

Адаптивная и резиновая верстка

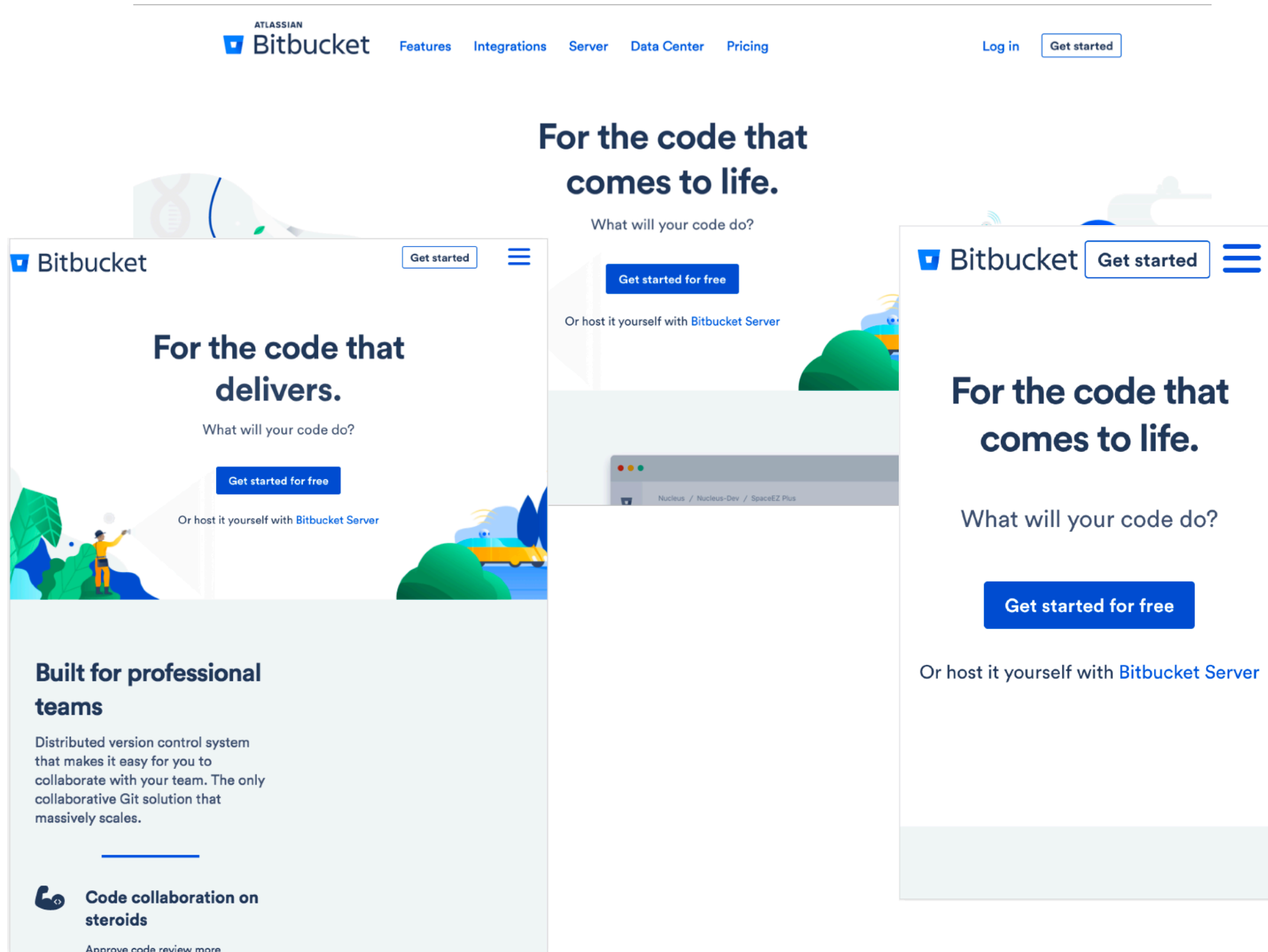
План урока

1. Определим потребности современного сайта
2. Что такое адаптивность?
3. Что такое резиновая или отзывчивая(Responsive) верстка?
4. Первый базовый приём для резиновой верстки. Величины в %.
5. Чего не нужно делать процентами?
6. Медиа запросы, как сделать верстку адаптивной.
7. Какие существуют правила для медиа запросов
8. Примеры медиа запросов.
9. Распространенные контрольные точки
10. Адаптивные изображения
11. Практика.

Каким должен быть сайт в 2017-2018?

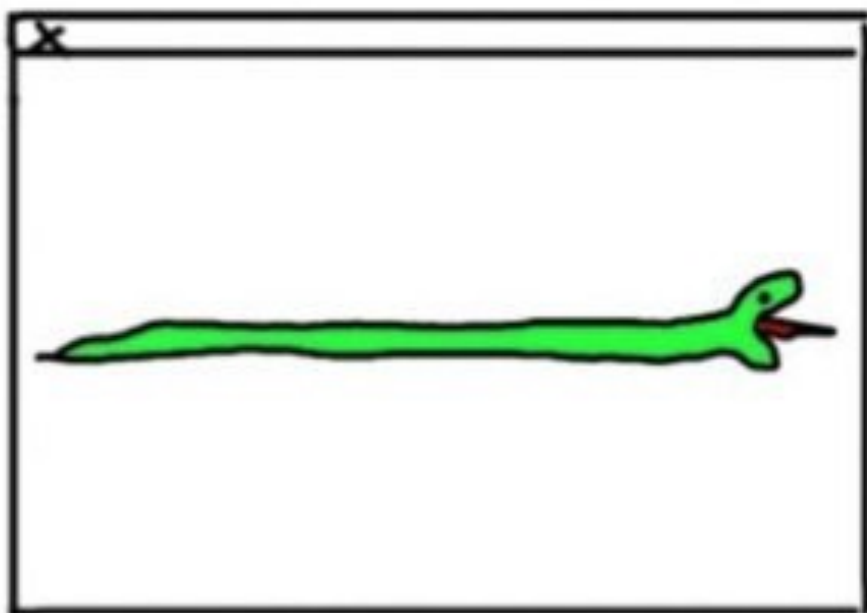
1. Должен хорошо отображаться на смартфонах, компьютерах, умных часах.
2. Должен быть удобным для любого из этих устройств.
3. Должен быть быстрым как с телефона так и с компьютера.
4. Максимальная динамичность сайта. Т.е красивые анимации, создание «живого» сайта.
5. Пользователь должен понимать что происходит на сайте и что ему делать для того чтобы добиться своей цели.
6. В моде минимализм переходящий в футуризм. То есть удобные, легкие, простые сайты с использованием новых технологий и возможностей.

Что такое адаптивность?



Отличие резиновой верстки от адаптивной.

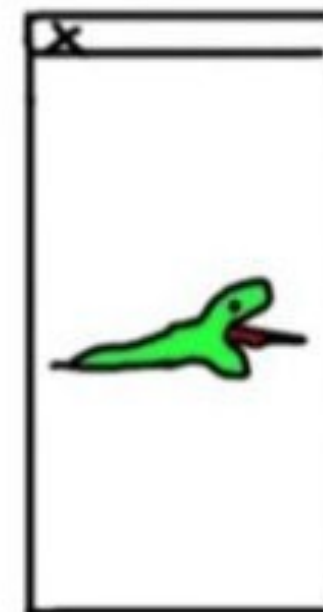
если змея была бы адаптивной,
то она бы выглядела



так?



или так?



Responsive meta тэг

```
<head>
```

```
  <meta name="viewport" content="width=device-  
width, initial-scale=1, maximum-scale=1, user-  
scalable=no">
```

```
</head>
```

Как сделать резиновую верстку?

Вместо

```
.selector{  
    width: 1024px;  
    margin:0 auto;  
}
```

Используем

```
.selector{  
    width: 100%;  
    max-width: 1024px;  
    margin:0 auto;  
}
```

Что мы никогда не делаем в %

1. Размеры маленьких, иконочных изображений
2. Отступы сверху и снизу между блоками, стараемся делать фиксированными
3. Размеры шрифтов.

При мобильной верстке, у нас первыми должны уменьшаться отступы, а затем, если не хватает места, блоки должны становиться друг под друга.

Медиа запросы. Контрольные точки.

```
@media(max-width: 1024px) {  
    .selector{  
        display: none;  
    }  
}
```

1. Медиа запросы ставятся в конце CSS-файла, действует правило каскадности
2. Медиа запросы с одинаковыми параметрами могут повторяться
3. В запросах может быть либо max-width(<) либо min-width(>), а так же специальные правила. Например orientation: **portrait** и **landscape**

Правила в медиазапросах

1. Задаются с помощью операторов **and** или **not**
2. Мы можем выбирать типы устройств
3. Можем регулировать характеристики устройства

Типы устройств

Тип	Описание
all	Подходит для всех типов устройств
print	Предназначен для страничных материалов и документов, просматриваемых на экране в режиме предварительного просмотра печати.
screen	Предназначен, в первую очередь для экранов цветных компьютерных мониторов.
speech	Предназначен для синтезаторов речи.

Характеристики устройств

Ориентация устройства:

@media (orientation: **landscape** или **portrait**)

Отношение пикселей:

@media (device-pixel-ratio: **2**)

Значения минимальной ширины/высоты экрана:

@media (max-width: **720px**)

Примеры медиа-запросов.

```
@media (max-width: 400px), (max-height: 550px) {  
  .wrapper{  
    width: 2rem;  
    line-height: 2rem;  
  }  
}
```

```
@media screen and (max-width: 550px) {  
  .wrapper{  
    width: 2rem;  
    line-height: 2rem;  
  }  
}
```

Распространенные контрольные точки

ширина	устройство
320px	Мобильные телефоны с узким экраном
480px	Современные смартфоны среднего\высокого уровня
768px	Таблетки
960px	Таблетки с широким экраном
1200px	Ноутбуки
1440px	Десктопы

Как скрывать блоки, которые нам не нужны?

```
@media(max-width: 1024px){  
  .selector{  
    display: none;  
  }  
}
```

Мы просто ставим значение `display: none;` в нужной нам контрольной точке

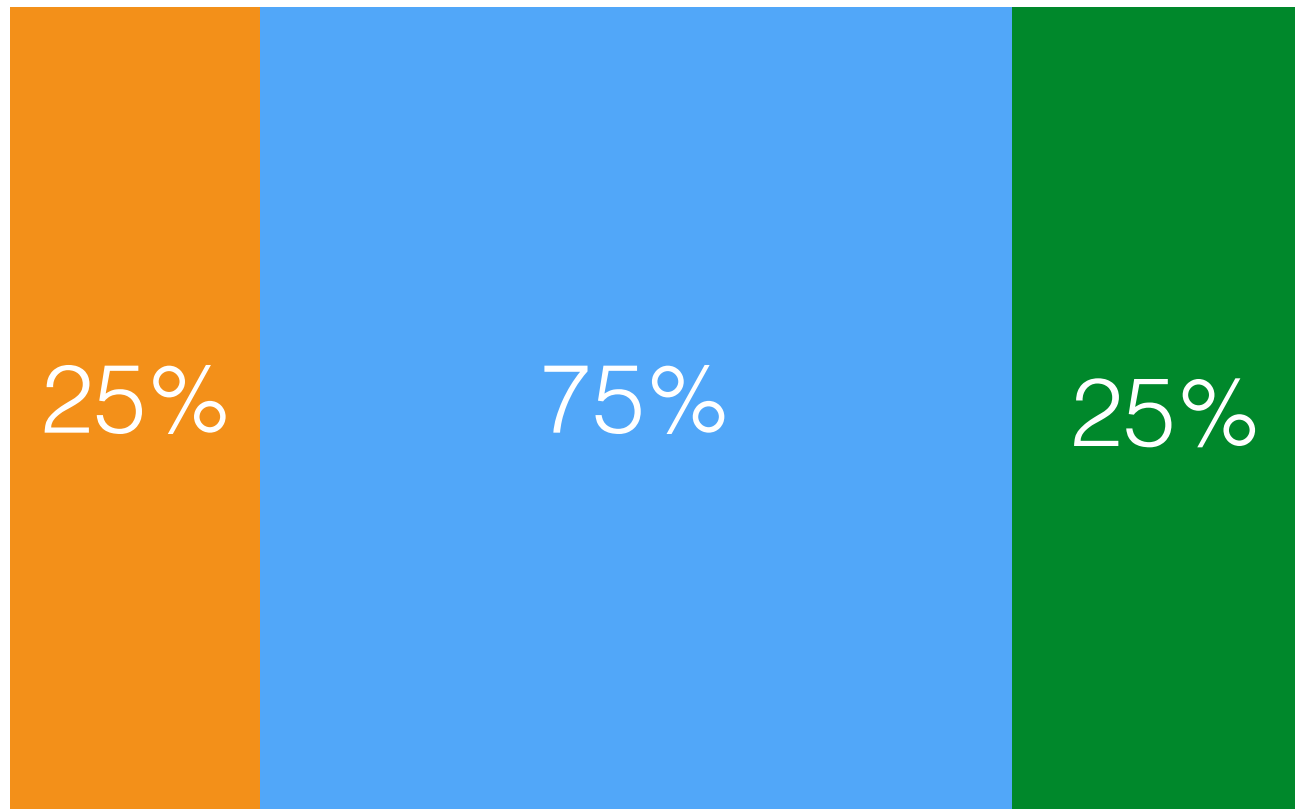
Адаптивные изображения. Атрибуты тегов `img`

- **srcset** - определяет набор изображений, которые мы разрешаем браузеру выбирать, и размер каждого изображения. В него, через запятую, мы записываем пары значений - url картинки и ее ширину в пикселях (для `srcset` это `w`).
- **sizes** - определяет набор условий среды (например, ширины экрана) и указывает, какой размер изображения лучше всего выбрать, когда выполняются определенные условия мультимедиа.
- **source** - тег, используется в `picture`, имеет 2 атрибута, `media` и `srcset`, которые в паре определяют медиаправило и изображение, которое будет отображено при его выполнении

Как отлаживать код в Chrome Dev Tools?

Практическое задание

На экранах > 1024px



На экранах < 1024px

