NODE JS

Урок №3

Кто слышал про понятие DRY?

Определение

DRY (don't repeat yourself) - это принцип разработки программного обеспечения, нацеленный на снижение повторения информации различного рода

НЕ ПОВТОРЯЙ СЕБЯ

Кто слышал про функции?

Определение

Функция - это описанный алгоритм действий, который выполнится при ее вызове.

Бывает исключение, когда функция вызывает сама себя. Это называется рекурсией.

Важность функций

Те, кто задумывался о том, что мы часто пишем один и тот же код в одной программе точно понимают, что это плохая практика. Функции позволяют:

- Написать один раз код и переиспользовать его множество раз
- Придерживаться принципа DRY

Виды функций

1. Именованные
2. Анонимные
3. Стрелочные

Где и с какими функциями мы встречались до этого?

Что это за функция?

```
1.  const express = require('express')
2.  const app = express()
3.
4.  app.get('/', function(){
5.      res.end('Hello from node')
6.  })
```

Виды функций

Именованная

Название!

```
function   myFunc()
{
    console.log(1)
}
```

Анонимная

```
function()
{
    console.log(1)
}
```

Стрелочная

```
()                  =>
{
    console.log(1)
}
```

Чтобы тело функции выполнялось, ее нужно вызывать!

Создание и вызов функции

```
1.  function helloFunc() {
2.      console.log('Привет из функции');
3.  }
4.
5.  helloFunc();
```

Сработает ли?

```
1.  helloFunc();
2.
3.  function helloFunc() {
4.      console.log('Привет из функции');
5.  }
```

ДА, СРАБОТАЕТ!

А еще у функций есть параметры и аргументы.

ОПРЕДЕЛЕНИЕ

Параметры - это локальные данные, которые принимаются функцией.

Аргументы - это передающиеся, при вызове функции, данные.

Параметры и аргументы

```
1.  function consolePlus(a) {
2.      console.log(a + 1)
3.  }
4.
5.  consolePlus(55)
```

Название параметра - это то, как он будет использоваться внутри функции.

Основная идея функции

Как правило, мы передаем функции аргументы, чтобы она с ними что-то сделала и вернула результат. Например, что-то посчитала.

- Вызываем функцию и передаем аргументы
- Функция принимает параметры
- Что-то с ними делает
- Возвращает результат

Как же что-либо вернуть из функции?

Оператор return

Вернуть значение

```
1.  helloFunc();
2.
3.  function helloFunc() {
4.      return “Эта строка вернется из функции”;
5.  }
```

А куда же вернется?

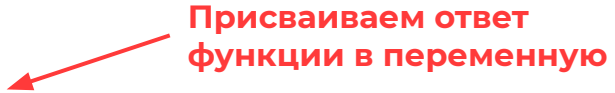
Туда, где функция вызывалась

Куда вернется значение

```
1.  helloFunc();
2.
3.  function helloFunc() {
4.      return “Эта строка вернется из функции”;
5.  }
```

Так как мы с вернувшимися из функции данными ничего не сделали, вызов был бессмысленным.

Куда вернется значение




```
1.  let result = helloFunc();
2.
3.  function helloFunc() {
4.      return “Эта строка вернется из функции”;
5.  }
```

Что будет в переменной result?

```
1.  let result = summ(1, 10);
2.
3.  function summ(a, b) {
4.      return a + b;
5.  }
```

Какие есть вопросы?

Практика

20 минут