

# Адаптивный дизайн

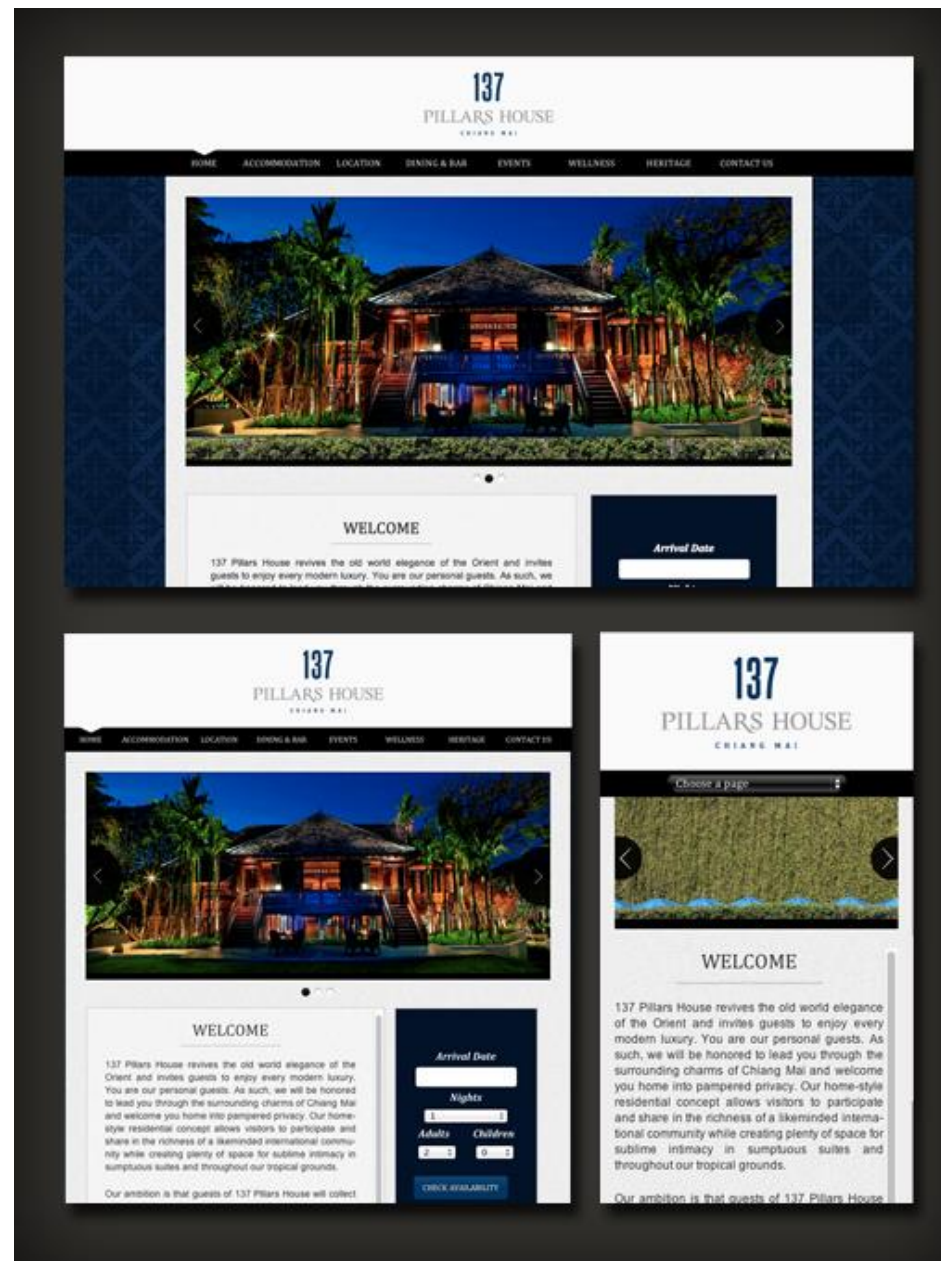
**HTML**



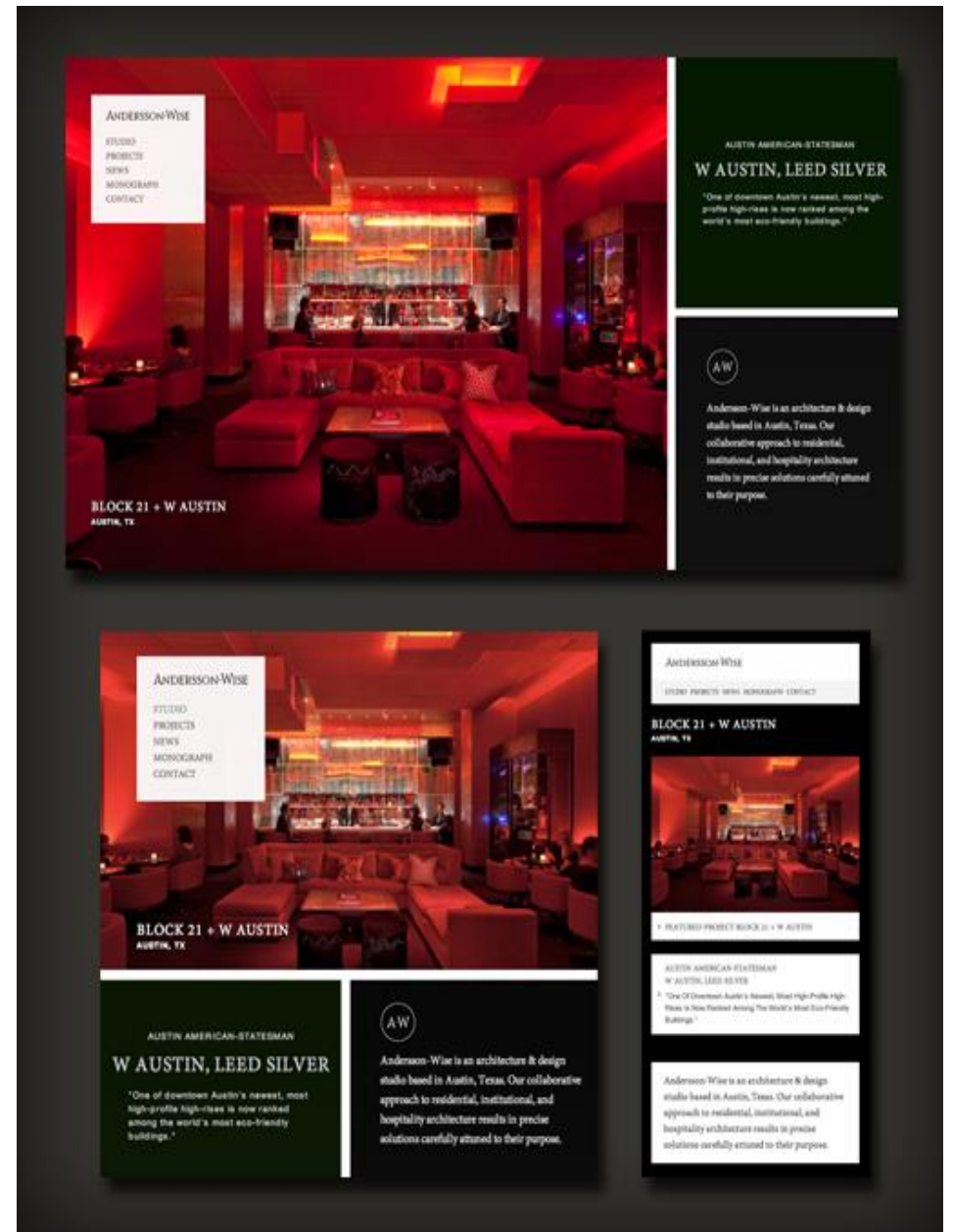
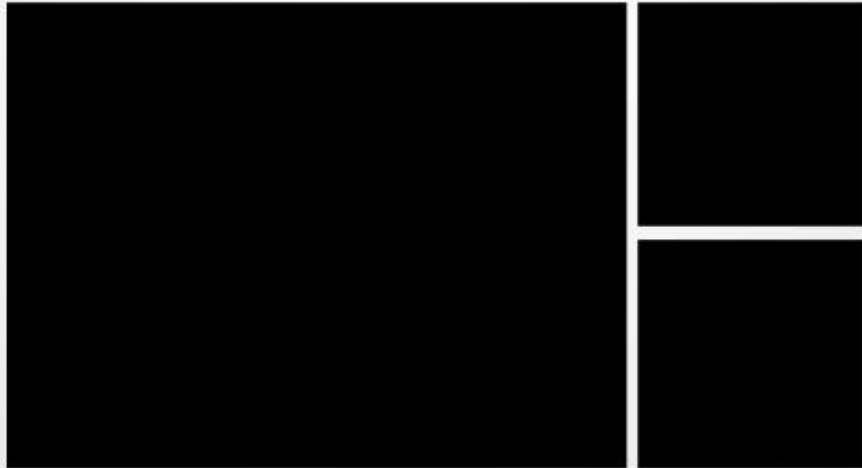
**CSS**



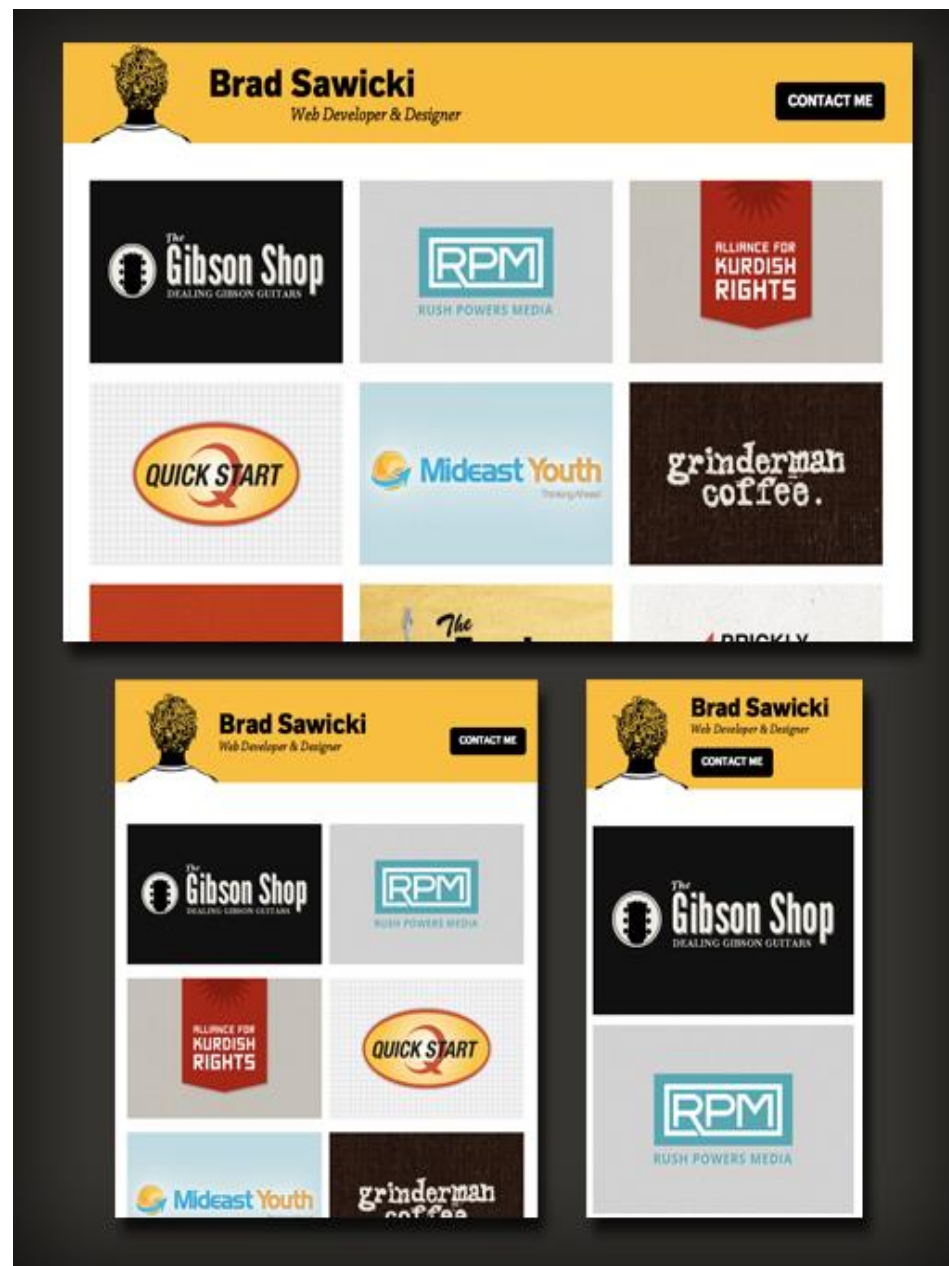
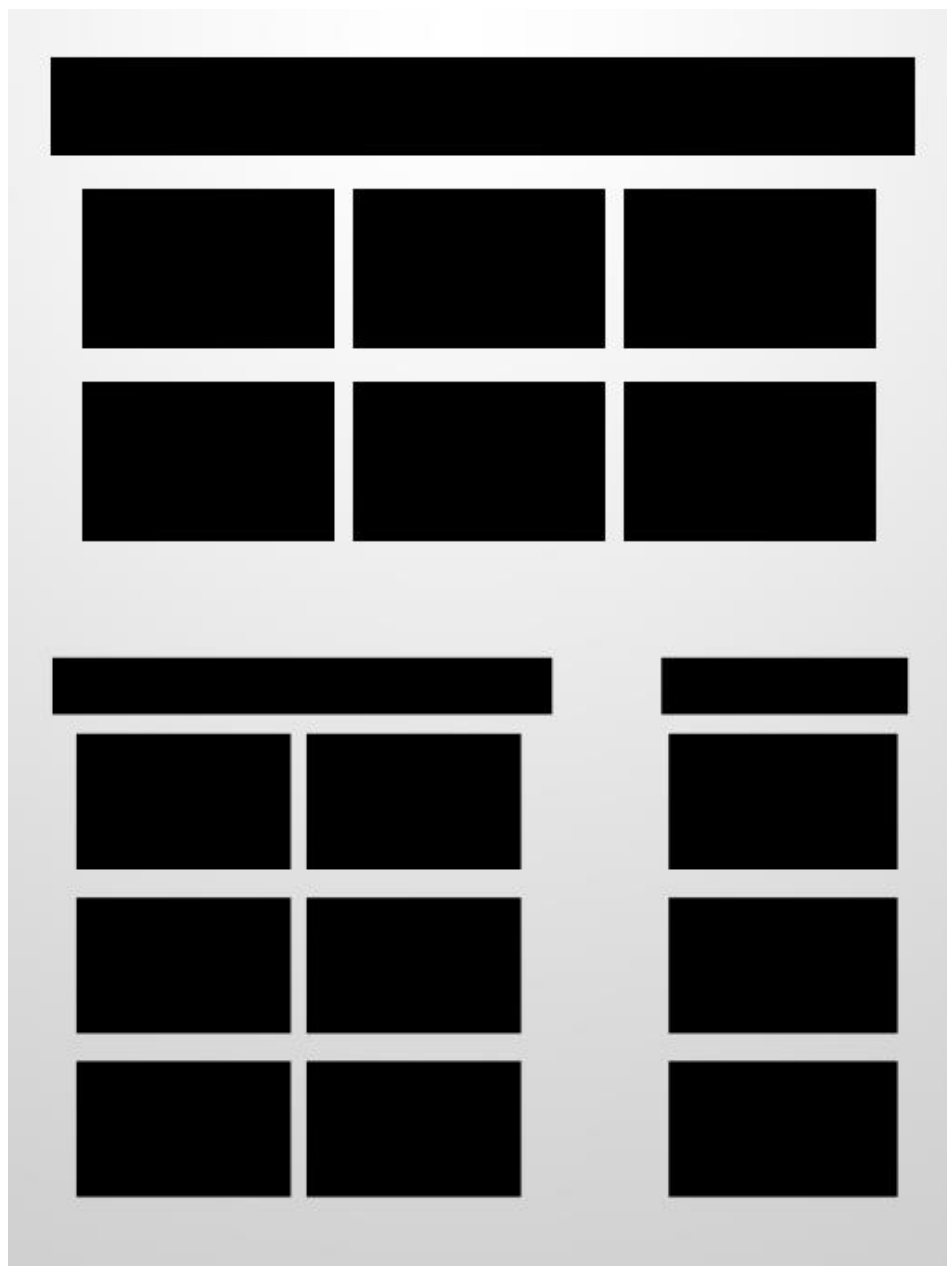
# Как проявляется адаптивность



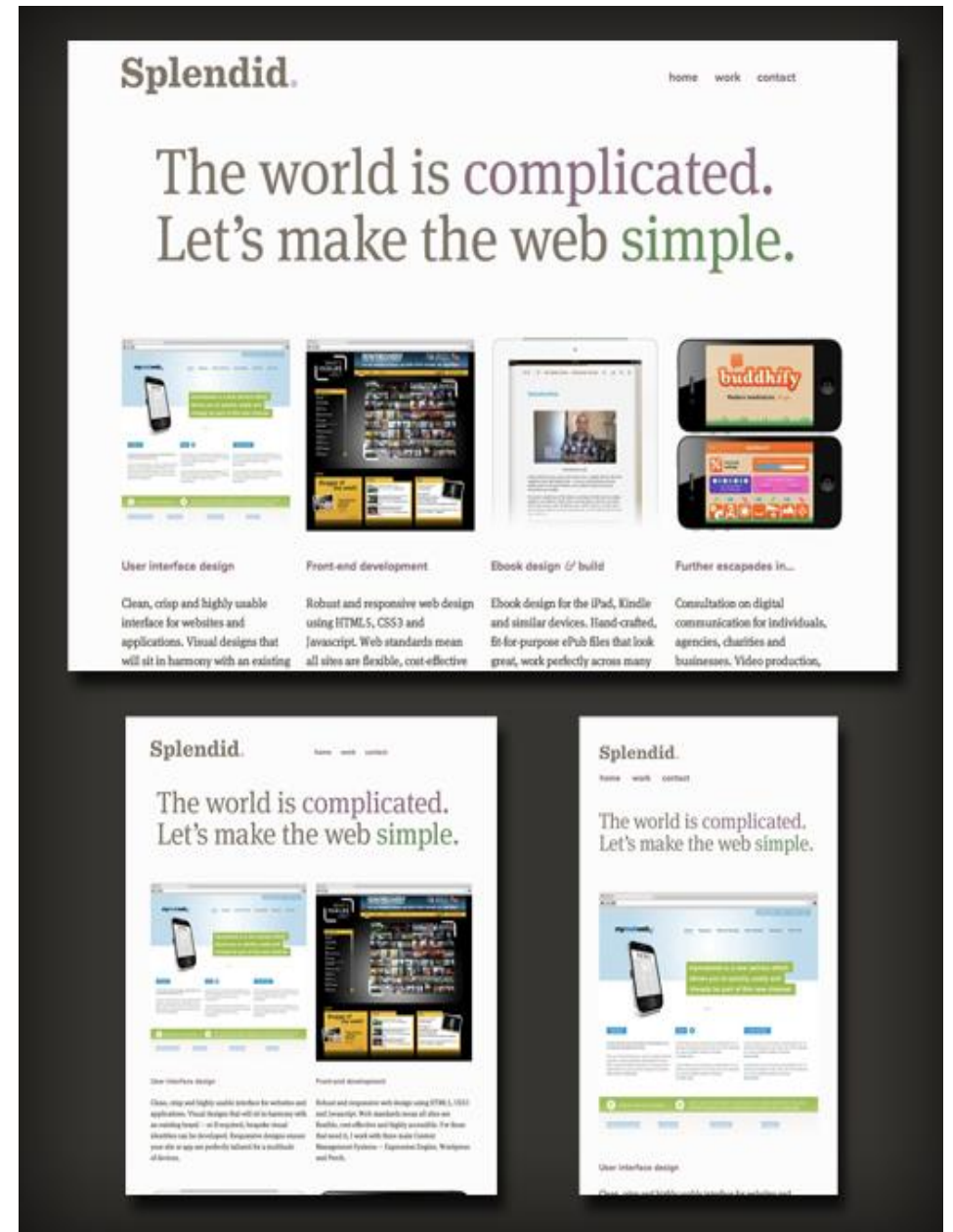
# Мондриан



# Базовая галерея



# Выделенные элементы





# Перемешивание



## orestis

webdevelopment

Voor een breed scala aan webgerelateerde diensten bent u bij Orestis Webdevelopment aan het juiste adres. Een nieuwe website? Geoptimaliseerd voor moderne devices als de iPad en iPhone? Een snelle, hoogwaardige PSD to HTML service? Freelance contracten? Het kan... allemaal!

### webdevelopment

Zou u nieuwe site vereisen, bestaande sites worden grondig onder handen genomen en naar een hoog niveau getild. Door het gebruik van HTML5, CSS3 en JavaScript krijgen bezoekers de best mogelijke user experience!

### slicing service

Als bureau of individu kunt u gebruik maken van de snelle, hoogwaardige PSD to HTML slicing service. Bepikt u tot de ontwerpfase, en laat hindernissen als browsercompatibiliteit en JavaScript over aan de specialist!

### responsive design

Mobiele apparaten zijn hot, maar responsive design is minstens zo hot! Stelt u zich eens voor, één enkele website die perfect opklimt naar de juiste schermgrootte, -verhoudingen en platform verandert. Het kan... en wel nu!

## orestis

webdevelopment

Voor een breed scala aan webgerelateerde diensten bent u bij Orestis Webdevelopment aan het juiste adres. Een nieuwe website? Geoptimaliseerd voor moderne devices als de iPad en iPhone? Een snelle, hoogwaardige PSD to HTML service? Freelance contracten? Het kan... allemaal!

### webdevelopment

Zou u nieuwe site vereisen, bestaande sites worden grondig onder handen genomen en naar een hoog niveau getild. Door het gebruik van HTML5, CSS3 en JavaScript krijgen bezoekers de best mogelijke user experience!

- ✓ Directe user steering, instant site online
- ✓ Integratie in bestaande webs en systemen
- ✓ Persoonlijk contact, korte lijnen
- ✓ Gedeelte ontwikkeling, ook na afloop!

## orestis

webdevelopment

Voor een breed scala aan webgerelateerde diensten bent u bij Orestis Webdevelopment aan het juiste adres. Een nieuwe website? Geoptimaliseerd voor moderne devices als de iPad en iPhone? Een snelle, hoogwaardige PSD to HTML service? Freelance contracten? Het kan... allemaal!

### slicing service

Als bureau of individu kunt u gebruik maken van de snelle, hoogwaardige PSD to HTML slicing service. Bepikt u tot de ontwerpfase, en laat hindernissen als browsercompatibiliteit en JavaScript over aan de specialist!

# Как настроить область просмотра

## Область просмотра (**viewport**)

Определяет, как веб-страница отображается на мобильном устройстве.

Если она не указана, ширина страницы считается равной стандартному значению для ПК, и она уменьшается, чтобы поместиться на экране.

```
<meta name="viewport" content="width=device-width, initial-scale=1" >
```

# Медиа запросы

## Медиа запросы (media queries)

Это простые фильтры, которые можно применять к стилям CSS.

Они позволяют изменять стили на основании характеристик устройства, связанных с отображением контента, включая тип, ширину, высоту, ориентацию и даже разрешение экрана.

Чаще всего используется фильтр по максимальной или минимальной ширине (**min-width: 768px**) , (**max-width: 991px**) или диапазон ширин.

```
@media only screen and (min-width: 320px) and (max-width: 768px) {  
    .col {  
        width: 50%;  
    }  
}
```



# Синтаксис медиа запросов

## @media <условие>

Условие может содержать:

- типы носителей (**all**, **screen**, **print**, ...);
- логические операторы (**and**, **not**, **only**);
- медиа-функции(включающие свойства CSS).

Чаще всего (**min-width: 320px**) ; (**max-width: 320px**) ;

Что бы не возникало проблем с приоритетами селекторов – адаптивные стили пишутся после основных стилей.

```
@media all and (min-width: 320px) {  
    .col {  
  
        width: 50%;  
    }  
}
```

# Как выбирать контрольные точки

**Не определяйте контрольные точки  
на основе классов устройств!**

Если Вы будете ориентироваться на все доступные устройства, бренды или операционные системы, техническое обслуживание сайта станет практически невозможным.

# На какие разрешения экрана можно ориентироваться

- 320px - мобильные устройства с портретной ориентацией
- 480px - мобильные устройства с ландшафтной ориентацией
- 600px - небольшие планшеты
- 768px - планшеты
- 1024px - планшеты, нетбуки и старые мониторы
- 1200px - PC

Оптимально задавать диапазоны(если PC first):

> 991px

- Основные стили(без медиа)

Max-width: 991px

— стили под планшеты и телефоны

Max-width: 768px

—телефоны и ландшафтная ориентация планшетов

Иногда Max-width: 480px

- ландшафтная ориентация телефонов

# Как выбирать ед.измерения

## **px**

Относительная единица измерения, привязанная к конкретному устройству. Плотность пикселей на разных экранах разная.

## **em, rem, %**

Относительная единица измерения, не привязанная к устройству, вычисляется относительно размера шрифта родительского элемента.

## **vw, vh**

Относительная единица измерения, не привязанная к устройству, вычисляется относительно размера области просмотра(viewport).

# Когда использовать %

## Absolute px margin-left



### Comment

Cras ultricies cursus nisl, eget congue tellus consequat nec. Cras id nibh neque, eu dignissim orci. Aenean at adipiscing uma. Suspendisse potenti. Suspendisse vitae tellus a neque commodo rutrum. Praesent ac neque vitae massa pharetra pellentesque.



### Comment

Cras ultricies cursus nisl, eget congue tellus consequat nec. Cras id nibh neque, eu dignissim orci. Aenean at adipiscing uma. Suspendisse potenti. Suspendisse vitae tellus a neque commodo rutrum. Praesent ac neque vitae massa pharetra pellentesque.



### Comm

Cras ultricies cursus nisl, eget congue tellus consequat nec. Cras id nibh neque, eu dignissim orci. Aenean at adipiscing uma. Suspendisse potenti. Suspendisse vitae tellus a neque commodo rutrum. Praesent ac neque

## Relative % margin-left



### Comment

Cras ultricies cursus nisl, eget congue tellus consequat nec. Cras id nibh neque, eu dignissim orci. Aenean at adipiscing uma. Suspendisse potenti. Suspendisse vitae tellus a neque commodo rutrum. Praesent ac neque vitae massa pharetra pellentesque.



### Comment

Cras ultricies cursus nisl, eget congue tellus consequat nec. Cras id nibh neque, eu dignissim orci. Aenean at adipiscing uma. Suspendisse potenti. Suspendisse vitae tellus a neque commodo rutrum. Praesent ac neque vitae massa pharetra pellentesque.



### Comment

Cras ultricies cursus nisl, eget congue tellus consequat nec. Cras id nibh neque, eu dignissim orci. Aenean at adipiscing uma. Suspendisse potenti. Suspendisse vitae tellus a neque commodo rutrum. Praesent ac neque vitae massa pharetra pellentesque.

## Использовать для:

- задания сетки макета;
- размеров изображений;
- видео и других объектов embed, object, video;
- «умных» margin и padding

# Как вычислить значение в em и %

$$\text{em} = \text{value} / \text{parentFontSize}$$

Где

- **value** — исходное значение в px
- **parentFontSize** — размер шрифта родительского элемента в px

$$\% = (\text{element-size} / \text{parent-size}) * 100$$

Где

- **parent-size** — размер родительского контейнера в макете
- **element-size** — размер элемента

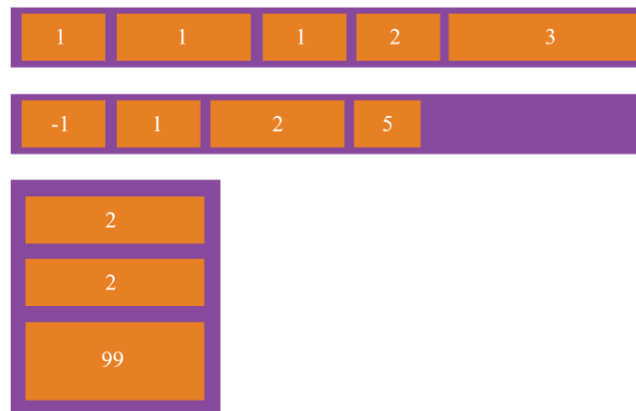
Единицы измерения parent-size и element-size должны совпадать!



Адаптивность на flexbox

# Преимущества Flexbox

- ❖ Блоки легко делаются “резиновым”. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- ❖ Выравнивание по вертикали и горизонтали, базовой линии текста.
- ❖ Порядок элементов в DOM не имеет значения. Его можно изменять в CSS.
- ❖ Элементы могут автоматически выстраиваться в несколько строк/столбцов, занимая все предоставленное место.
- ❖ Удобно делать responsive верстку



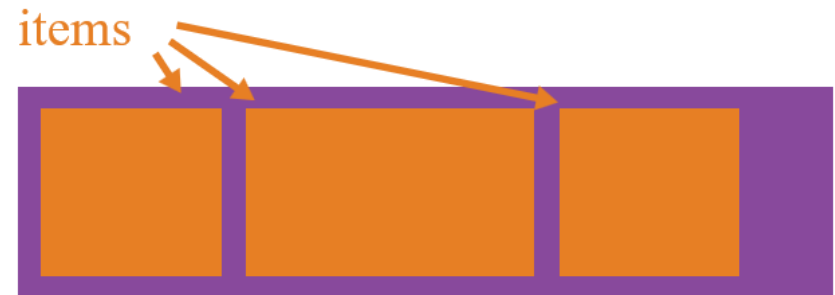
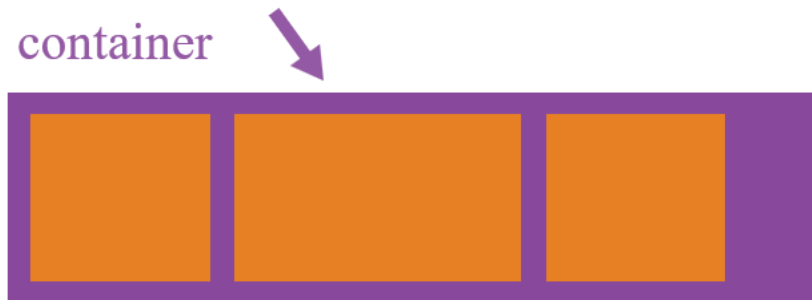
# Синтаксис Flexbox

Задается свойством **display:flex**;

Указывает на то, что элемент должен стать flex-контейнером, а его потомки будут подчиняться особым правилам позиционирования.

Пример кода:

```
<div class="flex-container">  
  <div class="flex-block">item1 </div>  
  <div class="flex-block">item2 </div>  
  <div class="flex-block">item3 </div>  
</div>
```



# Свойства Flex контейнера

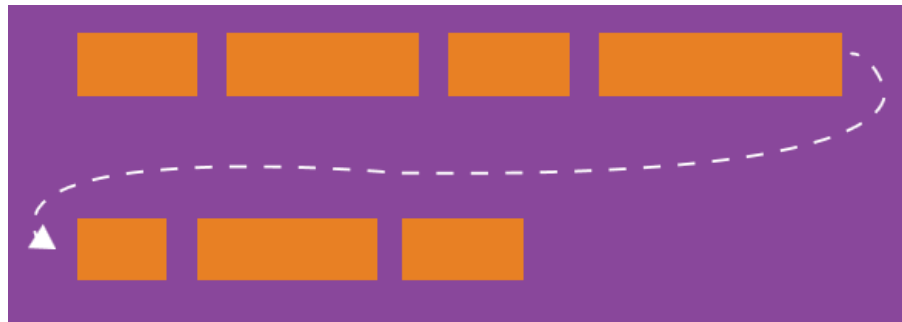
Направление расположения элементов внутри контейнера

**flex-direction:** `row` | `row-reverse` | `column` | `column-reverse`;



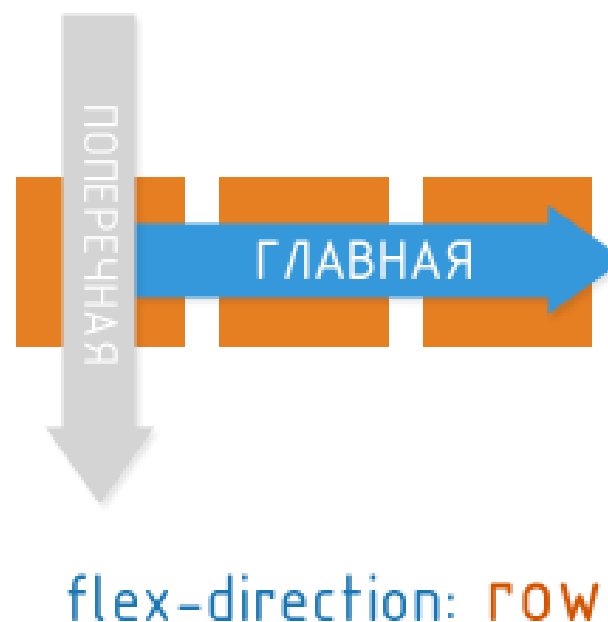
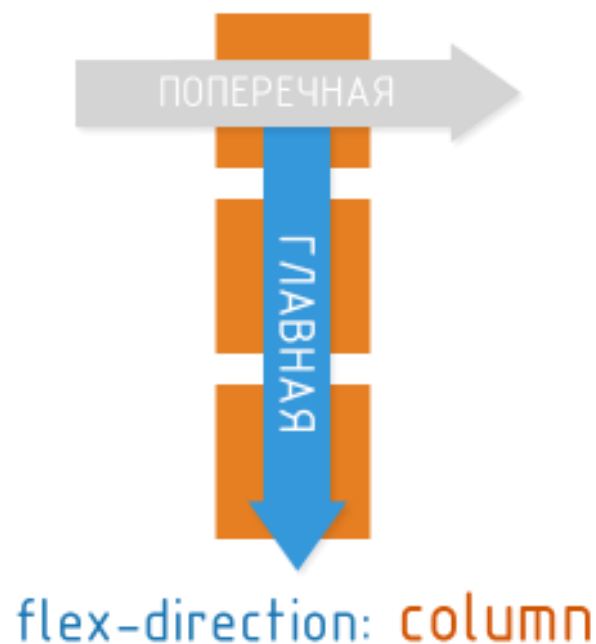
Переход элементов на новый ряд(wrap) внутри контейнера

**flex-wrap:** `nowrap` | `wrap` | `wrap-reverse`;



Применяется к: родительскому элементу flex-контейнера.

# Направление главной оси Flex контейнера

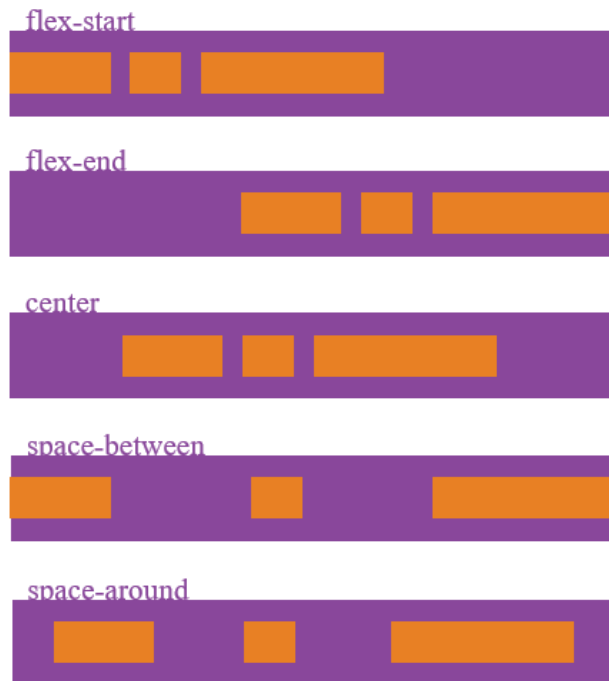


Применяется к: родительскому элементу flex-контейнера.

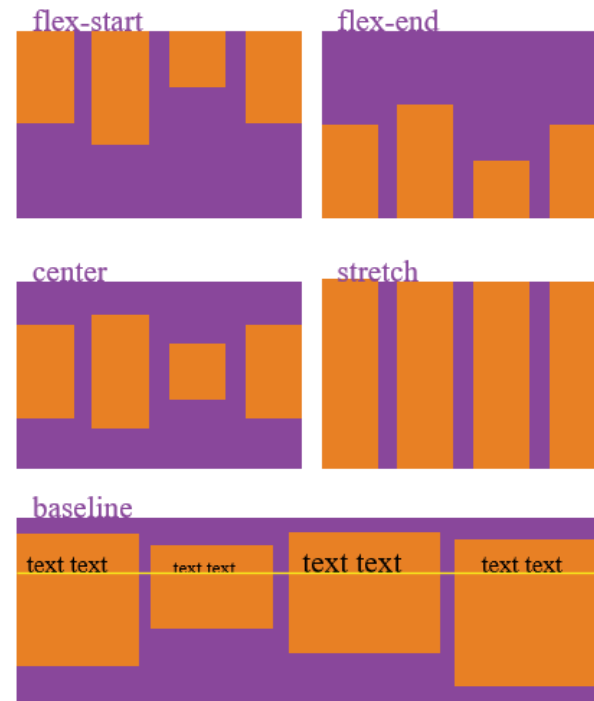
# Свойства Flex контейнера

Выравнивание элементов внутри контейнера по главной оси

**justify-content:** flex-start | flex-end | center | space-between | space-around;



justify-content



align-items

Выравнивание элементов внутри контейнера по поперечной оси

**align-items:** flex-start | flex-end | center | baseline | stretch;

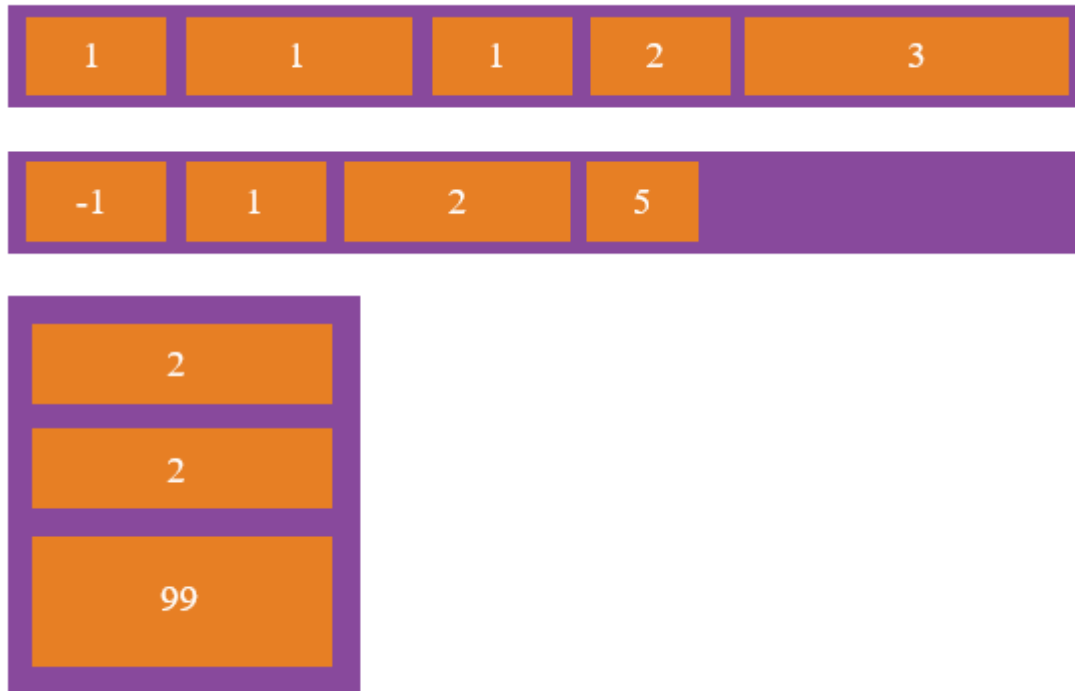
Применяется к: родительскому элементу flex-контейнера.



# Свойства элементов Flex контейнера

Порядок построения элементов вдоль главной оси

**order:** 1 | 0 | -1 | <любое целое число>;

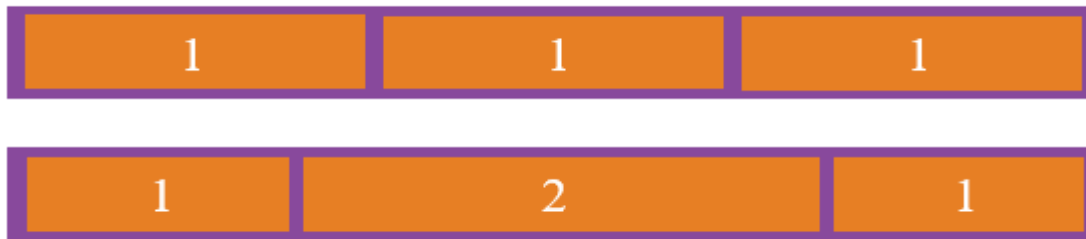


Применяется к: дочерним элементам flex-контейнера.

# Свойства элементов Flex контейнера

фактор “сжимаемости” отдельно взятого flex-блока

**flex-shrink:** `<number>; /* по умолчанию 1 */;`



фактор “расширяемости” отдельно взятого flex-блока

**flex-grow:** `<number>; /* по умолчанию 0 */;`

Применяется к: дочерним элементам flex-контейнера.

# Свойства элементов Flex контейнера

базовый размер отдельно взятого flex-блока

**flex-basis:** (px, em, %, ...) | auto ;

**Аналог свойства width.**

FLEX короткая запись для свойств flex-grow, flex-shrink и flex-basis

**flex:** none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]

```
/* т.е. ... */
.my-flex-block{
    flex-grow:12;
    flex-shrink:3;
    flex basis: 30em;
}
/* это то же самое, что ... */
.my-flex-block{
    flex: 12 3 30em;
}
```

Применяется к: дочерним элементам flex-контейнера.