

Основные виды верстки сайта

Фиксированная

В данном случае макет разрабатывается с фиксированными параметрами (например, с точным указанием размеров в пикселях). Такой подход устарел и применяется редко.

Мобильная

Сайт создается на выделенном поддомене и адаптируется только под мобильные устройства. Соответственно, ресурс станет удобен для чтения лишь пользователям смартфонов и планшетов. Мобильная верстка сайта по сравнению с адаптивной менее затратна. Выбирать этот вариант целесообразно, когда использование ресурса с компьютера или ноутбука изначально не предусматривается.

Резиновая

Ширина блока задается в процентах от ширины экрана пользователя. Благодаря этому «резиновые» сайты одинаково хорошо читаются с любых устройств. Тем не менее, адаптивная верстка, равно как и отзывчивая, постепенно вытесняет резиновую.

Отзывчивая (Responsive)

Во многом напоминает адаптивную верстку, но в то же время принципиально отличается от нее. Отличие заключается в действиях на брейк-поинтах. Если в адаптивном варианте браузер резко меняет сверстаный макет, то в отзывчивом происходит плавное изменение верстки между точками слома с максимальным сохранением стабильного вида страницы. Фактически данный подход сочетает в себе признаки адаптивности и «резиновости».

Адаптивная

Основные параметры сайта подстраиваются под характеристики пользовательского устройства. Ресурс в результате должен удобно читаться практически с любого гаджета или компьютера.

В процессе адаптивной верстки сайта HTML код пишется под несколько вариантов разрешений экрана. Как минимум, страницы адаптируют под ПК, ноутбуки, планшеты и смартфоны. Используемые при этом брейк-поинты (точки слома) сообщают браузеру места кода, где сайт необходимо трансформировать для обеспечения максимальной читаемости.

Адаптивный и отзывчивый дизайн

- Отзывчивый (Responsive) дизайн (RWD) создается с гибкой сеткой или другими подходящими, но статическими свойствами, позволяющими одному шаблону быть актуальным для нескольких гаджетов.
- Адаптивный (Adaptive) дизайн (AWD) проектируется с условиями, изменяющимися, исходя из устройства серфинга, в его базе есть уже ряд макетов, он динамический и ориентирован на диагональ экранов.

Во случае адаптивного дизайна во главу угла ставится функциональность – особенности устройств учитываются в обязательном порядке. От размеров дисплея напрямую зависит характер расположения блоков. При разработке отзывчивого дизайна ведущую роль играют медиазапросы и относительная сетка, заданная в процентах.

Есть разница и в работе серверных скриптов: при AWD они сначала определяют, с чего заходит человек, а потом загружают оптимизированную под устройство версию страницы.

Что лучше использовать? Зависит от структуры web-ресурса: если она «резиновая», отзывчивость будет весьма удобной, вот только от нее может снизиться скорость загрузки (при наличии большого количества визуальных элементов). Адаптивность сайта – это мощный инструмент во всех остальных случаях: он развязывает дизайнерам руки, дает возможность разделить пользователей на категории и предложить каждой ориентированные на нее решения. Проще говоря, владельцы смартфонов будут видеть один макет, созданный именно для их удобства, планшетов – другой, стационарных ПК – третий.



Медиазапросы

Одним из самых важных инструментов при создании адаптивной вёрстки является использование медиазапросов. Медиазапросы — специальные условные конструкции, которые позволяют применять стили только для определённых устройств.

Медиазапросы могут записываются как в css-файле, так и при подключении css-файла

Использование внутри css-файла:

```
@media (условия) {  
    /* CSS-код, который будет выполнен для данного условия */  
}
```

В качестве условия могут выступать различные значения и константы.

Медиазапросы: ориентация экрана

Для определения ориентации экрана используется ключ `orientation`, значением которого может выступать одно из двух значений:

- ❑ `landscape` — условие выполнится для устройств с горизонтальной ориентацией экрана. Горизонтальная — ориентация, при которой ширина viewport больше его высоты.
- ❑ `portrait` — условие выполнится для устройств с вертикальной ориентацией экрана. Вертикальная — ориентация, при которой высота viewport больше его ширины.

```
@media (orientation: landscape) {  
    /* При горизонтальной ориентации фоновым цветом сайта будет белый */  
    body {  
        background: #FFF;  
    }  
}
```

Медиазапросы: разрешение экрана

При использовании медиазапросов мы также можем исходить из ширины или высоты `viewport`. Для этого используются знакомые нам по обычным CSS-правилам условия `width`, `min-width`, `max-width` для ширины и `height`, `min-height`, `max-height` для высоты.

С помощью таких условий создаются `breakpoint` — контрольные точки. Это границы значений, по которым видоизменяется наш макет. Такие точки остановки позволяют иметь правила для мониторов, планшетов, телефонов и иных смарт-устройств.

CSS является каскадной таблицей, поэтому порядок стилей необходимо контролировать. Существует два подхода — `Desktop First` и `Mobile First`.

Desktop First. Вначале пишутся стили для больших мониторов, а в последствии, используя медиазапросы, дописываются стили для всё более маленьких значений `viewport`. Его характерная черта в медиазапросах — использование конструкции `max-width` в качестве условия.

Mobile First. Его особенностью является то, что вначале пишутся стили под мобильные устройства, а затем, используя медиазапросы, пишутся стили для больших размеров `viewport`. Если в `Desktop First` основной конструкцией являлось использование `max-width`, то в `Mobile First` используется `min-width`.

Медиазапросы: разрешение экрана

```
<style>
  /* Здесь будут все стили для устройств с
  viewport больше 1400 пикселей. */

  @media (max-width: 1400px) {
    /* Стили для устройств, у которых ширина
    viewport меньше или равно 1400 пикселей, но
    больше 990 пикселей. Эти стили будут
    использованы для планшетов и ноутбуков с низким
    разрешением */
  }

  @media (max-width: 990px) {
    /* Стили для устройств, у которых ширина
    viewport меньше или равно 990 пикселей, но
    больше 770 пикселей. Эти стили подойдут для
    некоторых мобильных устройств и планшетов */
  }

  @media (max-width: 770px) {
    /* Стили для устройств, у которых ширина
    viewport меньше или равно 770 пикселей. Это
    множество мобильных устройств */
  }
</style>
```

```
<style>
  /* Здесь будут все стили для мобильных
  устройств с viewport меньше 770 пикселей. */

  @media (min-width: 770px) {
    /* Стили для устройств, у которых ширина
    viewport больше или равно 770 пикселей*/
  }

  @media (min-width: 990px) {
    /* Стили для устройств, у которых ширина
    viewport больше или равно 990 пикселей, но
    меньше 1400 пикселей */
  }

  @media (min-width: 1400px) {
    /* Стили для устройств, у которых ширина
    viewport больше или равно 1400 пикселей*/
  }
</style>
```

Медиазапросы: логические операторы

Условия внутри медиазапросов можно комбинировать. Для этого существует три логических оператора: И, ИЛИ, НЕ.

Логическое «И». Означает, что несколько условий должны быть выполнены для того, чтобы CSS-стили применились к элементу. Для использования логического «И» используется ключевое слово `and`.

```
@media (orientation: portrait) and (min-width: 600px) { }
```

Логическое «ИЛИ». Свойства применятся в том случае, если хотя бы одно из условий будет выполнено. Условия для этого отделяются запятыми.

```
@media (orientation: portrait), (min-width: 600px) { }
```

Логическое «НЕ». Свойства применятся в том случае, если условие не выполняется. Используется ключевое слово `not`. Реальное использование этого оператора не велико, в виду сложности и не интуитивности происходящего.

Медиазапросы: типы устройств

В `@media` можно указывать определённые типы устройств:

- ❑ `all` — для всех;
- ❑ `print` — для принтеров и в режиме предварительного просмотра страницы перед печатью;
- ❑ `screen` — для устройств с экранами;
- ❑ `speech` — для программ чтения с экрана.

```
@media screen { ... }
```

```
@media screen, print { ... }
```

Использование медиазапросов при подключении CSS

Медиазапросы возможно писать не только внутри CSS-файла, но и использовать их в HTML при подключении файла стилей. В этом случае медиазапрос указывается в атрибуте `media`.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Подключение CSS файлов</title>
  <!-- Общие стили для проекта -->
  <link rel="stylesheet" href="style.css">
  <!-- Стили для экранов с viewport не менее 750px -->
  <link rel="stylesheet" media="screen and (min-width: 750px)" href="style750px.css">
</head>
```

Кроме `<link>`, их также можно использовать в `@import`:

```
@import url(mobile.css) screen and (max-width: 991.98px);
@import url(desktop.css) screen and (min-width: 992px);
```