



# JavaScript

## Урок №11: Основы. Циклы.

Автор Бахшиллоев Бехзод

Ментор по Frontend в **PRO** **UNITY**



# Циклы.

При написании скриптов зачастую встает задача сделать однотипное действие много раз.

Например, вывести товары из списка один за другим. Или просто перебрать все числа от 1 до 10 и для каждого выполнить одинаковый код.

Для многократного повторения одного участка кода – предусмотрены *циклы*.



# Цикл while и do while

**while** (*условие*) {...}

Пока *условие* приводится к **true** цикл будет работать.

```
let i = 0;
while (i < 10) {
  console.log(i);
  i++;
}
// 0..9
```

```
let i = 0;
while (i++ < 10) {
  console.log(i);
}
// 1..10
```

```
let i = 0;
while (++i < 10) {
  console.log(i);
}
// 1..9
```

```
let i = 10;
while (i--) {
  console.log(i);
}
// 9..0
```

**do {...} while** (*условие*)

Сначала выполнится тело потом пройдет проверка условия.

```
let index;
while (index) {
  console.log(index);
}
```

```
do {
  console.log(index);
} while (index)
```



# Цикл for

**for** ([initialization]; [condition]; [iterator]) { ... }

**break** - останавливает цикл, **continue** - прерывает текущую итерацию и переходит к следующей

```
for (let i = 0; i < 10; i++) {  
  console.log(i);  
} // 0..9
```

поток выполнения цикла: **начало** → (если **условие** → **тело** → **шаг**) → (если **условие** → **тело** → **шаг**) → ... и так далее, пока верно условие.

**let i** объявленная при инициализации цикла недоступна за его пределами.

```
let i = 0;  
for ( ; i < 10; i++) {  
  console.log(i);  
} // 0..9
```

```
let i = 10;  
for ( ; i--; ) {  
  console.log(i);  
} // 9..0
```



# Цикл for in

Для перебора всех свойств из объекта используется цикл по свойствам **for..in** .

```
let list = {  
  one: 'text',  
  two: 'two text',  
}  
  
for (let item in list) {  
  console.log(item) // one, two ...  
  console.log(list[item]) // text, two text ...  
}
```

Порядок перебора соответствует порядку объявления для нечисловых ключей, а числовые – сортируются (в современных браузерах).



## Цикл for of

Новый цикл **for .. of** предназначен для итерации по элементам **коллекций (итерируемые объекты)**, но в отличие от цикла **for .. in** при итерациях используется значение, а не ключ.

```
let myArray = [1, 2, 3, 4];
```

```
for (let value of myArray) {  
  console.log(value); // 1, 2, 3, 4 ...  
}
```

**for-of** не только для массивов. Он также работает с большинством массивоподобных объектов, вроде списков `NodeList` в DOM и со строками.

## Задачи:

1. На основе строки *"i am in the easycode"* сделать новую строку где первые буквы каждого слова будут в верхнем регистре. Использовать **for** или **while**.
2. Дана строка *"tseb eht ma i"*. Используя циклы, сделать строку-перевертыш (то есть последняя буква становится первой, предпоследняя - второй итд).
3. Факториал числа - произведение всех натуральных чисел от 1 до n включительно:  $3! = 3*2*1$ ,  $5! = 5*4*3*2*1$ . С помощью циклов вычислить факториал числа 10. Использовать **for**.
4. На основе строки *"JavaScript is a pretty good language"* сделать новую строку, где каждое слово начинается с большой буквы, а пробелы удалены. Использовать **for**.
5. Найти все нечетные числа в массиве от 1 до 15 включительно и вывести их в консоль. Массив [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] Использовать **for of**.
6. Дан объект:  

```
let list = {  
  name: 'denis',  
  work: 'easycode',  
  age: 29  
}
```

Перебрать объект и если значение в свойстве это строка то переписать ее всю в верхнем регистре. Использовать **for in**.