

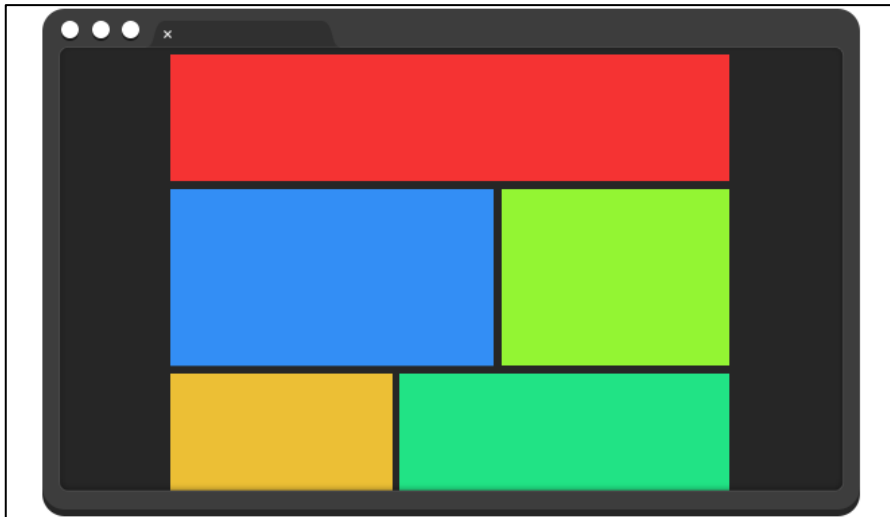
Лекция 7

Основы адаптивной вёрстки

Отзывчивый дизайн

Сайт с **отзывчивым дизайном** реагирует на изменение ширины браузера/экрана, подстраивая размещение элементов на сайте в соответствии с доступным пространством.

Если вы откроете **отзывчивый сайт** на полный экран, а затем измените размер окна браузера, контент будет динамически перемещаться, чтобы упорядочить себя оптимально для нового размера окна браузера. На мобильных телефонах этот процесс происходит автоматически: сайт проверяет наличие свободного места (анализирует размер экрана устройства) и затем отображает себя с идеальным расположением элементов.



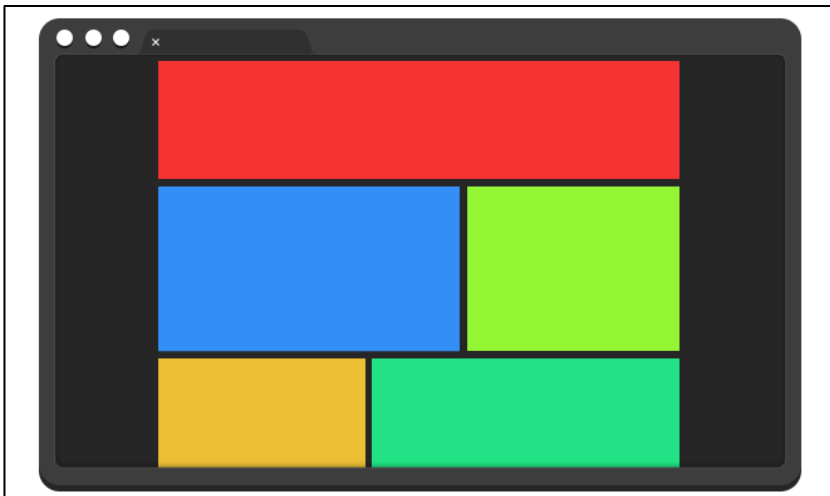
При изменении размера экрана отзывчивый дизайн выдаст плавный переход между возможными вариантами отображения.

Подробнее: <https://www.uprock.ru/articles/otzyvchivyy-dizayn-luchshie-praktiki-i-primery>

Адаптивный дизайн

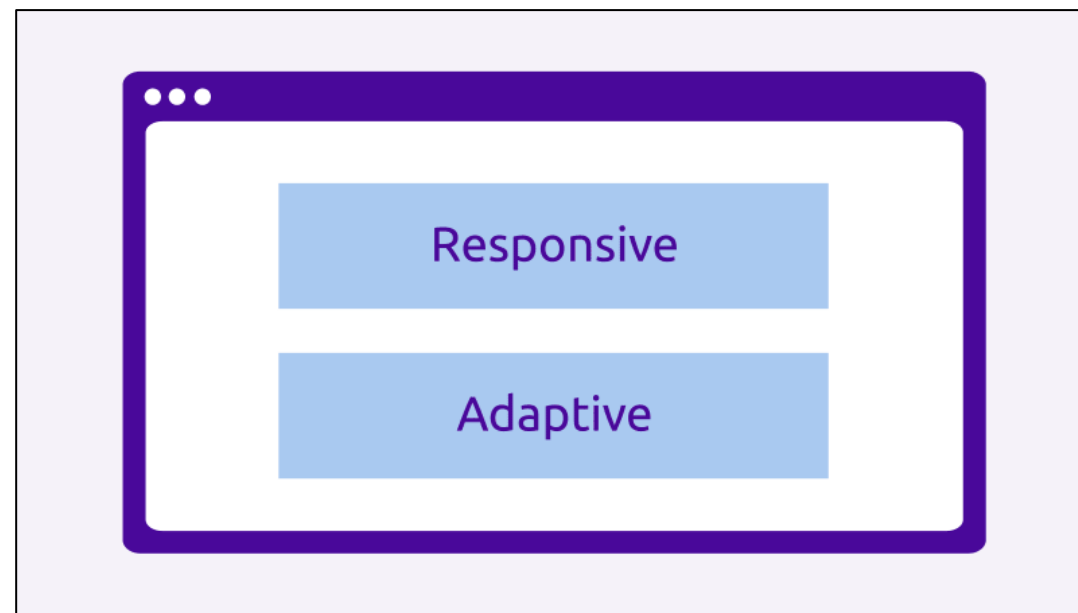
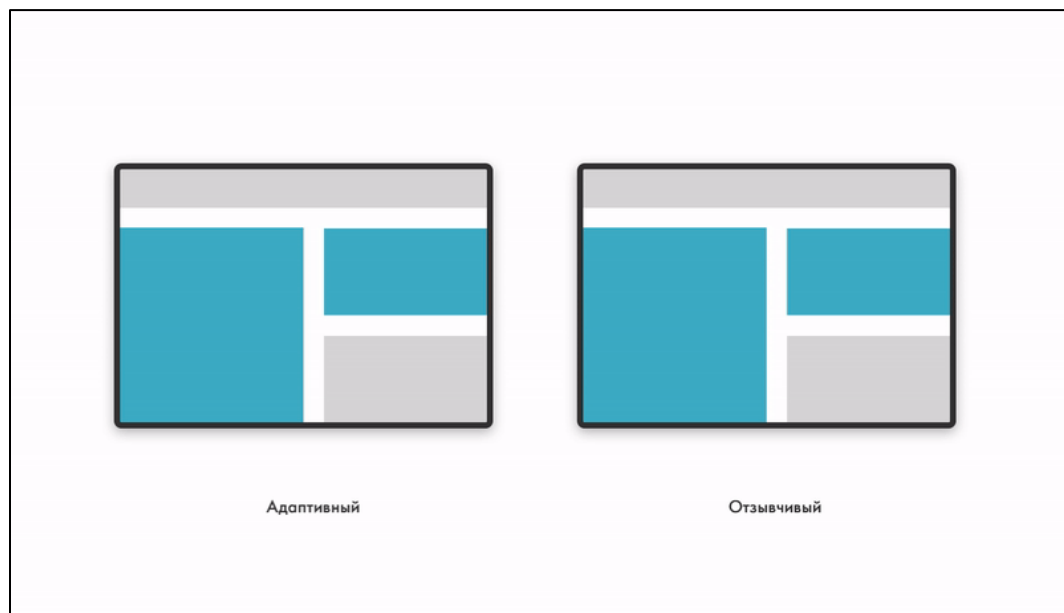
В то время как **отзывчивый дизайн** основан на плавном изменении сайта в соответствии с доступным ему пространством, **адаптивный дизайн** имеет несколько фиксированных размеров макета. Например, создаются версии для ПК, планшета и телефона. На всех планшетах будет одна версия сайта, на всех телефонах — другая, на всех ПК — третья, но без учёта размеров экранов отдельных устройств в каждой группе.

В адаптивном дизайне обычно разрабатывают шесть дизайнов для шести наиболее распространенных значений ширины экрана: 320, 480, 760, 960, 1200 и 1600 пикселей. Когда сайт анализирует доступное пространство, он показывает пользователю версию, наиболее подходящую для размера его экрана.



Если вы открываете браузер, сайт выбирает лучший макет для этого экрана, но изменение размера браузера не повлияет на изменение дизайна, в отличие от отзывчивого дизайна. Макет, отображаемый на мобильной версии сайта с использованием адаптивного дизайна, может отличаться от версии для ПК, потому что дизайнер мог счесть, что изменение расположения и размеров элементов — более удобный вариант для пользователя смартфона.

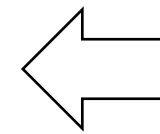
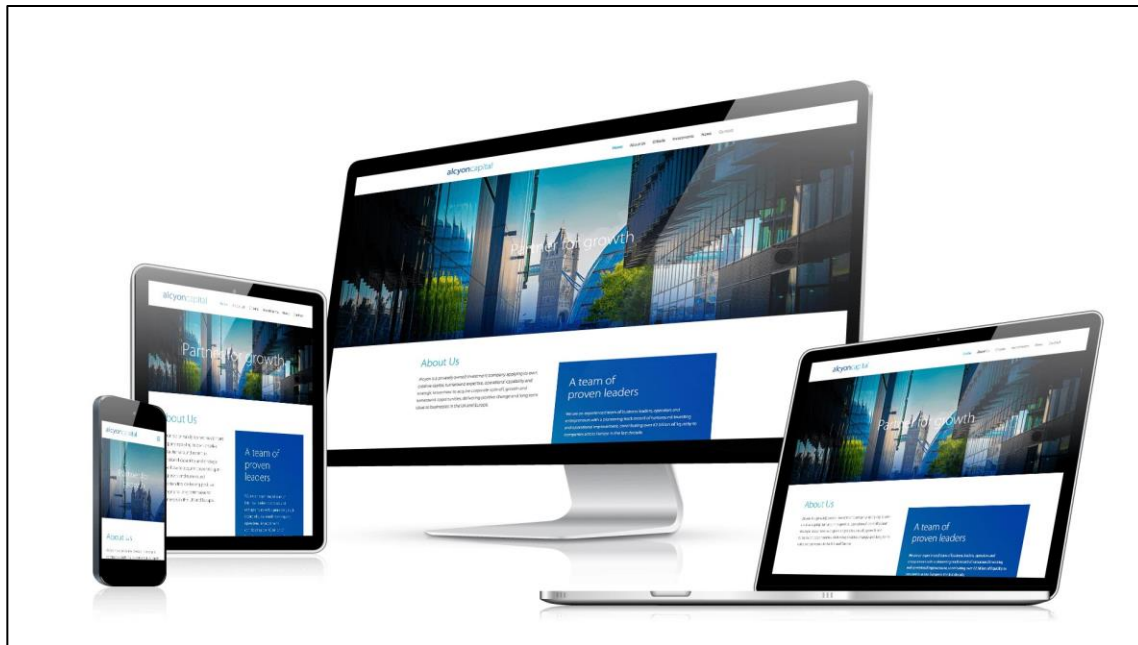
Адаптивный или отзывчивый дизайн



Адаптивный или отзывчивый дизайн

Отзывчивый дизайн нарисовать проще. Его разработка занимает меньше времени. Однако он обеспечивает меньший контроль над вашим дизайном для каждого размера экрана. На данный момент это, безусловно, предпочтительный метод для создания новых сайтов.

Адаптивный дизайн предполагает создание дизайна для использования на всех экранах. Дизайнеры обычно начинают его разработку со среднего разрешения, а затем делают корректировки под другие разрешения экранов.



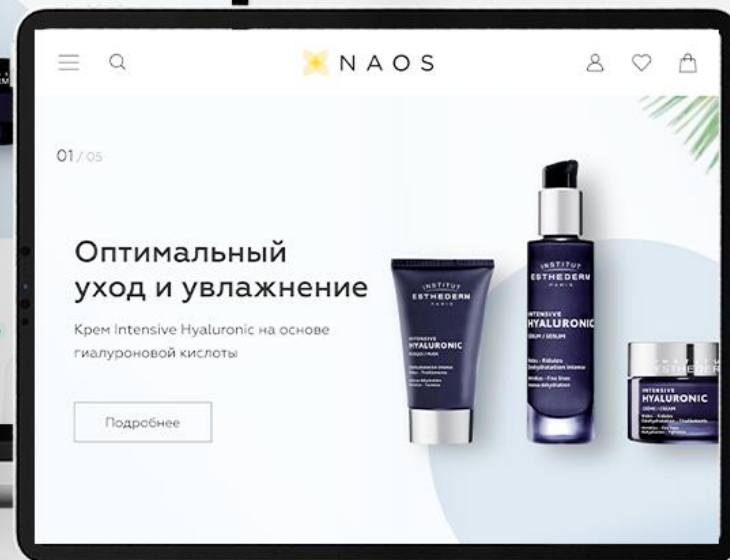
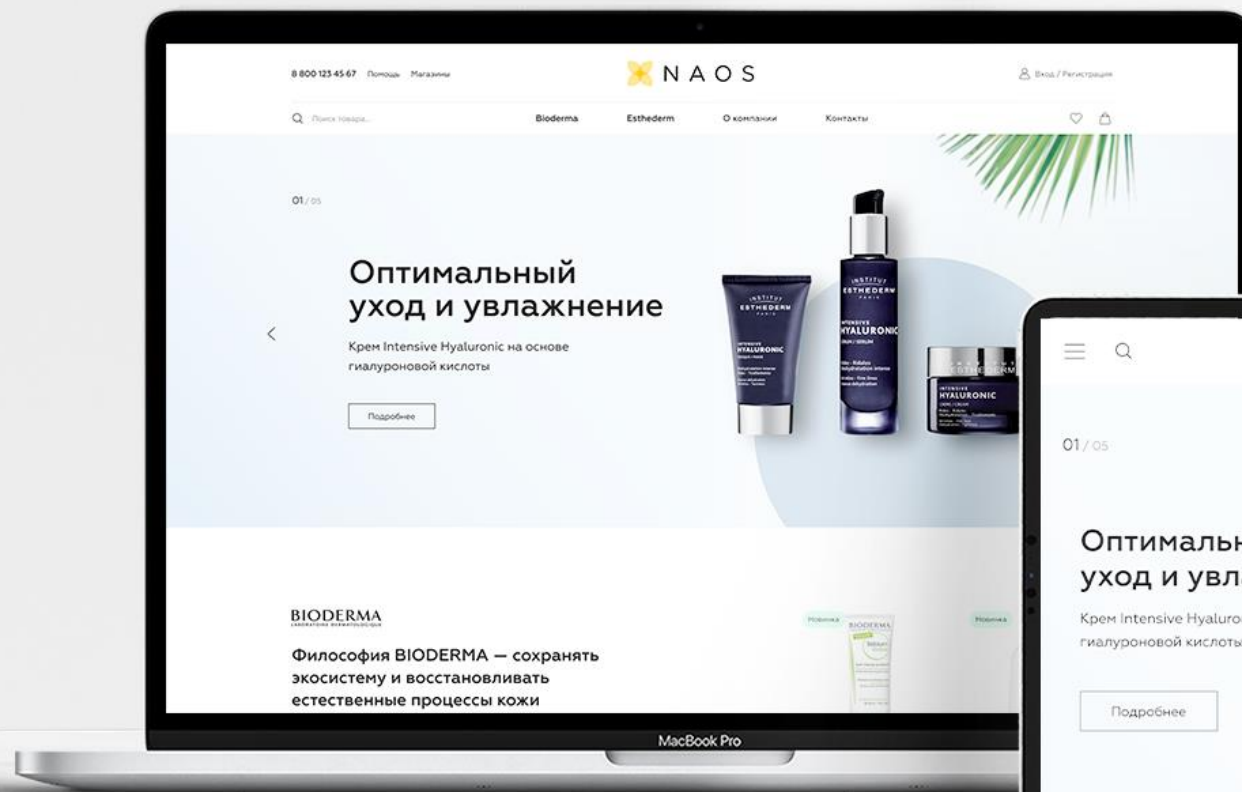
Пример отзывчивого дизайна. На всех экранах видим один и тот же дизайн, модифицированный под соответствующие габариты

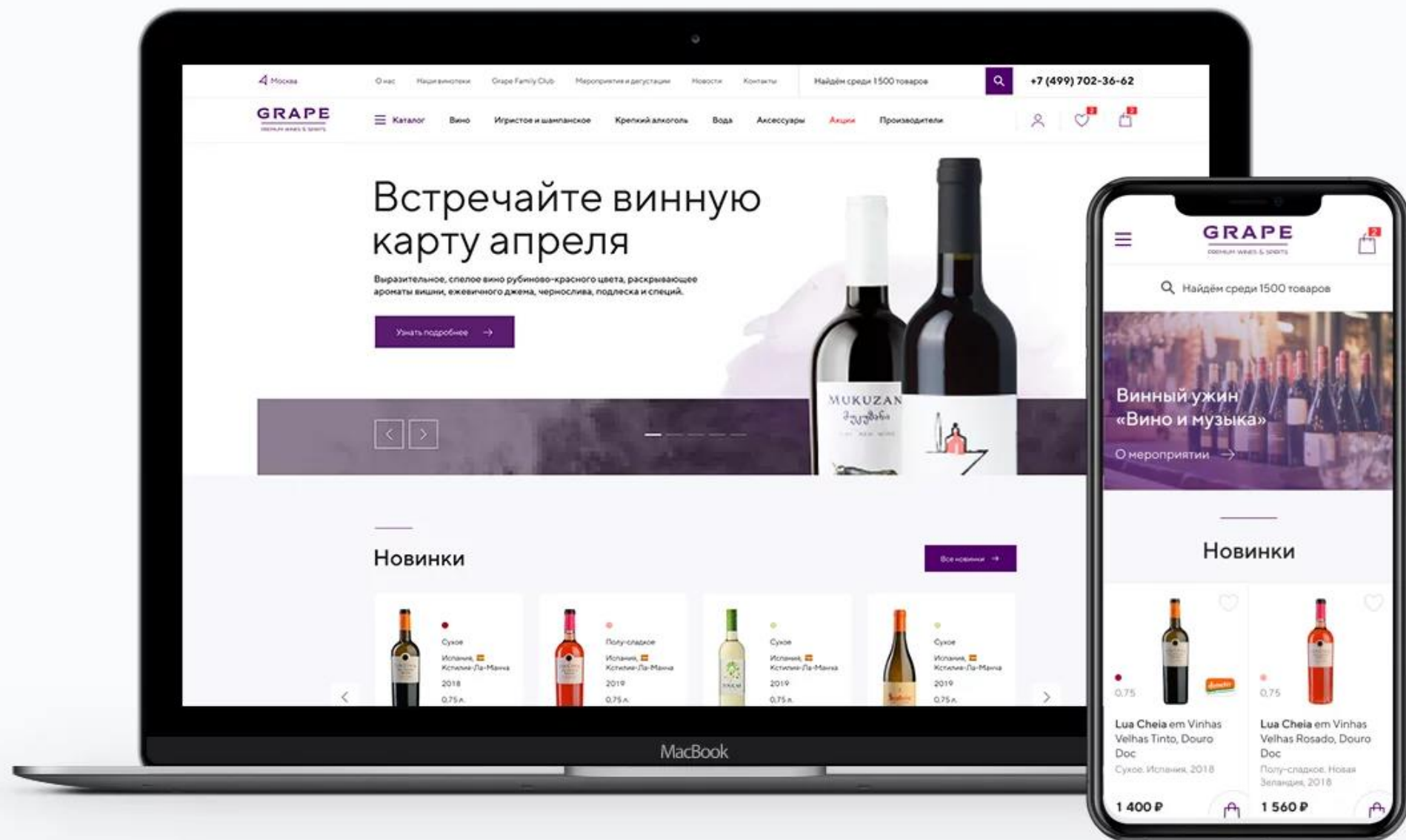
Адаптивный или отзывчивый дизайн

Адаптивный дизайн теоретически сможет создать лучший пользовательский интерфейс, потому что в отличие от **отзывчивого дизайна**, он предлагает индивидуальные решения под каждый экран (на телефоне можно увеличить кнопки, убрать лишние элементы, добавить пространства, а на ПК — разместить огромный баннер и т.п.). Другими словами, такое решение адаптируется к ситуативным потребностям и возможностям пользователя.

Однако здесь может получиться такая ситуация, что пользователи планшетов или ноутбуков могут остаться «в подвешенном состоянии». Зачастую дизайнеры рисуют версии для ПК и смартфонов. Что-то между остаётся за рамками, поэтому пользователи планшетов и ноутбуков, как правило, получают маленькую мобильную версию посередине своего окна браузера. Чтобы такого не случилось с вашим дизайном, важно предлагать ссылку, позволяющую пользователю переключаться между версиями.

Отзывчивый дизайн радует пользователей единообразием и целостностью: посетитель видит знакомый веб-дизайн, на экране любого устройства. Минус в таком подходе — это не в лучшую сторону влияет на скорость загрузки страницы.

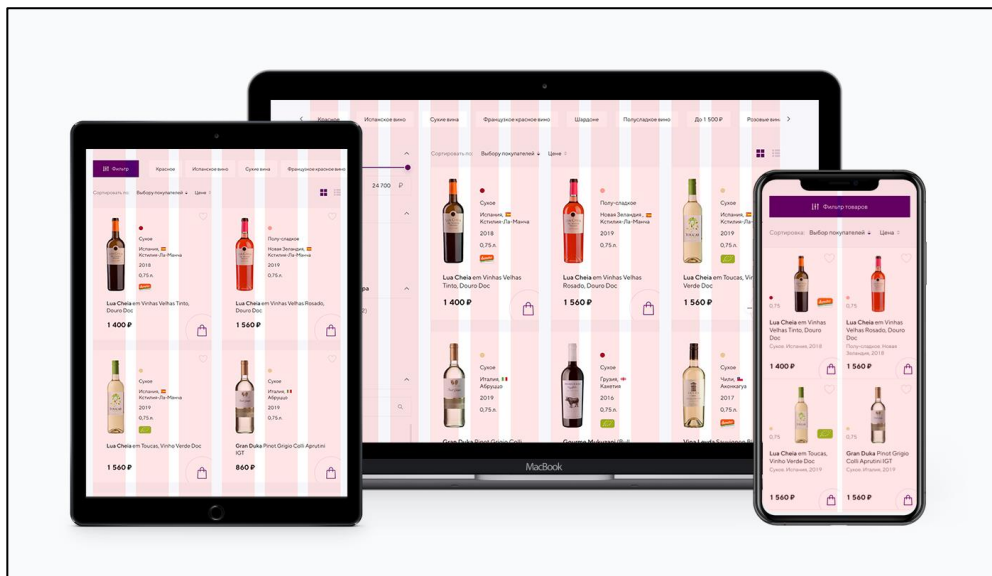




Сетка в Bootstrap

Система сеток в Bootstrap используется для создания раскладок, а в частности для создания их адаптивности. Понимание того, как это работает — является жизненно важным моментом при работе с Bootstrap. Сетка сделана из группирования строк и колонок внутри одного или нескольких контейнеров.

Сетки на Bootstrap могут использоваться отдельно, без Bootstrap JavaScript и других CSS компонентов. Вам надо только скачать и сослаться на "**bootstrap-grid.css**", который включает в себя flexbox классы и классы для сетки.



Сетка в Bootstrap

Вот самый простой пример применения сетки

```
16 <div class="container my-5">
17   <div class="row">
18     <div class="col">
19       <h4>Шаблон с одной колонкой</h4>
20 >   <p>...
32   </p>
33   </div>
34 </div>
35 </div>
```

HTML

Шаблон с одной колонкой

Here is my page content. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia cor magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur?

Браузер

Сетка в Bootstrap

А вот уже две колонки

```
36 <div class="container my-5">
37   <div class="row">
38     <div class="col">
39       <h4>Левая колонка</h4>
40 >   <p>...
48   </p>
49   </div>
50   <div class="col">
51     <h4>Правая колонка</h4>
52 >   <p>...
60   </p>
61   </div>
62   </div><!--/row-->
63 </div><!--container-->
```

HTML

Левая колонка

Here is my page content. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia cor magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor.

Правая колонка

Here is my page content. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia cor magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor.

Браузер

Сетка в Bootstrap

И вот на три колонки

```
68 <div class="container my-5">
69   <div class="row">
70     <div class="col">
71       <h4>Левая колонка</h4>
72 >     <p>...
77   </p>
78   </div>
79   <div class="col">
80     <h4>Центральная колонка</h4>
81 >     <p>...
86   </p>
87   </div>
88   <div class="col">
89     <h4>Правая колонка</h4>
90 >     <p>...
95   </p>
96   </div>
97   </div><!--/row-->
98 </div><!--container-->
```

HTML

Левая колонка

Here is my page content. Sed ut
perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque
laudantium, totam rem aperiam, eaque
ipsa quae ab illo inventore veritatis et
quasi architecto beatae vitae.

Центральная колонка

Here is my page content. Sed ut
perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque
laudantium, totam rem aperiam, eaque
ipsa quae ab illo inventore veritatis et
quasi architecto beatae vitae.

Правая колонка

Here is my page content. Sed ut
perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque
laudantium, totam rem aperiam, eaque
ipsa quae ab illo inventore veritatis et
quasi architecto beatae vitae.

Браузер

Правила работы с сеткой Bootstrap

1. Колонки должны быть прямыми потомками **row**
2. **row** используются только для того, чтобы включать в себя колонки и не для ничего больше.
3. **row** должны быть помещены внутри контейнера

Эти правила **ОЧЕНЬ ВАЖНЫ**. Строки и колонки всегда работают вместе и вам надо смотреть за тем, что один не оставался без другого.

Контейнер

В простом примере вы видели, что мы использовали **.container**, чтобы обернуть **.row**. Контейнер это ключевой элемент сетки в Bootstrap.

```
16 <div class="container">
17   <div class="row">
18     <div class="col">
19       <h4>Шаблон с одной колонкой</h4>
20 >   <p>...
32     </p>
33   </div>
34 </div>
35 </div>
```

Контейнер может использоваться для хранения любых элементов и самого контента. Он используется не только для строк и колонок сетки.

Контейнер

С первого взгляда, контейнер может показаться пустяковым и не сильно уж нужным, но он очень важен для контроля ширины шаблона. Контейнер также используется для равномерного выравнивания граней шаблона внутри вьюпорта браузера. Контейнер используется, чтобы противодействовать отрицательным внешним отступам **row**, о чем мы узнаем немного позже.

У Bootstrap есть 2 типа контейнера. В примерах мы использовали **.container**, но также есть и полноэкранный **.container-fluid**. Вы можете использовать любой из них.

1 — Контейнер с фиксированной шириной, для центровки контейнера по середине шаблона.

```
<div class="container"></div>
```

2 — Контейнер с шириной во весь экран.

```
<div class="container-fluid"></div>
```

Контейнер

.container масштабируется адаптивно по ширине экрана, так что в конце концов он может стать шириной на весь экран, как и **.container-fluid**, но на маленьких устройствах.

Помните, что контейнер может использоваться для любого контента, а не только строк и колонок сетки. Но! Если вы используете последние два элемента, то строки сетки должны быть размещены внутри контейнера.



HTML

```
32 <div class="container-fluid">
33   <div class="row">
34     <div class="col-md-12">.container-fluid</div>
35   </div>
36 </div>
37
38 <div class="container">
39   <div class="row">
40     <div class="col-md-12">.container</div>
41   </div>
42 </div>
```

Браузер

Контейнер

Применяя сетку, более одной строки может быть помещено внутри контейнера. Вы можете иметь их сколько угодно в самом контейнере и вы также можете иметь сколько хотите контейнеров на странице. Все зависит от того, какой шаблон вы пытаетесь сверстать, но пока что не сильно заморачивайтесь по этому поводу.

У строк (**rows**) есть отрицательные левые/правые внешние отступы в **-15px**. Внутренний отступ контейнера в **15px** используется для пресечения срабатывания отрицательных внешних отступов в строке контейнера. Это делается для равномерного выравнивания по граням в шаблоне. Если вы не поместите строку (**row**) в контейнер, то она будет выходить за пределы своего контейнера, вызывая нежелательные горизонтальные прокрутки.

Строки (Rows) и Колонки (Columns)

ИМХО: Сейчас, я бы хотел, чтобы **.row** в Bootstrap не были на самом деле именованы как **строки (row)**. Название **строка**, зачастую запутывающее и скрывает настоящее предназначение **.row**.

Когда вы думаете о **строке**, вы возможно думаете о горизонтальной линии, что вполне ОК, НО, лучше думать о **строке**, как о **родителе для колонок**.

Думайте о строках, как о **группе колонок (Columns)**.

Это потому, что колонки внутри **.row** не всегда располагаются горизонтально вдоль вьюпорта. Иногда нам надо, чтобы колонки в шаблоне были горизотальны, а иногда нам надо, чтобы они располагались вертикально. Концепция горизонтального vs. вертикального шаблона является сущностью адаптивного дизайна. Единственным предназначением **строки**, является содержание одной или более “колонок”.

Не вставляйте контент прямо в **строку**!

Строки (Rows) и Колонки (Columns)

Так делать нельзя

```
18 <div class="row">
19 |   This is very bad, wrong way, no bueno!!
20 </div>
21 <div class="row">
22 |   <p>This is also very bad, the wrong way!!</p>
23 </div>
24 <div class="row">
25 |   <h2>No headings either! This is the wrong way!!</h2>
26 </div>
```

“колонки” и только колонки, размещаются внутри “строк”, а контент размещается уже внутри “колонок”.

```
26 <div class="row">
27 |   <div class="col">Happy :-> This is the right way.</div>
28 </div>
```

Строки (Rows) и Колонки (Columns)

Есть разные типы колонок и разные способы использования их в шаблоне. Они как магия.

- Создают горизонтальные разделители по вьюпорту.
- Могут иметь разную ширину.
- Раскладку горизонтально слева направо, вертикально сверху вниз.
- Могут изменять позицию (порядок) относительно родственных элементов в той же строке.
- Имеют ту же высоту, что и другие родственные элементы в той же строке.
- Могут “расти” или “урезаться” по ширине.
- Могут автоматически врапиться или вертикально “слепляться” при необходимости или при нужной ширине экрана.
- Могут содержать больше строк и колонок при вложении.



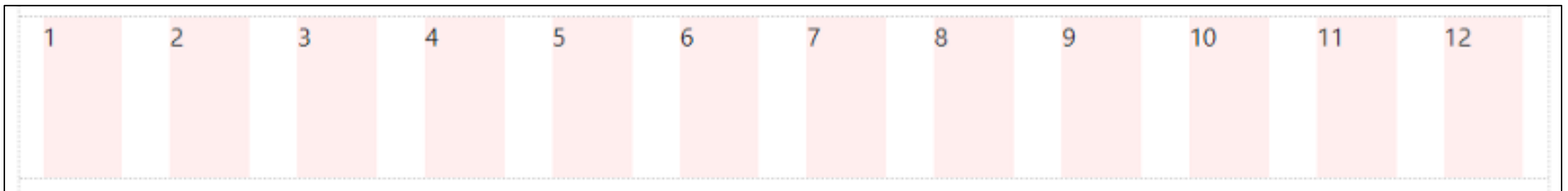
Строки (Rows) и Колонки (Columns)

Колонки должны быть прямыми потомками строк.

Колонки создают горизонтальные деления по вьюпорту. Пространство между колонками называется **"gutter"**.

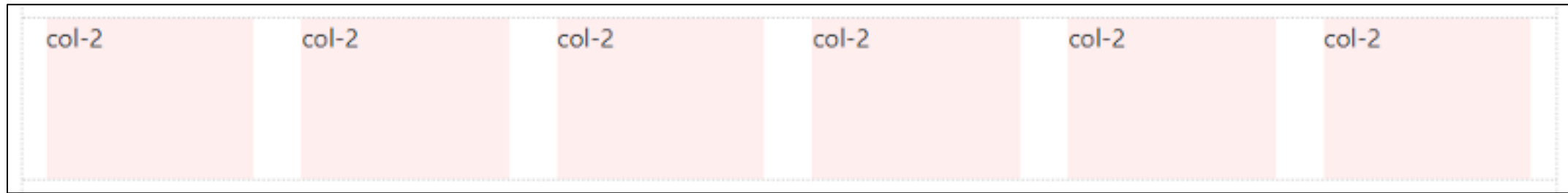


Классическая сетка Bootstrap имеет 12 колонок



Строки (Rows) и Колонки (Columns)

Таким образом, колонки могут быть равномерно разделены на 12 частей. Вот пример, 6 колонок ($12/6=2$)



И дальше уже дело простой математики...

4 columns ($12/4 = 3$)

col-3

col-3

col-3

col-3

3 columns ($12/3 = 4$)

col-4

col-4

col-4

2 columns ($12/2 = 6$)

col-6

col-6

1 column ($12/1 = 12$)

col-12

Строки (Rows) и Колонки (Columns)

Колонки могут быть разделены с использованием любой части из 12 элементов.

col-4	col-8
col-9	col-3
col-5	col-7

col-4	col-6	col-2
col-3	col-6	col-3
col-6	col-3	col-3

Адаптивный дизайн с Bootstrap

Ширина колонки может изменяться в зависимости от ширины экрана. Это и есть работа с адаптивным дизайном.

В Bootstrap есть 5 адаптивных рядов (или брейкпоинтов).

Адаптивные брейкпоинты, основаны на ширине экрана:

- **(xs)** — ширина экрана $< 576\text{px}$. Это стандарт.
- **sm** — ширина экрана $\geq 576\text{px}$
- **md** — ширина экрана $\geq 768\text{px}$
- **lg** — ширина экрана $\geq 992\text{px}$
- **xl** — ширина экрана $\geq 1200\text{px}$

xs это дефолтное прерывание, **-xs** инфикс, который использовался в Bootstrap 3, больше не используется в Bootstrap. Так что вместо **col-xs-6**, просто **col-6**.

Адаптивный дизайн с Bootstrap

Bootstrap использует медиа запросы из CSS, что установить адаптивные точки прерываний. Они дают вам возможность контролировать поведение колонок при разных размерах экрана.

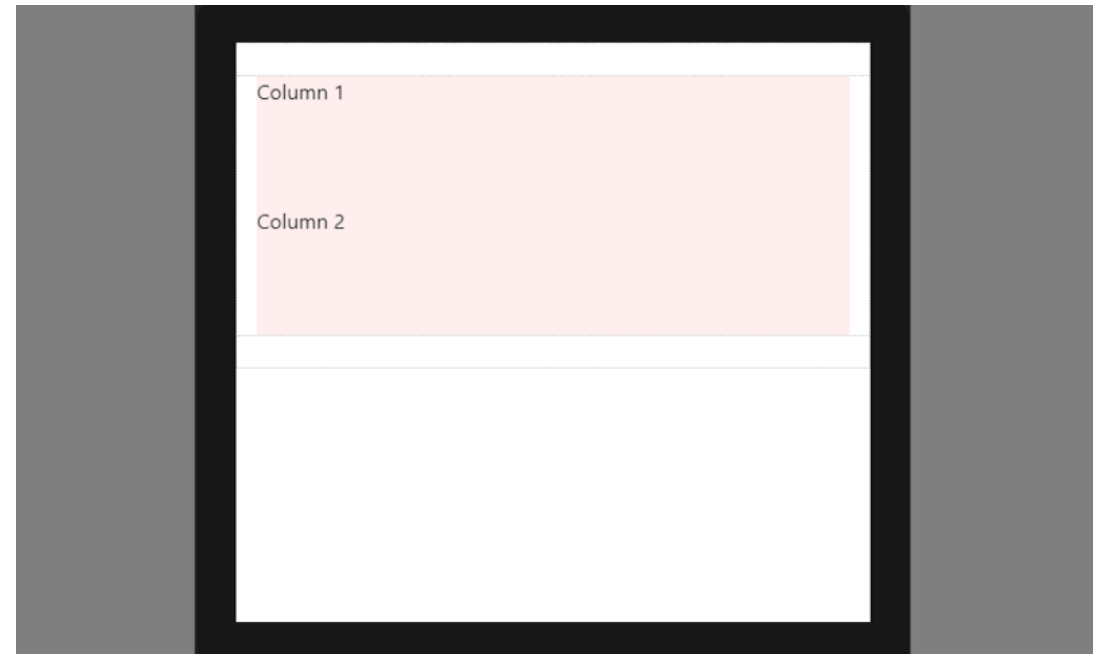
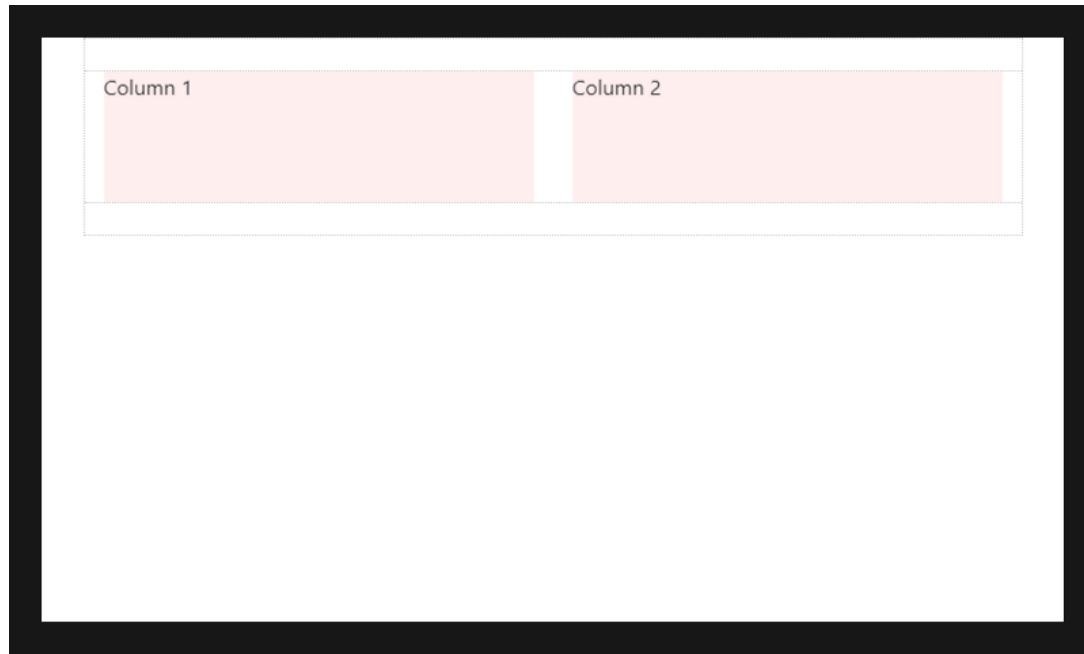
Для примера: вот 2 колонки, каждая шириной 50%

col-sm-6 означает использование 6 колонок из 12, то есть 50% ширины на типичных маленьких размерах экранов. Которые больше или равны **768px**

```
17 <div class="container">
18   <div class="row">
19     <div class="col-sm-6">Колонка 1</div>
20     <div class="col-sm-6">Колонка 2</div>
21   </div>
22 </div>
```

Адаптивный дизайн с Bootstrap

На экранах меньше, чем **768px**, 2 колонки станут шириной 100% и встанут вертикально



Это происходит, потому что (xs) по дефолту или подразумевает брейкпоинт. Так как мы не указываем дефолтную ширину колонки, 50% ширина была применена только на **768px** и шире для **sm** брейкпоинта.

Mobile-First

Так как (xs) это дефолтный брейкпоинт, то подразумевает col-12.

Следовательно:

```
<div class="col-12 col-sm-6">Колонка</div>
```

Работает так же, как и:

```
<div class="col-sm-6">Колонка</div>
```

xs(default) > переписывает sm > переписывает md > переписывает lg > переписывает xl

Или в обратном порядке... xl > переписывает lg > переписывает md > переписывает sm > переписывает (xs)

Следовательно, col-sm-6 реально означает 50% ширины на мелких и выше дисплеях. Для одинаковой ширины колонок на всех брейкпоинтах, просто выставите ширину для самого маленького ряда как хотите:

```
<div class="col-lg-3 col-md-3 col-sm-3">..</div>
```

тоже самое, что и:

```
<div class="col-sm-3">..</div>
```

Mobile-First

Для разной ширины колонки на разных уровнях, используйте подходящие брейкпоинты для перезаписи мелких прерываний. Для примера, 3 колоночная ширина на **sm** и 4 колоночная ширина на **md** и выше.

```
<div class="col-sm-3 col-md-4">..
```

Auto-layout колонки в Bootstrap 4 также работают адаптивно. Auto-layout колонки идеальны для любых сценариев шаблонов, где необходима равная ширина колонок. Но не забудьте, что 12-ти колоночные юниты могут быть смешаны при необходимости.

Auto-layout сетка

3 равные колонки. **'col'** остаются горизонтальным на всей широте и не встают вертикально, так как **xs** прерывание дефолтно.

```
25 <div class="container">
26   <div class="row">
27     <div class="col">1</div>
28     <div class="col">2</div>
29     <div class="col">3</div>
30   </div>
31 </div>
```

3 равные колонки, адаптивные. В этом примере, **'col'** остаются горизонтальным до прерывания **sm** на **576px**, а затем они становятся вертикальными. Помните, что вы можете заменять **sm** на каком угодно брейкпоинте (md,lg,xl), если нужно:

```
33 <div class="container">
34   <div class="row">
35     <div class="col-sm">1</div>
36     <div class="col-sm">2</div>
37     <div class="col-sm">3</div>
38   </div>
39 </div>
```

Auto-layout сетка

2 колонки, левый сайдбар.

А вот пример комбинирования классически определённой ширины колонок с колонками auto-layout. Правая колонка будет автоматически расти, чтобы занять ширину, так как мы используем auto-layout **.col**. Сайдбар будет 16.6% ширины на больших экранах и затем встанет над контентом при **sm** брейкпоинте с **576px**:

```
41 <div class="container">
42   <div class="row">
43     <div class="col-sm-2">sidebar</div>
44     <div class="col">main content</div>
45   </div>
46 </div>
```

Auto-layout сетка

3 колонки, правый сайдбар

В этом примере есть левый сайдбар, центральная область контента и правый сайдбар, который сокращается по ширине, чтобы подстроиться под свой контент.

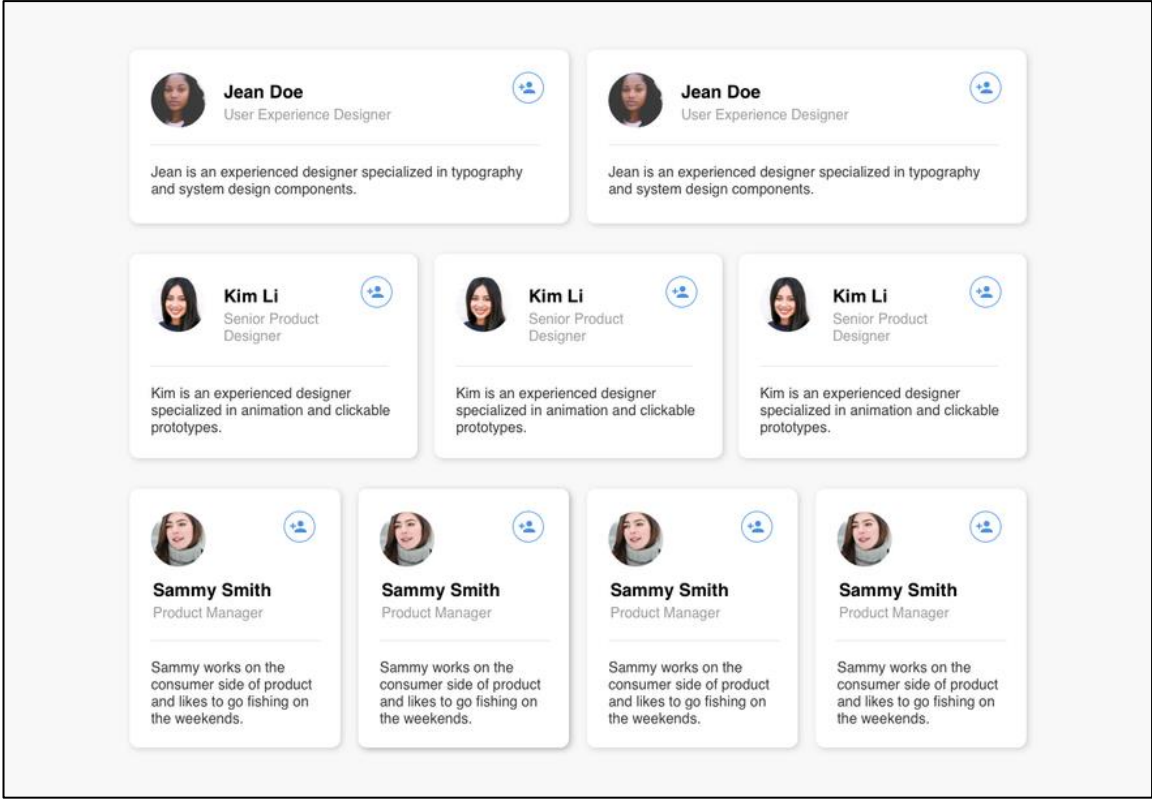
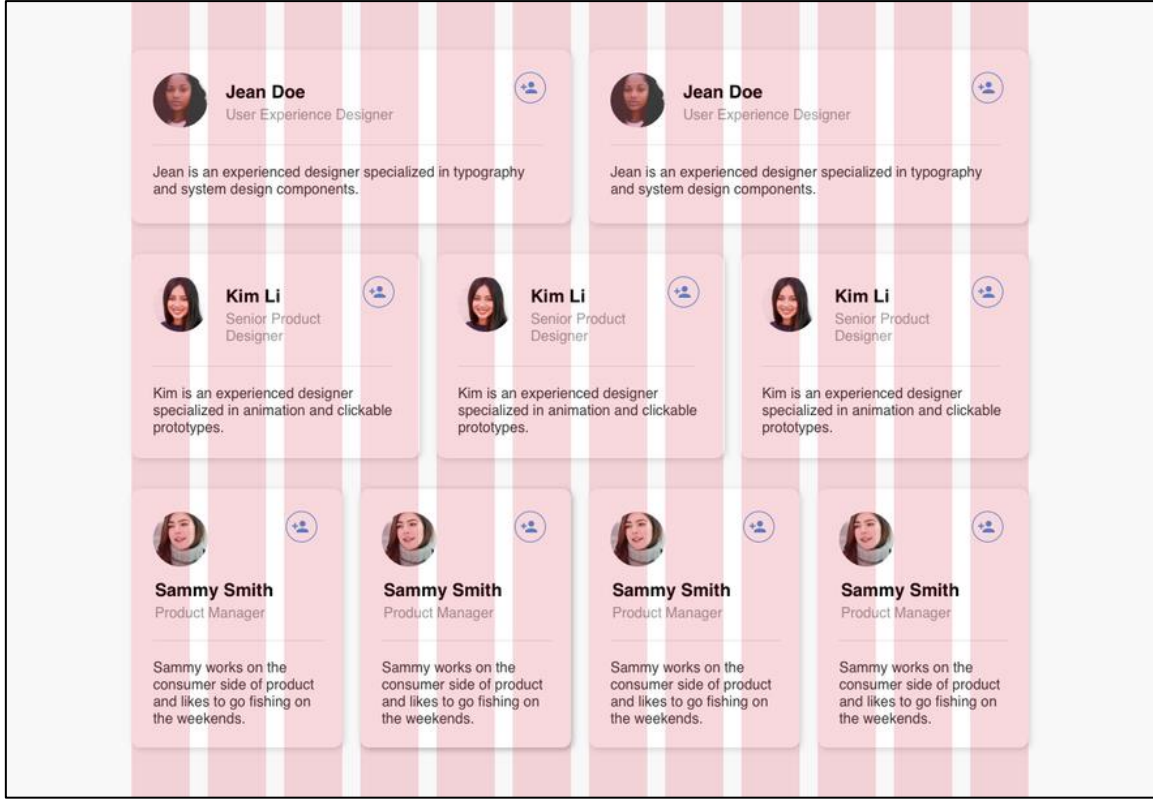
```
48 <div class="container">
49   <div class="row">
50     <div class="col-sm-2">left</div>
51     <div class="col">main content</div>
52     <div class="col-auto">right</div>
53   </div>
54 </div>
```


Ключевые моменты адаптивного дизайна используемые в сетке Bootstrap

Колонки встанут вертикально и станут шириной во весь экран на устройствах с маленьким разрешением, если вы не используете `col-*` класс в HTML разметке. Используйте `col-*` для предотвращения такого вертикального выстраивания.

Классы сеток поменьше, также применяются на больших экранах, пока не переписутся конкретно под ширину большего экрана. Следовательно, `<div class="col-md-6"></div>` в сущности тоже самое, что и `<div class="col-md-6 col-lg-6"></div>`. Следовательно, вам только надо использовать класс для самых маленьких разрешений, которые вам нужно поддерживать.

Строки (row) —имеют **display: flex** и следовательно колонки имеют равную высоту в одном и том же ряду. Используйте **auto-margin** или Flexbox **align-item** и **justify-content** для горизонтального или вертикального выравнивания.



How to stir fry chicken

1

Cut the chicken

Cut your pieces into bite size pieces to prepare for the stir fry, and be extra careful on how you handle the knife.

2

Stir until white

Add your chicken to the lightly oiled pan, and continue to stir on medium to high heat until the surfaces are all white

3

Add seasoning

Add some seasoning like salt and pepper and soy sauce, and mix well until the insides of the chicken have been cooked through

How to stir fry chicken

1

Cut the chicken

Cut your pieces into bite size pieces to prepare for the stir fry, and be extra careful on how you handle the knife.

2

Stir until white

Add your chicken to the lightly oiled pan, and continue to stir on medium to high heat until the surfaces are all white

3

Add seasoning

Add some seasoning like salt and pepper and soy sauce, and mix well until the insides of the chicken have been cooked through



Jean Doe

User Experience Designer

Jean is an experienced designer specialized in typography and system design components.



Jean Doe

User Experience Designer

Jean is an experienced designer specialized in typography and system design components.



Jean Doe

User Experience Designer

Jean is an experienced designer specialized in typography and system design components.

Welcome to Boop + Booper



About Us

Lorem ipsum dolor sit amet, id velit. Dolor sit in convallis velit rutrum, euismod at lorem odio venenatis error, euismod in nibh sociis, sodales eros laoreet mauris tincidunt morbi, et metus non in lacinia. Vestibulum vestibulum at vehicula ac ante, in magna donec wisi necessitatibus pulvinar, libero blandit sem justo nunc

Our Mission

Lorem ipsum dolor sit amet, id velit. Dolor sit in convallis velit rutrum, euismod at lorem odio venenatis error, euismod in nibh sociis, sodales eros laoreet mauris tincidunt morbi, et metus non in lacinia. Vestibulum vestibulum at vehicula ac ante, in magna donec wisi necessitatibus pulvinar, libero blandit sem justo nunc



✓ Yes

Welcome to Boop + Booper



About Us

Lorem ipsum dolor sit amet, id velit. Dolor sit in convallis velit rutrum, euismod at lorem odio venenatis error, euismod in nibh sociis, sodales eros laoreet mauris tincidunt morbi, et metus non in lacinia. Vestibulum vestibulum at vehicula ac ante, in magna donec wisi necessitatibus pulvinar, libero blandit sem justo nunc

Our Mission

Lorem ipsum dolor sit amet, id velit. Dolor sit in convallis velit rutrum, euismod at lorem odio venenatis error, euismod in nibh sociis, sodales eros laoreet mauris tincidunt morbi, et metus non in lacinia. Vestibulum vestibulum at vehicula ac ante, in magna donec wisi necessitatibus pulvinar, libero blandit sem justo nunc



✓ Yes



Hello, we are Paperpillar

A team of passionate
designers and developers from
Yogyakarta



SERVICES

What We Do



UI/UX

Landing Pages, User Flow,
Wireframing, Prototyping,
Mobile App Design, Web App



Icon/Illustration

Character Design, Icon Set,
Illustration Guide, Illustration
Set



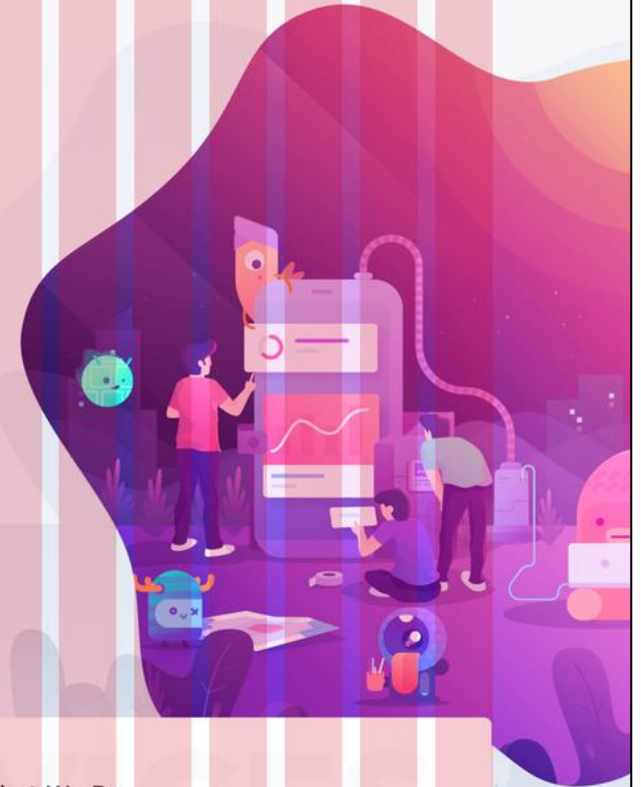
Branding

Visual Identity, Stationary Kit,
Marketing Materials



Hello, we are Paperpillar

A team of passionate
designers and developers from
Yogyakarta



SERVICES

What We Do



UI/UX

Landing Pages, User Flow,
Wireframing, Prototyping,
Mobile App Design, Web App



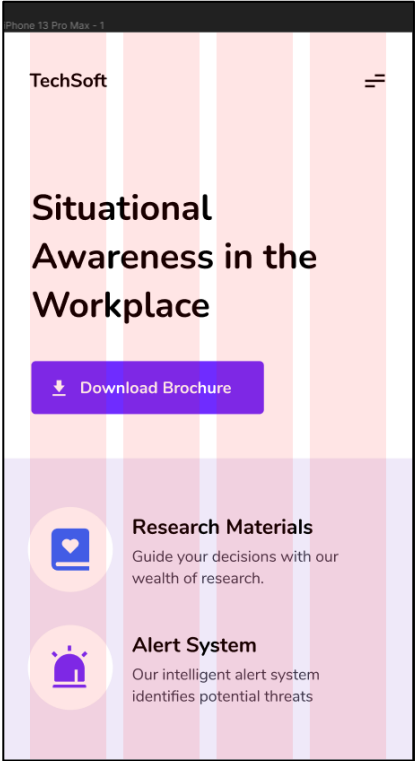
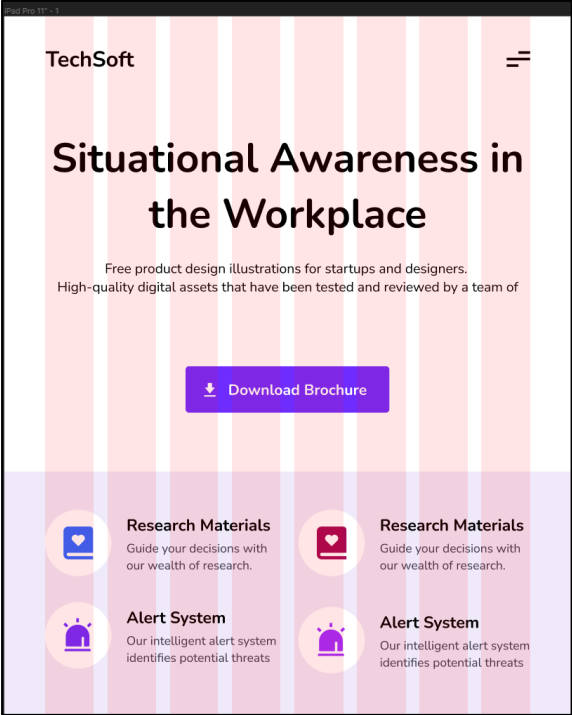
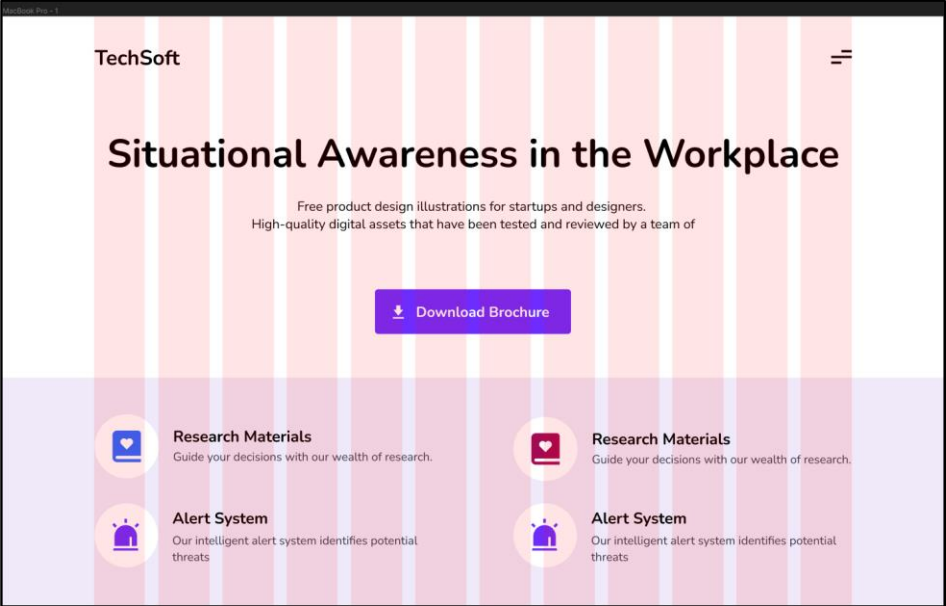
Icon/Illustration

Character Design, Icon Set,
Illustration Guide, Illustration
Set



Branding

Visual Identity, Stationary Kit,
Marketing Materials



Спасибо за внимание

Форма обратной связи по модулю: <https://forms.gle/qKmwLEFAFSe8heNo6>