Урок 3.7-3.8. Циклы, функции

Материалы

- Области видимости на Habr

Summary

Циклы

`while`

Цикл с условием. Условие проверяется до выполнения: если условие сразу не `true`, цикл не выполнится ни разу

```
isLoopContinues = true;
let i = 0;

while (isLoopContinues) {
    i = Math.random(); // случайное дробное значение от 0 до 1

    if (i > 0.5) {
        isLoopContinues = false;
    }
}
```

`do..while`

Цикл с постусловием. Условие проверяется после первого выполнения: цикл гарантированно выполнится один раз, вне зависимости от условия.

```
isLoopContinues = true;
let i = 0;

do {
    i = Math.random(); // случайное дробное значение от 0 до 1

    if (i > 0.5) {
        isLoopContinues = false;
    }
} while (isLoopContinues);
```

`for`

Цикл с известным числом повторений.

В конструкции объявления три части, разделённые точкой с запятой: инициализация счётчика, условие продолжения и действие изменения счётчика.

1. Инициализация счётчика выполняется один раз перед началом цикла
2. Затем проверяется условие
3. Потом выполняется тело цикла
4. Далее выполняется действие по изменению счётчика
5. Переходим к шагу 2.

```
for (let i = 0; i < 10; i++) {
    // do something
}
```

Функции

- Функции созданы для того, чтобы какую-то часть кода выполнять несколько раз, не дублируя код.
- Название функции должно отражать действие (`getUserPosts`, `sendResponse`, `writeMessage`)
- Функция должна выполнять одно осмысленное действие, которое можно объединить одним простым названием

Объявление функции

```
// Function Declaration
function sayHello() {
    console.log('Hello World');
}
```

```
// Function Expression
const sayHello = function () {
```

```
    console.log('Hello World');
}
```

```
// Function Expression (lambda - стрелочная)
const sayHello = () => {
    console.log('Hello World');
}
```

```
// IIFE - Immediately Invoked Function Expression
// Такая функция выполняется сразу
(function(name) {
    console.log(`Hello, ${name}`)
})('Имя');
```

Аргументы функции

В функцию можно передать данные: аргументы (или параметры). Они указываются в скобках через запятую.

```
function sayHelloToUser(userName) {
    console.log(`Hello, ${userName}`);
}

sayHelloToUser('Андрей'); // вызов функции с параметром
```

```
function sayHelloToUser(userName, userLastName) {
    console.log(`Hello, ${userName} ${userLastName}`);
}

sayHelloToUser('Андрей', 'Гулин'); // вызов функции с двумя параметрами
```

Параметры по умолчанию

Если функция содержит аргументы, но при вызове они не были указаны, то по умолчанию они равны `undefined`. Однако, можно задать значение параметра по умолчанию

```
function sayHello(userName = 'Guest') {
    console.log(`Привет, ${userName}`);
```

```
}

sayHello(); // параметр не указан, в консоль выведется 'Привет, Guest'
sayHello('Андрей'); // параметр указан, в консоль выведется 'Привет, Андрей'
```

Локальные и глобальные переменные

- Функция имеет доступ и может изменять переменные, объявленные в основном коде программы
- Переменные, созданные внутри функции, недоступны вне этой функции

Возвращаемое значение

Функция может вернуть значение в место кода, откуда она была вызвана

```
function getSumOfTwoNumbers(a, b) {
    return a + b;
}

// в консоль выведется возвращённое функцией значение
console.log(getSumOfTwoNumbers(1, 2));

// result присвоится возвращённое функцией значение
const result = getSumOfTwoNumbers(3, 4);
```