учитываем полосу прокрутки) или меньше он, скорее всего, будет выглядеть не очень хорошо. Вот и ваша первая контрольная точка: необходимо внести правки в дизайн, если сайт просматривается в окне шириной менее чем 1200 пикселей.

При создании гибких сеток часто применяется определенный набор медиа-запросов, предусмотренный для трех различных контрольных точек: смартфон, планшет, монитор ПК. Контрольных точек может быть и больше. Например, в Bootstrap 4 используется четыре контрольных точки в одном направлении и еще четыре — в обратном направлении:

```
/* ----- */
/* ----- От маленьких экранов к большим ----- */
/* ----- */

/* Стили для очень маленьких экранов (телефоны в книжной ориентации,
дисплей менее 576 пикселей). Записываются без медиа-запросов,
поскольку в Bootstrap это значение по умолчанию */

/* Стили для маленьких экранов (телефоны в альбомной ориентации,
дисплей 576 пикселей и более) */
@media (min-width: 576px) { ... }

/* Стили для средних экранов (планшеты, дисплей 768 пикселей и более) */
@media (min-width: 768px) { ... }

/* Стили для больших экранов (ПК, дисплей 992 пикселей и более) */
@media (min-width: 992px) { ... }

/* Стили для очень больших экранов (ПК с большим монитором,
дисплей 1200 пикселей и более) */
@media (min-width: 1200px) { ... }

/* ----- */
/* ----- От больших экранов к маленьким ----- */
/* ----- */

/* Стили для очень больших экранов (дисплей 1200 пикселей и более).
Записываются без медиа-запросов, так как контрольная точка
для очень больших экранов не имеет верхней границы по ширине */

/* Стили для больших экранов (дисплей максимум 1199 пикселей) */
@media (max-width: 1199px) { ... }

/* Стили для средних экранов (дисплей максимум 991 пикселей) */
@media (max-width: 991px) { ... }

/* Стили для маленьких экранов (дисплей максимум 767 пикселей) */
@media (max-width: 767px) { ... }

/* Стили для очень маленьких экранов (дисплей максимум 575 пикселей) */
@media (max-width: 575px) { ... }
```

Кроме этого, в Bootstrap задействуются еще несколько вариаций контрольных точек для записи стилей под конкретный диапазон ширины экранов. При этом используются сразу оба условия — для минимума и для максимума:

```
/* Стили для очень маленьких экранов (дисплей максимум 575 пикселей) */
@media (max-width: 575px) { ... }

/* Стили для маленьких экранов (дисплей шириной от 576 пикселей до 767
пикселей) */
```

```

@media (min-width: 576px) and (max-width: 767px) { ... }

/* Стили для средних экранов (дисплей шириной от 768 пикселей до 991
пикселей) */
@media (min-width: 768px) and (max-width: 991px) { ... }

/* Стили для больших экранов (дисплей шириной от 992 пикселей до 1199
пикселей) */
@media (min-width: 992px) and (max-width: 1199px) { ... }

/* Стили для очень больших экранов (дисплей минимум 1200 пикселей) */
@media (min-width: 1200px) { ... }

```

Mobile First или Desktop First?

Когда вы пишете стили для контрольных точек, вы не создаете весь дизайн с нуля, а лишь корректируете существующую верстку. Возникает вопрос, под что верстать изначально: под настольные компьютеры с последующей корректировкой под мобильные устройства или же наоборот — сначала создать версию для небольших экранов, а затем расширить ее? Рассмотрим оба подхода.

- Desktop First.** При этом подходе вы сначала полностью верстаете дизайн для больших экранов, без использования медиа-запросов. После этого, используя контрольные точки в медиа-запросах, корректируете дизайн под средние и маленькие экраны (уменьшаете шрифт, перестраиваете макет, скрываете второстепенные элементы и т. п.). Преимущество такой верстки заключается в том, что старые браузеры (к примеру, Internet Explorer 8), которые не знают, что такое медиа-запросы, смогут отобразить ваш сайт, поскольку это исходный дизайн, который вы писали без использования директивы `@media`.
- Mobile First.** Если вы выбираете данный подход, то сначала верстаете дизайн для самых маленьких экранов, не используя медиа-запросы. После этого, создавая контрольные точки, вы корректируете верстку под все более и более широкие экраны.

Оба подхода подразумевают сначала создание определенного базового набора стилей, актуальных для любой версии сайта (например, цвет ссылок, размер основного шрифта и др.), а затем написание тех стилей, которые будут отличаться для различных экранов.

Синтаксис медиа-запросов

Чтобы добавить медиа-запрос в таблицу стилей, запишите следующее:

```

@media (min-width: 576px) {
  /* здесь будут CSS-стили */
}

```

Внутри скобок медиа-запроса поместите стили CSS, как вы делаете это обычно:

```

@media (min-width: 576px) {
  .column {
    width: 100%;
  }
  h1 {
    font-size: 24px;
  }
  /* и так далее... */
}

```

Желательно размещать все медиа-запросы под обычными стилями, а не над ними. Если вы используете контрольные точки, отталкиваясь от минимальной ширины экрана (`min-width`), размещайте их в порядке возрастания ширины экрана. Если вы применяете `max-width`, тогда расположите медиа-запросы в порядке уменьшения ширины. В противном случае одни медиа-запросы будут перезаписывать предыдущие.