



## Урок 7

# Разметка сайта и знакомство с Bootstrap

Создание основной разметки сайта. Заполнение разметки частями нарезанного макета. Приёмы позиционирования элементов. Знакомство и применение Bootstrap.

### Оглавление

[Первый способ прижатия футера к низу экрана](#)

[Способ первый](#)

[Недостаток данного приема](#)

[Второй способ](#)

[calc\(\)](#)

[vh](#)

[Недостаток данного приема](#)

[Выводы](#)

[Медиа-запросы](#)

[Синтаксис](#)

[Плюсы](#)

[Условия для Media Queries](#)

[Bootstrap](#)

[Основные преимущества](#)

[Основные инструменты](#)

[Домашнее задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

# Первый способ прижатия футера к низу экрана

Все пользователи, привыкшие к полноценно оформленным страницам сайтов, предпочитают вид «прижатого» (sticky) к низу страницы футера. Например, когда открываем короткие по высоте страницы сразу бросается в глаза, что информация, призванная быть в нижней части окна просмотра, прилипает к содержанию и находится где-то посередине, а то и в верхней части окна, когда внизу — пусто. Что означает «прижать footer к низу экрана»? Это означает, что когда содержимого страницы, а именно контента, который находится между хедером (шапкой) и футером (подвалом) мало, то footer все равно остается внизу экрана, а точнее внизу окна браузера. Так вот, прижать footer к низу экрана означает, что как бы вы не изменяли размер окна браузера, footer всегда будет внизу окна. Если размер окна не будет вмещать в себе контент и футер, тогда появиться прокрутка.

Почему нужно прижать footer к низу экрана? Это требование по умолчанию, норма, которую нужно запомнить и исполнять, тем более, что прижать footer к низу экрана довольно просто. Иногда дизайнер может сделать макет, в котором не нужно прижимать footer к низу экрана, но не делать то, что вы уже умеете легко. В 99% при создании сайтов нужно прижать footer к низу экрана (к низу окна браузера). Кроме этого, прижать footer к низу экрана (к низу окна браузера) нужно, чтобы сайт корректно отображался по высоте на разных мониторах и расширениях экрана.

Запомните: прижать footer к низу экрана (к низу окна браузера) означает корректное отображение сайта по высоте, несмотря на количество контента, на разных мониторах: например, на 13-дюймовом нетбуке, или на 24-дюймовом настольном мониторе. Также прижать footer к низу экрана (к низу окна браузера) означает не беспокоиться о том, какое расширения экрана стоит у пользователя, 1920 на 1080 или 800 на 600 - сайт будет правильно отображаться по высоте при любом расширении, от которого напрямую зависит размер окна браузера.

## Способ первый

- подвал прижат к низу экрана, при высоте окна браузера больше высоты страницы, независимо от контента;
- футер находится на положенном ему месте при объеме контента больше, чем высота окна браузера;
- работает во всех популярных браузерах.

Для примера возьмем простую страницу, состоящую из трех основных блоков: шапки (header), основного содержания (content) и подвала (footer). Сделаем так, чтобы основной блок и шапка заняли всю площадь окна браузера, независимо от количества контента, при этом футер прижмём к низу экрана так, чтобы в браузере не появилась вертикальная полоса прокрутки.

```
<div class="container">
  <div class="header"></div>
  <div class="content">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Adipisci iure
  tempore similique inventore, aliquam dicta placeat.</div>
</div>
<div class="footer"></div>
```

Основной контейнер (.container) растягиваем на всю высоту экрана браузера (min-height:100%), подвалу жестко указываем высоту (height). При этом общая высота сайта составит высота экрана + высота подвала.

Отрицательным отступом (margin-top) "въезжаем" в основной блок, чтобы высота сайта составляла только 100% высоты экрана.

```
* { margin: 0; padding: 0; }
html, body { height: 100%; }
.container { min-height: 100%; }
.header {
  height: 100px;
  background-color: orange;
}
.content { background-color: green; }
.clr { height: 100px; }
.footer {
  height: 100px;
  background-color: red;
  margin-top: -100px;
  clear: both;
}
```

При таком расположении блоков и при достаточном количестве контента (например, текста) в основном блоке, возможны накладки контента в основном блоке на подвал. Для избежания этой потенциальной проблемы, добавляем в конец основного блока пустой блок, высотой не меньше высоты подвала (а лучше немного больше, чтобы был отступ между контентом и подвалом).

При использовании блочной верстки и плавающих основных блоков (колонок) для .footer следует добавить clear: both, чтобы подвал расположился под колонками.

```
<div class="container">
  <div class="header"></div>
  <div class="content">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Adipisci iure
  tempore similique inventore, aliquam dicta placeat.</div>
  <div class="clr"></div>
</div>
<div class="footer"></div>
```

Теперь, если контента будет много, он будет двигать пустой блок вниз. А это будет опускать и подвал, не давая налезть на него контенту.

Если вы уже немного освоили CSS, может возникнуть вопрос: "Зачем использовать дополнительный элемент, если можно воспользоваться padding-bottom?". Ответ: так просто его здесь использовать нельзя, т.к. размер блока равен его ширине и высоте + сумме внутренних отступов + сумме толщин бордюров. Связка min-height: 100% и padding-bottom даст высоту сайта больше высоты экрана. В итоге, даже при отсутствии контента вовсе, подвал будет за пределами "первого экрана"

## Недостаток данного приема

- В вышеописанном приеме футер поднимали отрицательным отступом вверх. При этом возникает потенциальная проблема с z-слоями;
- Годится только для фиксированного по высоте подвала.

## Второй способ

Особенность данного способа заключается в использовании calc() и единицы измерения vh, которые поддерживаются только современными браузерами.

### calc()

Используется для указания вычисляемого значения свойств, которые в качестве значений используют размер. Это позволяет задавать значения, основанные на сложении или вычитании разных единиц измерений, например, можно задать 100% - 20px. Если значение не может быть вычислено, оно игнорируется.

Браузеры:

- Firefox до версии 16.0 поддерживает значение -moz-calc.
- Chrome до версии 26.0 поддерживает значение -webkit-calc.
- Safari с версии 6.0 поддерживает значение -webkit-calc.
- Internet Explorer 9.0

## vh

Относительная единица измерения. 1% от высоты области просмотра.

html

```
<div class="content">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Adipisci iure tempore  
similique inventore, aliquam dicta placeat.</div>  
<div class="footer"></div>
```

css

```
* {  
margin: 0;  
padding: 0;  
}  
.content {  
background-color: blue;  
min-height: calc(100vh - 100px);  
}  
.footer {  
background-color: green;  
height: 100px;  
}
```

100vh - высота окна браузера.

100px - высота футера.

### Недостаток данного приема

- Поддержка только современными браузерами;
- Годится только для фиксированного по высоте подвала.

## Выводы

Существует огромное количество способов прижатия футера к низу экрана, мы рассмотрели некоторые из них, на практике вам покажут несколько методов, как разобранных в методичке, так и альтернативных. Прижатый футер - это правило хорошего тона. Сделать качественный сайт - важная часть веб-разработки.

# Медиа-запросы

Наряду с типами носителей, в CSS3 включена поддержка различных технических параметров устройств, на основе которых требуется загружать те или иные стили. К примеру, можно определить смартфон с максимальным разрешением 640 пикселей и для него установить одни стилевые свойства, а для остальных устройств другие. Также можно выявить различные характеристики вроде наличия монохромного экрана, ориентации (портретная или альбомная) и др. Все характеристики легко комбинируются, поэтому допустимо задать стиль только для устройств в альбомной ориентации с заданным разрешением экрана.

Возможности медиа-запросов не ограничиваются выявлением мобильных устройств, с их помощью можно создавать адаптивный макет. Такой макет подстраивается под разрешение монитора и окна браузера, меняя при необходимости ширину макета, число колонок, размеры изображений и текста. Медиа-запросы ограничивают ширину макета и, при достижении этого значения (к примеру, за счёт уменьшения окна или при просмотре на устройстве с указанным размером), уже применяется другой стиль.

## Синтаксис

Все запросы начинаются с правила `@media`, после чего следует условие, в котором используются типы носителей, логические операторы и медиа-функции.

## Плюсы

- четкое отображение страниц на экране с любым разрешением;
- возможность просмотра группы контента на любом устройстве;
- отсутствие горизонтальной полосы прокрутки независимо от размера окна.

## Условия для Media Queries

```
@media screen and (max-width: XXXpx) {}  
  
@media screen and (min-width: XXXpx) {}  
  
@media screen and (min-width: XXXpx) and (max-width: YYYpx) {}  
  
@media screen and (max-device-width: XXXpx) {}
```

С их помощью мы можем отслеживать разрешение экрана пользователя и отображать необходимые стили для каждого разрешения или устройства.

Пример:

```
@media screen and (min-width: 1024px) {  
    .content {  
        background-color: blue;  
    }  
}
```

В результате, если у пользователя экран больше или равен 1024px, то задний фон для content будет синим.

# Bootstrap

Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. Bootstrap использует самые современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров.

## Основные преимущества

- Экономия времени — Bootstrap позволяет сэкономить время и усилия, используя шаблоны дизайна и классы, и сконцентрироваться на других разработках;
- Высокая скорость — динамические макеты Bootstrap масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке;
- Гармоничный дизайн — все компоненты платформы Bootstrap используют единый стиль и шаблоны с помощью центральной библиотеки. Дизайн и макеты веб-страниц согласуются друг с другом;
- Простота в использовании — платформа проста в использовании, пользователь с базовыми знаниями HTML и CSS может начать разработку с Twitter Bootstrap;
- Совместимость с браузерами — Twitter Bootstrap совместим с Mozilla Firefox, Yandex Browser, Google Chrome, Safari, Internet Explorer и Opera;
- Открытое программное обеспечение — особенность Twitter Bootstrap, которая предполагает удобство использования, посредством открытости исходных кодов и бесплатной загрузки.

## Основные инструменты

- Сетки — заранее заданные размеры колонок, которые можно сразу же использовать, например, ширина колонки 140px относится к классу `.span2` (`.col-md-2` в третьей версии фреймворка), который можно использовать в CSS описании документа.
- Шаблоны — фиксированный или резиновый шаблон документа.
- Типографика — описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.
- Медиа — представляет некоторое управление изображениями и видео.
- Таблицы — средства оформления таблиц, вплоть до добавления функциональности сортировки.
- Формы — классы для оформления форм и некоторых событий происходящих с ними.
- Навигация — классы оформления для табов, вкладок, страничности, меню и тулбара.
- Алерты — оформление диалоговых окон, подсказок и всплывающих окон.

# Домашнее задание

## Создание новых страниц интернет-магазина

1. Доделать главную страницу, если не доделали.
2. Создание новых страниц по желанию, выбор следующих страниц, произвольный
3. \* Познакомиться с сеткой, присутствующей в макете.

Задачи со \* предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

## Дополнительные материалы

1. [Несколько способов прижатия футера к низу экрана](#)
2. [Как прижать подвал к низу страницы](#)
3. [Прижатый футер своими руками](#)
4. [Адаптивный дизайн на основе медиа запросов](#)
5. [Улучшение отзывчивости с помощью медиазапросов CSS](#)
6. [Адаптивная вёрстка](#)
7. [Bootstrap от новичка до профессионала](#)

## Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://www.zakharov.ms/footer/>
2. <http://htmlbook.ru/css/value/media>
3. <https://developers.google.com/web/fundamentals/design-and-ui/responsive/fundamentals/use-media-queries?hl=ru>
4. <http://www.wcdt.com.ua/yak-prytysnuty-footer-do-nyzu-ekrana-za-dopomohoyu-sss/>
5. <http://alexdev.ru/1497/>
6. <http://getbootstrap.com/>