



Frontender[1.0] JavaScript - Замыкания (closure). Область видимости переменных

YouTube	https://youtu.be/2gH8Opn9KfY
Telegram	https://t.me/Dmitry_Kolotilshikov
Github	https://github.com/DmitryKolotilshikov/
Boosty	https://boosty.to/dmitry_ko
# Номер урока	38

Задачи к этому уроку тут https://boosty.to/dmitry_ko

Полезные ссылки: <https://learn.javascript.ru/closure>



ЗАМЫКАНИЯ

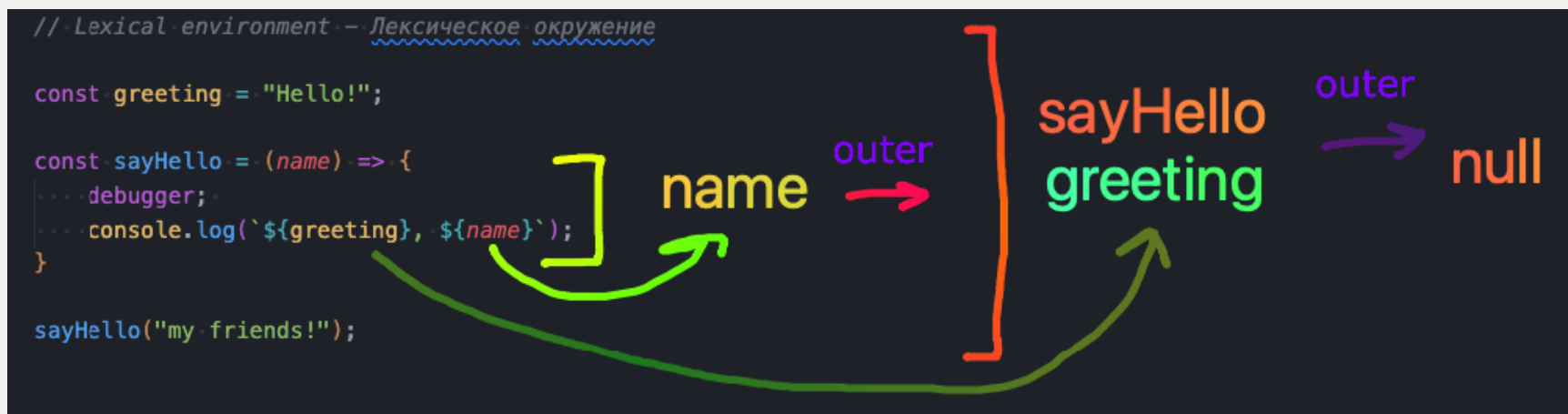
Замыкание – это функция, которая запоминает свои внешние переменные и может получить к ним доступ. В некоторых языках это невозможно. Но, в JavaScript, **все функции изначально являются замыканиями**. То есть функции автоматически запоминают, где были созданы, с помощью скрытого свойства `[[Environment]]`, и все они могут получить доступ к внешним переменным.

В JavaScript у каждой выполняемой функции, блока кода `{...}` и скрипта есть связанный с ними внутренний (скрытый) объект, называемый *лексическим окружением* `LexicalEnvironment`.

Объект лексического окружения состоит из двух частей:

1. *Environment Record* – объект, в котором как свойства хранятся все локальные переменные (а также некоторая другая информация, такая как значение `this`).
2. Ссылка на *внешнее лексическое окружение* – то есть то, которое соответствует коду снаружи (снаружи от текущих фигурных скобок).

«Переменная» – это просто свойство специального внутреннего объекта: `Environment Record`. «Получить или изменить переменную», означает, «получить или изменить свойство этого объекта».



Замыкание – это функция, которая запоминает свои внешние переменные и может получить к ним доступ. В некоторых языках это невозможно. Но, в JavaScript, **все функции изначально являются замыканиями**. То есть функции автоматически запоминают, где были созданы, с помощью скрытого свойства `[[Environment]]`, и все они могут получить доступ к внешним переменным.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="counter.js"></script>
  <!-- <script src="multiply.js" type="module"></script> -->
  <!-- <script src="lexicalEnvironment.js"></script> -->
  <title>Замыкание - closure</title>
</head>
<body>

</body>
</html>
```

```
// counter.js
// Замыкание - closure

const log = console.log;

const createCounter = () => {
  let count = 0; // переменная будет доступна из функции ниже

  return () => {
    count += 1;
    return count;
  };
}

const counter = createCounter();

// debugger;
log(counter()); // 1
log(counter()); // 2
log(counter()); // 3
```

```
// multiply.js
// debugger;
// в дебаге можно увидеть что функция multiplyBy уже готова к использованию,
// в то время как double и triple еще <value unavailable>

// const channel = "Youtube";

function multiplyBy (factor) {
  // console.log(channel);

  return (number) => {
    return number * factor;
  };
}

const double = multiplyBy(2);
const triple = multiplyBy(3);

// debugger;
console.log(double(5)); // 10
console.log(triple(5)); // 15
```

```
// lexicalEnvironment.js
// Lexical environment - Лексическое окружение
// «Лексическое окружение» – это объект

const greeting = "Hello!";

const sayHello = (name) => {
  debugger; // Scope Local - лексическое окружение вызова
  console.log(`${greeting}, ${name}`);
}

sayHello("my friends!");
```

