Горизонтальная линия



ПЛАН ОБУЧЕНИЯ

Автор Сергей Никитченко

Ученик Андрей

# Введение

## Представиться, узнать какие цели преследует ученик

Здравствуйте, Андрей меня зовут Сергей.

…

Предлагаю перейти на ты?

…

Можно поинтересоваться, для чего вы решили изучать верстку сайтов/ну или в принципе JS

…

## Узнать какие знания имеются у пользователя

Вы ведь уже занимались изучением программирование, что именно освоили

…

А были какие-то вопросы, сложности, непонятки. Только отвечаем честно.

…

Мне важно чтобы вы мне говорили о ошибках и всём перечисленном выше. Чтобы спрашивали, так как без этого продуктивности не будет

## Введение в структуру курса

Сразу предупрежу процентов 70 каждого урока это будет практика.

Также буду давать много домашних заданий, но непосредственно отталкиваясь то вашего свободного времени

А сейчас я немного познакомлю вас со структурой курса

Я буду обучать такому направлению как frontend

Расскажу что такое вообще этот frontend.

Это направление которое служит для того чтобы создавать интерфейс и обрабатывать взаимодействия пользователей с этим же самым интерфейсом и серверной частью сайта.

А именно:

* написание структуры сайта это — HTML
* добавить стили это — CSS
* ну и как раз таки добавление магии на сайт это — JS

Так как вы выбрали Курс обучение JS, то я вас познакомлю лишь с базовыми знаниями HTML/CSS

## Рассказать о плане изучения

А план обучения у меня таков:

* Изучение HTML/CSS
* Изучение JS
  + Основы
  + Es6
* React

Будем внедрять динамические скрипты

Познакомлю со сборщиком пакетов npm

После чего мы будем Осваивать Git

И добавлять сайт на хостинг

А также я познакомлю вас с node.js, который позволит вам создавать клиент серверные приложения

## Узнать о текстовом редакторе

А о компьютере

…

Тогда я сегодня сброшу видео с установкой и настройкой редактора

# содержание

Оглавление

[1 Введение 2](#_Toc39304073)

[1.1 Представиться, узнать какие цели преследует ученик 2](#_Toc39304074)

[1.2 Узнать какие знания имеются у пользователя 2](#_Toc39304075)

[1.3 Введение в структуру курса 2](#_Toc39304076)

[1.4 Рассказать о плане изучения 3](#_Toc39304077)

[1.5 Узнать о текстовом редакторе 4](#_Toc39304078)

[2 содержание 5](#_Toc39304079)

[3 основы html/css 7](#_Toc39304080)

[3.1 Синтаксис 7](#_Toc39304081)

[3.2 Основные концепции html/css 7](#_Toc39304082)

[3.3 Семантика 7](#_Toc39304083)

[3.4 Sass/Less 7](#_Toc39304084)

[3.5 Grid / Flexbox 7](#_Toc39304085)

[4 Основы js 8](#_Toc39304086)

[4.1 Переменные 8](#_Toc39304087)

[4.2 Типы данных 8](#_Toc39304088)

[4.3 Основные концепции js 9](#_Toc39304089)

[4.4 Объекты Начальные данные 9](#_Toc39304090)

[4.5 Качество кода 10](#_Toc39304091)

[5 Объекты: основы 11](#_Toc39304092)

[6 Типы данных 12](#_Toc39304093)

[7 Продвинутая работа с функциями 13](#_Toc39304094)

[8 Свойства объекта, их конфигурация 14](#_Toc39304095)

[9 Прототипы, наследование 15](#_Toc39304096)

[10 Классы 16](#_Toc39304097)

[11 Обработка ошибок 17](#_Toc39304098)

[12 Промисы, async / await 18](#_Toc39304099)

[13 Генираторы, продвинутая итерация 19](#_Toc39304100)

[14 Модули 20](#_Toc39304101)

[15 Дополнительно 21](#_Toc39304102)

# основы html/css

## Синтаксис

## Основные концепции html/css

## Семантика

## Sass/Less

## Grid / Flexbox

# Основы js

## Переменные

* var
* let
* const

let и var они схожи с возможностями, так что если увидишь в каком-то скрипте var, смело заменяй на let, в основном будет работать правильно.

Скажу сразу var сейчас я запрещаю пользоваться, так как это старая технология с очень коварной областью видимости

А в основном будем пользоваться const. А let, если данные нужно заменить

-- Вопросы есть?

….

## Типы данных

Бывают разные типы данных

* <https://learn.javascript.ru/data-types>
* String
* Number
* Object
* Undefended
* Null
* Bulean
* Symbol

## Основные концепции js

* console.log();
* точка с запятой
* функции

## Проверка знаний

Написать функцию которая будет выводить сообщение с input по нажатию энтер

Html

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
</head>  
<body>  
<form onsubmit="return false">  
 <input id="kinput">  
</form>  
<script src="index.js"></script>  
</body>  
</html>

Js

kinput.onkeydown = *handle*;  
  
**function** *handle*(e) {  
 **if**(e.**key** === **"Enter"**){  
 **const** mess = ***document***.getElementById(**"kinput"**);  
 ***console***.log(mess.**value**);  
 }  
}

## Объекты Начальные данные

* Массивы

[ , ] Основная запись

* Объекты

{ , } Основная запись

* Массивы объектов

[ {} , {} ]

* Объекты массивов

{ array: [] }

* Объект с функцией

{

fans () {

…

},

function: fun = () => {

…

}

}

## Качество кода

* Использование строгого режима “use strict”
* Разделение бизнес логики и интерфейса
* Использование методов массивов

# Объекты: основы

# Типы данных

# Продвинутая работа с функциями

# Свойства объекта, их конфигурация

# Прототипы, наследование

# Классы

# Обработка ошибок

# Промисы, async / await

# Генираторы, продвинутая итерация

# Модули

# Дополнительно