



# Основы веб-технологий

Василика Климова

Разработчик интерфейсов

@lik04ka

# Artec

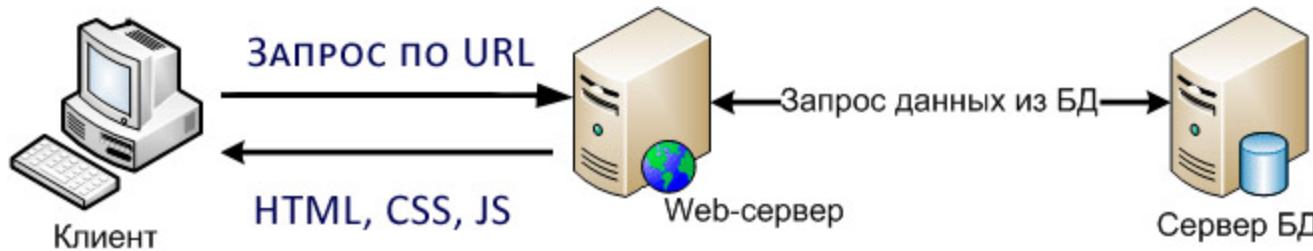
- Shapify
- Viewshape
- Shapify Pro



# День 1

- Основные понятия
- HTML
- CSS
- Верстка

# Веб-сервер



**HTML** - структура веб-страницы.

**CSS** - внешний вид веб-страницы.

**JavaScript** - поведение элементов на веб-странице.

# Internet Protocol Address

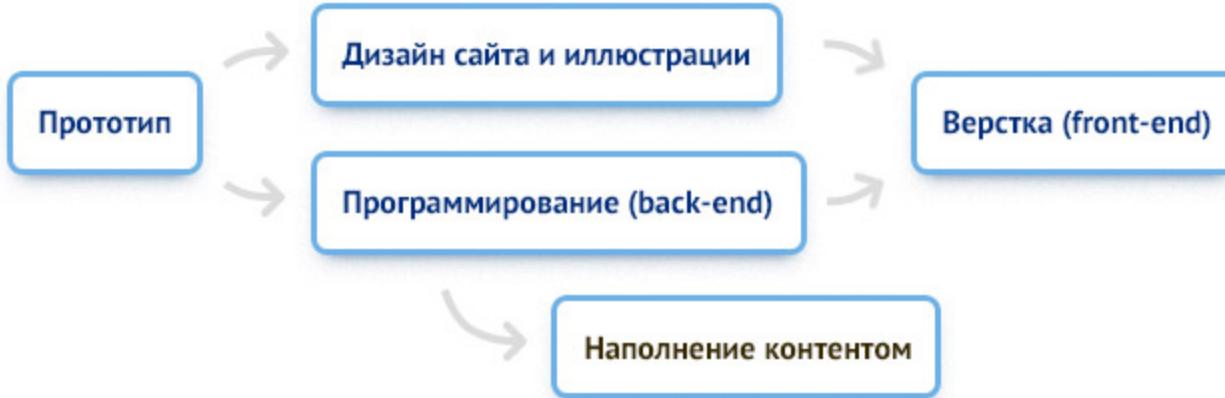
188.40.41.77

[www.htmlweb.ru](http://www.htmlweb.ru)

# Домены

- **http://yandex.ru**
- **www.yandex.ru**
- **https://news.yandex.ru/**
- **https://news.yandex.ru/mynews**
- **https://maps.yandex.ru/?text=moscow**

# Процесс веб разработки



# Прототип



# Дизайн

# Верстка





# Редактор текста

- Sublime Text
- Package Control
- Emmet

**HTML**

# Основные понятия верстки

Тег - HTML команды

Атрибут - различные свойства тегов

```
<тег атрибут="значение">содержимое<тег>
```

Элемент разметки - совокупность пары тегов, атрибутов и содержимого

# HTML

```
<!DOCTYPE html>
```

Тип текущего документа и его синтаксис

```
<html>
```

Контейнер для всего содержимого веб-страницы

```
</html>
```

# Тег head

<head>

Включает заголовок страницы

и техническую информацию

Не отображается напрямую на веб-странице

</head>

# Тег body

```
<body>
```

Тег хранит содержимое веб-страницы

```
</body>
```

# Head. Тег title

```
<title>
```

Заголовок страницы на вкладке браузера

```
</title>
```

# Head. Тег meta

```
<meta>
```

Хранит информацию для браузеров и поисковых систем

```
<meta name="keywords" content="HTML,  
метатег, тег">
```

# Head. Тег style

Стили элементов веб-страницы

```
01.<style type="text/css">  
02. h1 {  
03.     font-size: 22px;  
04.     color: #666666;  }  
05.</style>
```

# Head. Тег link

```
<link>
```

Устанавливает связь с внешним документом вроде файла со стилями или со шрифтами.

```
<link rel = "stylesheet" href = "ie.css">
```

# Head. Тег script

```
<script>
```

Описание скриптов JavaScript

```
</script>
```

```
<script type="тип" src="URL"></script>
```

# Body. Блочные элементы

<div>

Выделяет часть документа для изменения ее вида

<h1></h1> <h6></h6>

Заголовки разного уровня

<p></p>

Текстовый абзац

# Body. Строчные элементы

<span>

Выделяет часть внутри блока для изменения ее вида

<a href = "news.html">Новости</a>

Ссылки

<img src = "images/news1.png">

Изображения



css

# Типы верстки

- Фиксированная верстка
- Резиновая верстка
- Адаптивная верстка
- Отзывчивая верстка

Разница между типами

# Фиксированная верстка

Размеры блоков определены заранее.

Верстка блоков в пикселях.

Пример

# Фиксированная верстка

## Плюсы

- Достаточно одной версии макета
- Предсказуемое поведение макета
- Просто

## Минусы

- Маленькие экраны - горизонтальный скролл

# Резиновая верстка

Ширина блоков зависит от размера окна браузера.

Верстка блоков в процентах.

Пример

# Резиновая верстка

## Плюсы

- Сайт по всей ширине экрана

## Минусы

- Непредсказуемость
- Возможны длинные строки

# Адаптивная верстка

Сайт подстраивается под устройство пользователя.

Может отображать контент по-разному.

Используются CSS медиа-запросы.

Пример

# Адаптивная верстка

## Плюсы

- Отображение зависит от размеров устройства
- Можно опционально скрывать контент

## Минусы

- Макет под каждый тип устройства
- Разработка займет больше времени

# Отзывчивая верстка

Совмещение подходов резиновой и адаптивной верстки.

Блоки зависят от размера браузера.

Используются CSS медиа-запросы.

Пример

# Медиа-запросы

Если ширина окна браузера меньше 650 пикселей

```
@media screen and (max-width: 650px) { }
```

# Отзывчивая верстка

## Плюсы

- Сайт везде выглядит хорошо
- Плавность при изменении размеров браузера

## Минусы

- Макет под каждый тип устройства
- Разработка трудозатратна

# Семантическая верстка

Подход использования тегов согласно их назначению

`<em>` , `<strong>` - смысловое выделение текста

`<i>` , `<b>` - просто оформление курсивом

# Семантическая верстка. Плюсы

- Код лучше анализируется поисковиками
- Хорошие позиции в выдачах поисковиков
- Меньше элементов - быстрей скачивается
- Правильное распознавание голосовыми браузерами

# Подходы к веб разработке

- Изящная деградация
- Прогрессивное улучшение
- Сначала мобильные

# Изыщная деградация

Представление урезанной функциональности для  
пользователей старых браузеров

Возможен полный отказ от поддержки старых браузеров

# Изыщная деградация

## Плюсы

- Просто
- Быстро

## Минусы

- Ориентир только на современные технологии
- Потеря пользователей

# Прогрессивное улучшение

Предоставление базового функционала для всех типов пользователей

# Прогрессивное улучшение

## Плюсы

- Стабильность
- Доступность сайта для всех

## Минусы

- Разработка трудозатратна

# Сначала мобильные

Проектирование сайта начиная с мобильной версии

Затем разработка и дополнение для широких экранов

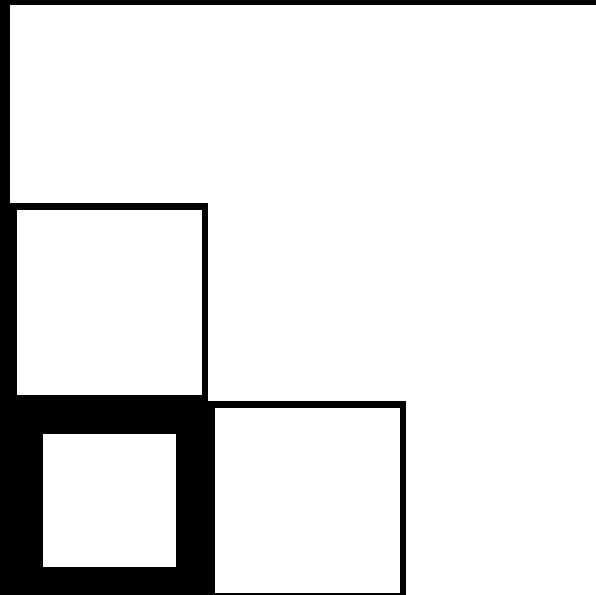
## Когда использовать подход?

Если есть достаточный процент пользователей с  
мобильных устройств

# Полезные ссылки

- HTML5
- htmlbook.ru
- Progressive Enhancement/Graceful Degradation

# Практическая часть



# HTML5

<header>

<footer>

<nav>

<aside>

<audio>

<video>

...

# Универсальные атрибуты

`id`

`class`

`title`

`style`

...

# Свойства CSS

color

background

font-size

text-align

...

# Отличия margin от padding



# Селекторы CSS

#id

.class

div

div > a - все элементы <a>, у которых родитель <div>

a :hover - наведение на элемент <a>

...

# Дерево DOM

```
<!DOCTYPE html>
▼<html lang="en">
  ▼<head>
    <meta charset="UTF-8">
    <title>День 1</title>
    <link rel="stylesheet" href="style.css">
    <link rel="shortcut icon" type="image/x-icon" href="favicon.ico">
  </head>
  ▼<body>
    <div class="hello-block">
      Привет! Меня зовут Лика.
    </div>
    <div class="fixed-block">Фиксированный блок</div>
    <div class="rubber-block">Резиновый блок</div>
    <div class="adaptive-block">Адаптивный блок</div>
    <div class="responsive-block">Отзывчивый блок</div>
  </body>
</html>
```

DevTools

# Практическая часть

## Выводим на странице

- Привет! Меня зовут <Имя> <Фамилия>

# Практическая часть

## Выводим на странице

- Добавляем странице заголовок: **День 1**

# Практическая часть

- Добавляем блок
- Указываем ему класс
- Задаем свойства background
- Свойства margin, padding

# Собственные шрифты

- Скачиваем нужный шрифт
- Или находим на Google Fonts

# Правило @font-face

Формат WOFF самый легковесный

```
01. @font-face {  
02.   src: url(fonts/gentium.woff);  
03.   font-family: Gentium;  
04.   font-weight: 400;  
05. }
```

# Исходники

[github.com/Likita](https://github.com/Likita)



# Василика Климова



likita



vasilika.klimova



lik04ka