

БАЗОВЫЕ ПОНЯТИЯ. ПЕРЕМЕННЫЕ И ЧИСЛА



КОНСТАНТИН ПОЛЯНСКИЙ



КОНСТАНТИН ПОЛЯНСКИЙ



kv.polyanskiy@gmail.com



[@kvpolyanskiy](https://t.me/kvpolyanskiy)

ПЛАН ЗАНЯТИЯ

1. [Добро пожаловать](#)
2. [План курса](#)
3. [Как работает веб \(клиент—сервер\)](#)
4. [Введение в JavaScript](#)
5. [Программисты](#)
6. [Переменные \(let\)](#)
7. [Целые числа и числа с плавающей точкой + операторы](#)
8. [Операция / выражение](#)
9. [Чему мы научились](#)

ПРИВЕТ!

Приветствуем вас на базовом курсе по основам программирования на примере популярного языка **JavaScript**.

На этом курсе мы с вами научимся:

- работать с основными понятиями;
- применять базовые навыки;
- мыслить как инженер;
- искать ошибки в нашем коде;
- писать красивый и чистый код.

Закрепить полученные знания вам поможет **домашняя работа** после каждой лекции. В конце курса вы напишете свою первую собственную игру!



ПЛАН НАШЕГО КУРСА

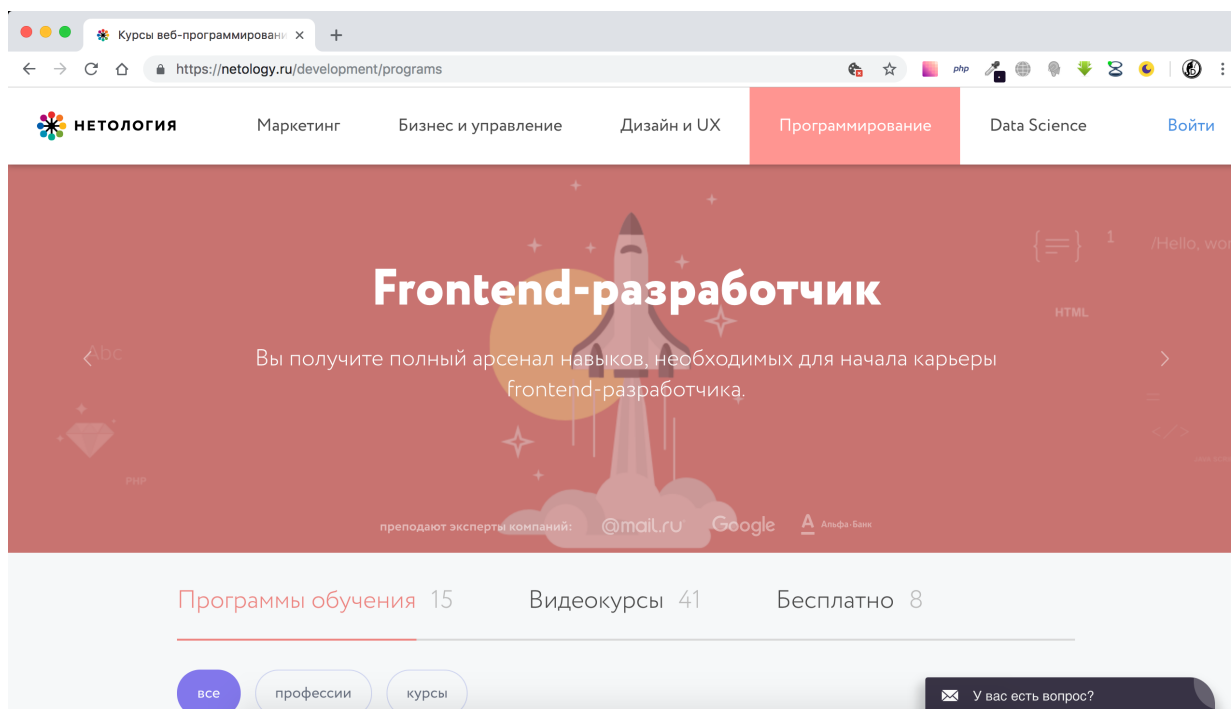
- Блок 1: Типы данных
- Блок 2: Структуры данных и циклы
- Блок 3: Функции
- Дипломная работа (своя собственная игра)

Главный **message** нашего курса:

JavaScript — это увлекательно!

ВАШИ МЫСЛИ

1. Как вы думаете, как работает сайт?
2. Видим ли мы всю работу сайта?
3. Что происходит после того, когда мы вводим адрес сайта?

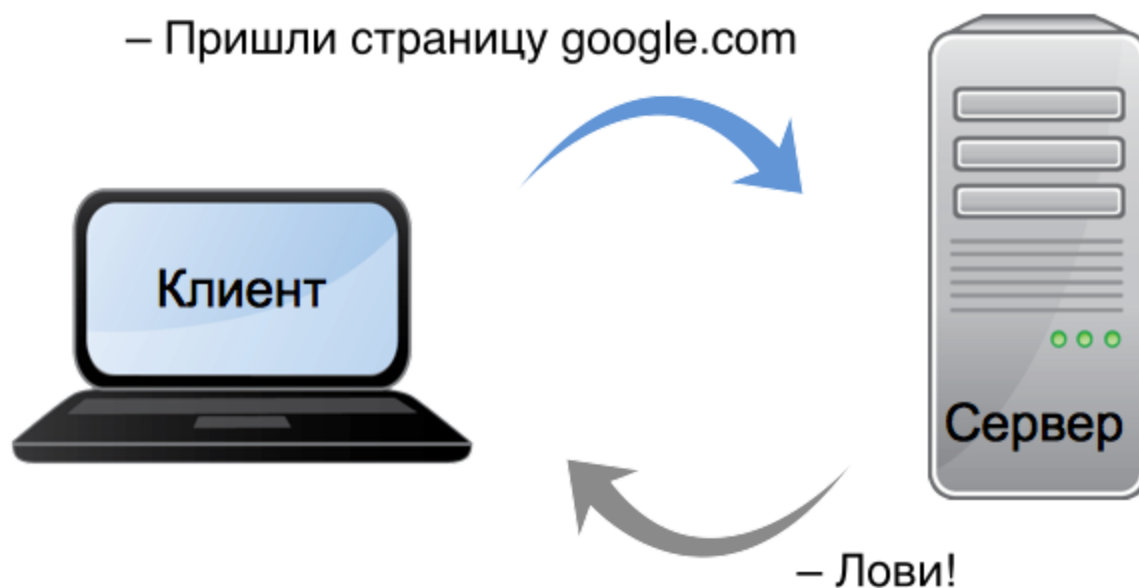


КАК РАБОТАЕТ ВЕБ (КЛИЕНТ–СЕРВЕР)

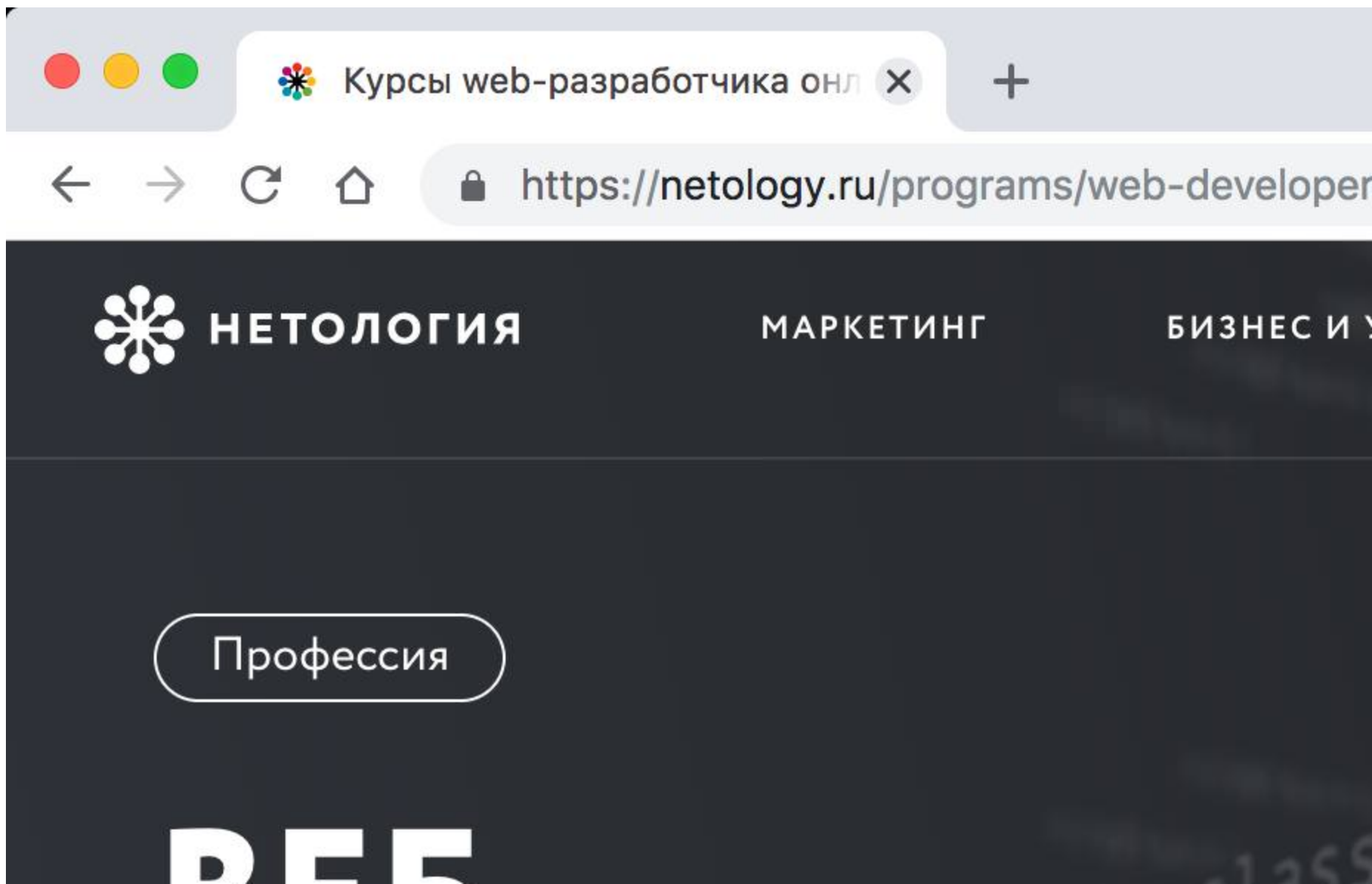
Клиент посылает запросы, а сервер присылает **ответы**.

HTTP ответы сервера содержат код ответа

– Пришли страницу google.com



ОСНОВЫ ПРОГРАММИРОВАНИЯ



ОСНОВЫ ПРОГРАММИРОВАНИЯ

Фронтэнд — это фасад сайта, то, с чем взаимодействует посетитель: весь текст, картинки, кнопки и так далее. На нашем [курсе по фронтэнду](#) мы изучаем HTML, CSS и JavaScript. Пройдя его можно стать полноценным junior-разработчиком в короткие сроки и получить 5 готовых проектов в резюме.

Бекэнд — это задний двор: устройство веб-сервера, которое отвечает за то, чтобы код работал правильно, письма отправлялись, статьи отображались и так далее. На нашем [курсе по бекэнду](#) мы изучаем PHP и базы данных.

ВАЖНО

Наш с вами курс — это **основы программирования**. Независимо от того профессионального пути который вы выберете, базовые знания и основы понадобятся как при работе с фронтэндом, так и с бекэндом.



ПРОГРАММЫ

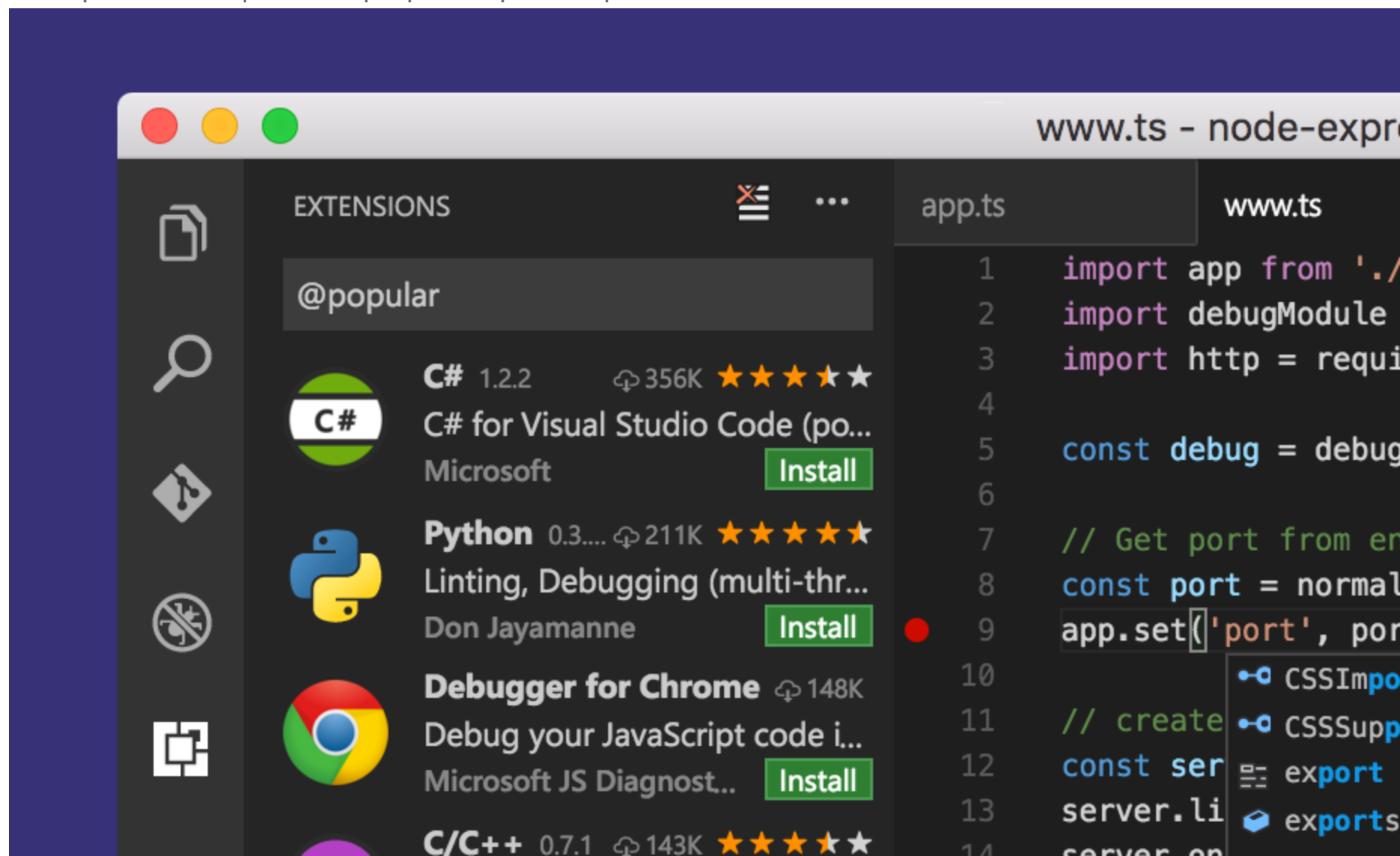
На сервере и на клиенте работают программы/скрипты.

Программа — код и данные, предназначенных получения определённого результата. Программы обычно большие и сложные, их нужно устанавливать.

Помимо программ существуют *скрипты*. Скрипты обычно выполняют набор простых действий.

ПРОГРАММЫ

На картинке изображена программа редактора кода:






С КАКИМИ ПРОГРАММАМИ ВЫ РАБОТАЕТЕ?

Мы все знакомы с *MS Word, PowerPoint, Excel, Outlook Express*, а также браузерами для работы в интернете — *Opera, Google Chrome, Mozilla Firefox* и др.

Ваши варианты?

ЧТО ПРОИСХОДИТ КОГДА МЫ ПЕРЕХОДИМ НА САЙТ?

 <https://www.google.com>

1. В строке браузера пишем адрес сайта (напр. www.google.com)
2. Запрос отправляется на сервер (*фронтэнд*)
3. Скрипт обрабатывает запрос на сервере (*бекэнд*)
4. Сервер возвращает ответ (*бекэнд*)
5. Браузер получает веб-страницу с кодом html и JavaScript и интерпретирует ее (*фронтэнд*)
6. После интерпретации мы видим веб страницу со всеми картинками, текстами и другими элементами



ЯЗЫК ПРОГРАММИРОВАНИЯ

Это набор синтаксических правил, соблюдая которые получится написать программу.

Языки:

- C++;
- C#;
- Java;
- Python;
- JavaScript;
- PHP.



ПРОГРАММИСТ

Программист — это специалист, который занимается разработкой компьютерных программ.

- *Системные программисты* разрабатывают операционные системы, работают с сетями.
- *Web-программисты* пишут программную составляющую сайтов, создают динамические веб-страницы, web-интерфейсы для работы с базами данных.
- *Прикладные программисты* — это специалисты, занимающиеся разработкой и внедрением программного обеспечения.



ЗНАКОМИМСЯ С JAVASCRIPT

JavaScript — скриптовый язык который выполняется интерпретатором, чаще всего браузером.

JavaScript делает web-странички «живыми». Программы на этом языке называются скриптами.

Для написания кода будем использовать [Repl.it](https://repl.it)

КАК РАБОТАТЬ С REPL.IT

Repl — это онлайн платформа где можно пробовать программировать.



repl.it

jobs

blog

features

Get

Stop wasting time
instant IDE

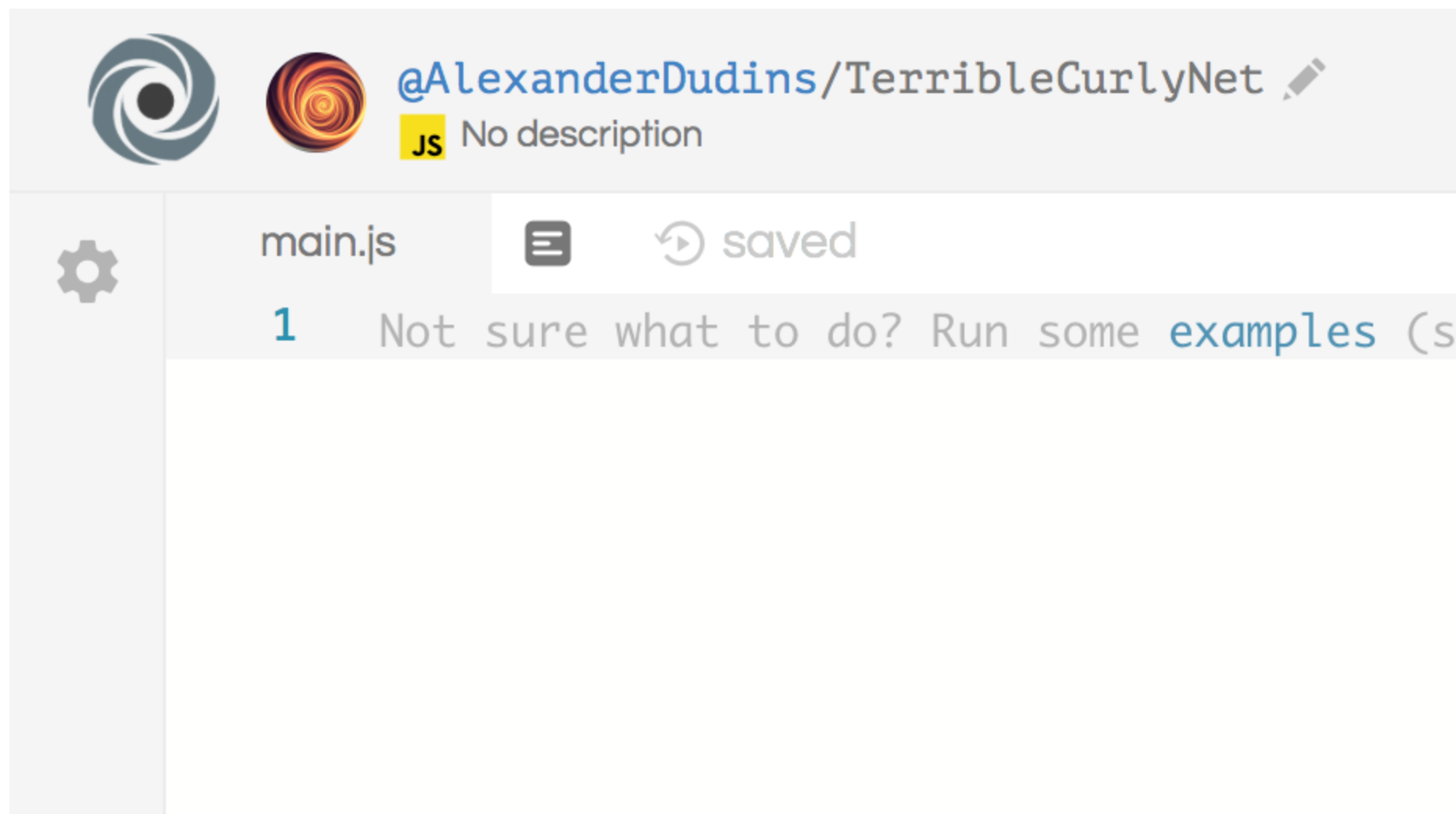


КАК РАБОТАТЬ С REPL.IT

1. Переходим на repl.it
2. Регистрируемся, нажав *Sign up*
3. Пропускаем tutorial нажав *Start coding now*
4. Выбираем язык — JavaScript

КАК РАБОТАТЬ С REPL.IT

Если вы видите



ПОРАБОТАЕМ В REPL.IT

В левой светлой половине мы будем писать код, а в правой будем получать результат.

Напишем пример и нажмем кнопку **Run!**

```
5 + 5;  
10 * 10 + 15;  
(100 - 8) / 2;
```

ВАЖНО

Код выполняет сверху вниз! Сначала выполняется первая строка, потом вторая и так далее до конца.

УСЛОЖНЯЕМ

Решим пару задач

Задача 1: Расстояние между пунктами А и В равно 100км. Человек движется с постоянной скоростью 5км/ч из пункта А. За сколько времени наш герой достигнет пункта В?

Задача 2: Расстояние между пунктами А и С равно 140км. Человек движется с постоянной скоростью 5км/ч. За сколько времени наш герой достигнет пункта С?

Решение 1:

$$100 / 5;$$

Решение 2:

$$140 / 5;$$

Скорость одинаковая в обеих задачах. Дважды писать лень и поэтому мы можем воспользоваться *переменной*!

ПЕРЕМЕННЫЕ (`let`)

Переменные используются для хранения информации.

В *JavaScript* для создания переменной используется слово let

```
1 | let number;
```

Чтобы положить информацию в переменную

```
1 | number = 10;  
2 | // number равно 10
```

Значение можно изменять

```
1 | number = 20;  
2 | // а теперь number равно уже 20 !!!
```

НЕ ОБЪЯВЛЯЙТЕ ДВА РАЗА ПЕРЕМЕННУЮ С ОДИНАКОВЫМ ИМЕНЕМ

Мы можем менять значение у переменной, но объявлять одну и ту же два раза мы не можем, будет ошибка:

```
1  let number;  
2  number = 5;  
3  
4  let number; // ошибка!
```

СОЗДАНИЕ ПЕРЕМЕННОЙ И ПРИСВАИВАНИЕ ЗНАЧЕНИЙ МОЖНО ОБЪЕДИНИТЬ

Вместо данной записи:

```
1 | let number;  
2 | number = 5;
```

Можно написать короче:

```
1 | let number = 5;
```




ЧТО МЫ МОЖЕМ ХРАНИТЬ В ПЕРЕМЕННОЙ?

Это зависит от того, с какой информацией вы работаете.

Мы рассмотрим работу с некоторыми вариантами:

- Целые числа (*10, 5, 16*)
- Числа с плавающей точкой (*10.12, 5.2, 1.4*)
- Строки (*“Маша”, “Фрукты”, “слова”*)

ЦЕЛЫЕ ЧИСЛА И ЧИСЛА С ПЛАВАЮЩЕЙ ТОЧКОЙ + ОПЕРАТОРЫ

В JavaScript можно записывать как целые, так и дробные числа. Числа с плавающей точкой имеют целую и дробную часть. Например:

```
1 | let number1 = 10;  
2 | let number2 = 10.25;
```

РАССМОТРИМ ПРИМЕРЫ РАБОТЫ ОСНОВНЫХ ОПЕРАТОРОВ

1. Операции присваивания
2. Арифметические операторы

```
1  let operator = 0;  
2  let operator == 'Равенство';  
3  
4  let a = 10, b = 5;  
5  console.log(a + b); // 15  
6  console.log(a - b); // 5  
7  console.log(a * b); // 50  
8  console.log(a / b); // 2  
9  
10 // совмещаем:  
11 let c = a + b;
```

СТОП! ЧТО ЗА `console.log()` ?

С помощью `console.log()` можно информацию на печать.

Например:

А КАК НАПЕЧАТАТЬ ТЕКСТ?

В JavaScript текстовые данные называются строками.

```
1 let string = "Это строка";  
2  
3 let anotherString = 'Другая строка';  
4  
5 let name = "Alex";
```

Кавычки можно использовать как двойные, так и одинарные.

```
console.log(name); // Alex
```

А КАК ВЫВЕСТИ ТЕКСТ СО ЗНАЧЕНИЕМ ПЕРЕМЕННОЙ?

Мы можем подклеить значение переменной в конец строки с помощью оператора `+`:

```
1 let appleQuantity = 5;  
2  
3 let text = "Количество яблок: ";  
4  
5 console.log(text + appleQuantity);
```

или даже без переменной `text`:

```
1 let appleQuantity = 5;  
2  
3 console.log("Количество яблок: " + appleQuantity);
```

ТЕПЕРЬ МЫ МОЖЕМ КРАСИВО ОФОРМИТЬ НАШУ ЗАДАЧКУ

Расстояние между пунктами **A** и **B** равно 100км. Человек движется с постоянной скоростью 5км/ч из пункта **A**. За сколько времени наш герой достигнет пункта **B**?

```
1 let distance = 100;  
2 let speed = 5;  
3 let time = distance / speed;  
4  
5 console.log(time); // 20
```

С переменными наш код стал читаться лучше!

ЧЕМУ МЫ НАУЧИЛИСЬ?

1. Разобрали функционирование сайтов и как работает веб;
2. Что такое фронтенд и бэкенд;
3. Познакомились с JavaScript;
4. Рассмотрели базовые операторы;
5. Поработали с разными типами данных.

РАДУЕМСЯ!



ДОМАШНЕЕ ЗАДАНИЕ

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаем в группе Facebook!
- Задачи можно сдавать по частям.
- Зачет по домашней работе проставляется после того, как приняты все **3 задачи**.



ЧТО ПОЧИТАТЬ

- [Как работает интернет](#)
- [Выражения и операторы](#)
- [Введение в JavaScript](#)



А ПОСМОТРЕТЬ?

- [Еще одна лекция о числах в JavaScript](#)
- [Как работает JavaScript](#)
- <https://www.youtube.com/watch?v=fXi49EkrMPI>



Спасибо за внимание! Время задавать вопросы

КОНСТАНТИН ПОЛЯНСКИЙ



kv.polyanskiy@gmail.com



[@kvpolyanskiy](https://www.instagram.com/kvpolyanskiy)