

Функции высшего порядка

Цель работы

Получить навыки работы со встроенными функциями высшего порядка в JS.

Задания для выполнения

1. Сравните два варианта:

Решение задачи без использования функций высшего порядка

```
const arr1 = [1, 2, 3];
const arr2 = [];
for(let i = 0; i < arr1.length; i++) {
  arr2.push(arr1[i] * 2);
}
// выводит [ 2, 4, 6 ]
console.log(arr2);
```

Решение задачи с помощью функции высшего порядка map

```
const arr1 = [1, 2, 3];
const arr2 = arr1.map(function(item) {
  return item * 2;
});
console.log(arr2);
```

Объем этого кода можно даже сократить, если воспользоваться стрелочной функцией:

```
const arr1 = [1, 2, 3];
const arr2 = arr1.map(item => item * 2);
console.log(arr2);
```

Воспользовавшись методом map(), создайте новый массив из имеющегося, который содержит увеличенные элементы на 10 и в три раза.

2. Имеется массив «год рождения». Необходимо вывести их возраст.

3. Предположим, у нас имеется массив, содержащий объекты, в свойствах которых хранятся сведения об имени и возрасте представителей некой группы людей. Нам надо создать массив, в котором будут сведения только о совершеннолетних представителях этой группы (тех, чей возраст достиг 18 лет).

Решение задачи без использования функций высшего порядка

```
const persons = [
  { name: 'Peter', age: 16 },
  { name: 'Mark', age: 18 },
  { name: 'John', age: 27 },
  { name: 'Jane', age: 14 },
  { name: 'Tony', age: 24 },
];
const fullAge = [];
for(let i = 0; i < persons.length; i++) {
  if(persons[i].age >= 18) {
    fullAge.push(persons[i]);
  }
}
console.log(fullAge);
```

Решение задачи с помощью функции высшего порядка *filter*

```
const persons = [
  { name: 'Peter', age: 16 },
  { name: 'Mark', age: 18 },
  { name: 'John', age: 27 },
  { name: 'Jane', age: 14 },
  { name: 'Tony', age: 24},
];
const fullAge = persons.filter(person => person.age >= 18);
console.log(fullAge);
```

Используя данный пример, реализуйте проверку по условию на право редактирования сайта и вывод имени админа.

4. Используя синтаксис:

```
const sum = arr.reduce(function(accumulator, currentValue) {
  return accumulator + currentValue;
});
```

Вычислите средний балл студентов.

5. Используя синтаксис, создайте собственную функцию высшего порядка, которая из заданного массива возвращает массив, состоящий из нулей (для тех элементов длинна которых меньше либо равна 3) и единиц (в другом случае).

```
const strArray = ['JavaScript', 'Python', 'PHP', 'Java', 'C'];

function mapForEach(arr, fn) {
  const newArray = [];
  for(let i = 0; i < arr.length; i++) {
    newArray.push(
      fn(arr[i])
    );
  }
  return newArray;
}

const lenArray = mapForEach(strArray, function(item) {
  return item.length;
});
// выводит [ 10, 6, 3, 4, 1 ]
console.log(lenArray);
```

Методические указания

Функцией **высшего порядка** называется такая функция, которая принимает функцию как аргумент или возвращает функцию в виде выходного значения. Например, встроенные функции JavaScript `Array.prototype.map`, `Array.prototype.filter` и `Array.prototype.reduce`.

Метод **map()** создаёт новый массив, вызывая, для обработки каждого элемента входного массива, коллбек, переданный ему в виде аргумента. Этот метод берёт каждое возвращённое коллбеком значение и помещает его в выходной массив.

Метод **filter()** создаёт, на основе массива, новый массив, в которой попадают элементы исходного массива, соответствующие условию, заданному в переданной этому методу функции обратного вызова. Эта функция принимает, как и в случае с методом **map()**, 3 аргумента: **element**, **index** и **array**.

Метод **reduce()** обрабатывает каждый элемент массива с помощью коллбэка и помещает результат в единственное выходное значение. Этот метод принимает два параметра: коллбэк и необязательное начальное значение (**initialValue**).

Контрольные вопросы

1. Функция `Array.prototype.map` какого порядка?
2. Проанализируйте код. Является ли функция `lenArray` высшего порядка?

```
const strArray = ['JavaScript', 'Python', 'PHP', 'Java', 'C'];

function mapForEach(arr, fn) {
  const newArray = [];
  for(let i = 0; i < arr.length; i++) {
    newArray.push(
      fn(arr[i])
    );
  }
  return newArray;
}

const lenArray = mapForEach(strArray, function(item) {
  return item.length;
});
// выводит [ 10, 6, 3, 4, 1 ]
console.log(lenArray);
```

Дополнительные задания

1. Организовать симулятор в виде функции высшего порядка, который выдает только три случайных значения – красный, черный и белый (0, 1, 2). Запустить симуляцию 1000 000 раз. Узнать, какая последовательность из красных, черных или белых значений была самой короткой.

Дополнительные материалы

<https://habr.com/ru/company/ruvds/blog/428570/>