

# JavaScript: массивы

---

**WEB  
COURSE  
ORT DNIPRO**

---

**ORTDNIPRO.ORG/WEB**

# 1. Массивы



## Массивы / Arrays

**Массив** – это нумерованный набор значений (по сути много переменной с номерками внутри одной «большой»).

# Массивы / Arrays

```
1
2   let array = ['Jhon', 42, true, 3.14, 'Hello world'];
3   //           0      1      2      3      4
4
5   console.log( array[4] );
6
7   array[4] = 'New Text';
8
9   console.log( array[4] );
10
```

**Массив** – это нумерованный набор значений  
(по сути - много переменных, с номерками, внутри одной  
«большой» переменной).

```
1 let array = ['Jhon', 'Maria', 'Anna', 'Peter'];
2
3 console.log('Initial Length:', array.length); //4
4
5 array.shift();
6
7 console.log('Length After Shift:', array.length); //3
8
9 array.pop();
10
11 console.log('Length After Pop:', array.length); //2
12
13 array.push('Jane');
14
15 console.log('Length After Push:', array.length); //3
16
17 array.unshift('Alice');
18
19 console.log('Length After Unshify:', array.length); //4
20
21 console.log(array);
22
23
```

## Массивы / Arrays

**Массив**, в JavaScript, динамическая структура данных. У массива есть понятие длины (которую в любой момент можно узнать через свойство **array.length**). К ним можно добавлять элементы при помощи методов **array.push()** и **array.unshift()**, или же удалять, задействовав **array.pop()** и **array.shift()**.

## 2. Цикл `for` и массивы

# Цикл **for** и перебор массива

```
1
2   let array = ['Jhon', 'Maria', 'Anna', 'Peter'];
3
4   for(let i = 0; i < array.length; i++){
5
6       console.log(`User #${i}: ${array[i]}`);
7
8   }
9
```

Цикл **for** удобен для тех случаев, когда заранее известно (или можно просчитать на основе уже имеющихся данных), сколько раз нужно будет повторить то или иное действие. Например: **перебор и обработка элементов массивов**.

# 3. Цикл for-of



# Цикл for-of – перебор элементов массива

```
1
2   let array = ['Jhon', 'Maria', 'Anna', 'Peter'];
3
4   for(let item of array){
5
6       console.log(`User ${item}`);
7
8   }
9
```

Цикл **for-of** удобен для тех случаев, когда нам необходимо перебрать все элементы массива, при этом их нумерация нам не важна.

## 4. Алгоритмы обработки набора данных

## Немного практики: базовые алгоритмы работы с данными

```
3  
4  var usd = [26.4, 24.54, 26.08, 24.06, 25.27, 25.24, 24.61,  
26.81, 24.12, 25.8, 25.5, 24.59, 26.67, 26.74, 25.79, 24.28,  
26.12, 26.58, 24.27, 24.36, 24.97, 25.51, 25.23, 26.33, 26.  
5  7, 24.4, 25.25, 26.98, 25.51, 24.49];
```

Шаблон с данными в репозитории занятия:

[./src/array-template](#)

# Базовые алгоритмы работы с данными

1. Поиск максимального (минимального) элемента, среднего значения;
2. Создание нового набора данных (массива) на основе имеющегося;

...

*В некоторых задачах взаиморасположение элементов может влиять на результат, а в некоторых нет.*

# Базовые алгоритмы работы с данными

*Поиск максимального (минимального) элемента  
и среднеарифметического значения*

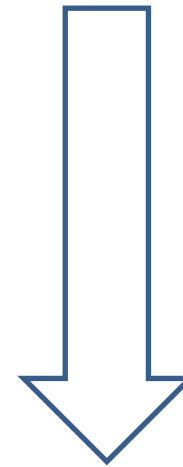
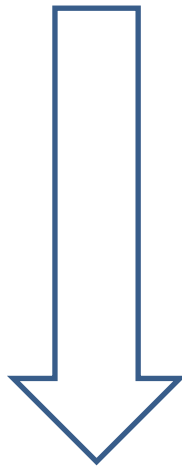
[62, 64, 81, 16, 74, 11, 56, 47, 63, 99]



# Базовые алгоритмы работы с данными

*Создание нового набора данных на основе имеющегося набора*

[4, 99, 65, 40, 66, 65, 92, 24, 4, 26]



[59, 35, 12, 17, 6, 20, 6, 29, 18, 77]

# Домашнее задание

# Домашнее задание «Заморозки» #B1

```
//Массив хранит значение температуры за 90 дней зимы
```

```
let winterDays = [-48, -46, 48, 27, -20, -35, 43, 4, 9, 10, 41, -46, -4, 0, -38,  
-49, 25, -46, -48, -23, -25, -22, 12, 38, 19, -20, 26, 4, 19, 23, 26, -41, 4, -13,  
-9, -11, -7, 38, 27, 41, 14, -35, -38, -44, -44, -22, -24, 29, -32, 41, 7, -25, 3,  
27, -45, 10, 48, 8, -34, -49, 17, -16, 41, -11, -50, -6, -34, 20, 14, -18, 39,  
-28, -33, -27, -48, 40, -37, -44, 0, 46, 36, -34, -50, 8, -3, 26, 40, 10, -36, 24];
```

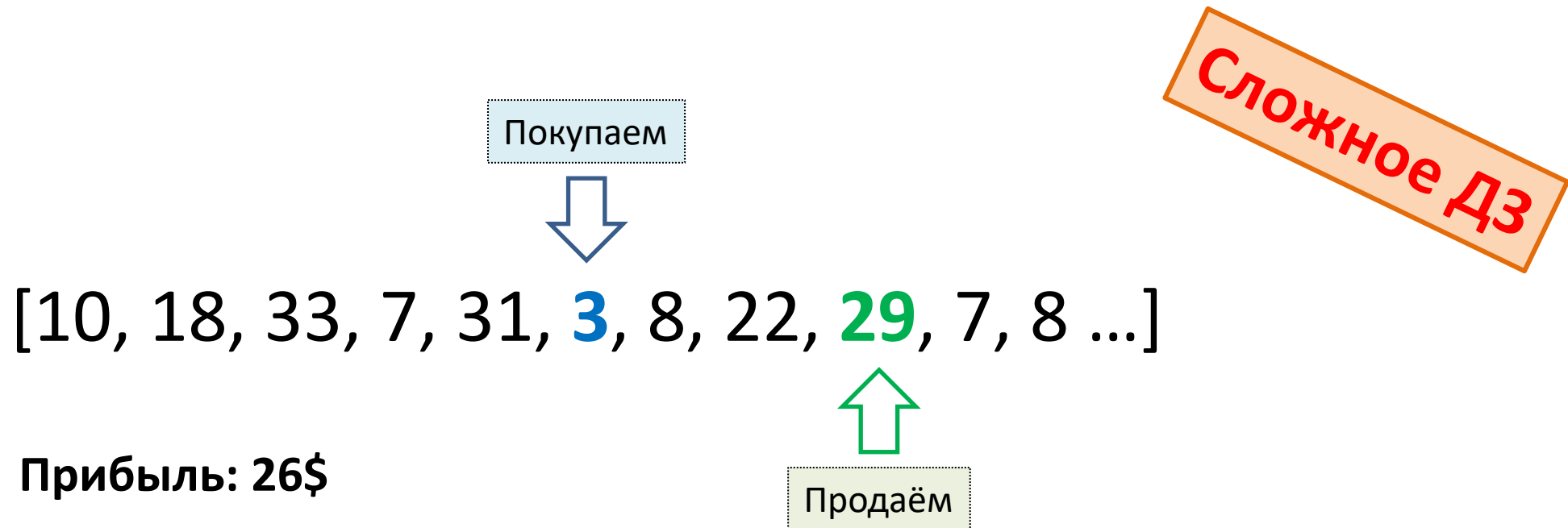
**Задача:** в массиве содержатся данные о ежедневной температуре в течении зимних месяцев. Необходимо подсчитать **сколько было замерзаний** в этот период (замерзания это когда в какой-то день температура плюсовая, а на следующий минусовая). **Сложное задание:** определить какой **самый длинный** по продолжительности был **период заморозков** (сколько дней подряд была температура ниже нуля) .

Шаблон с данными в репозитории занятия:

[./src/homework-template](#)



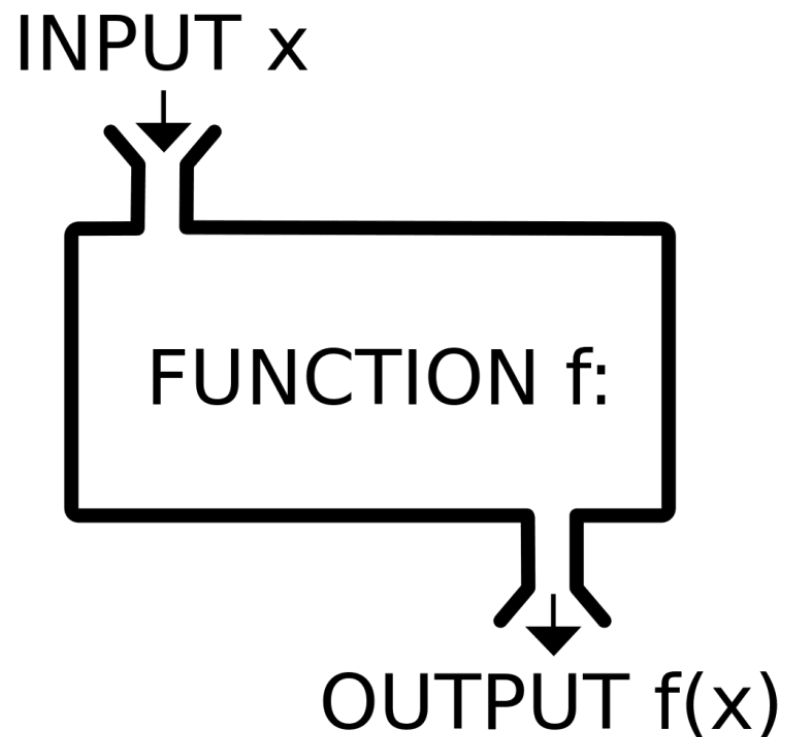
# Домашнее задание #B2



**Задача:** в массиве содержатся данные о ежедневной цене биткоина (**длина массива и его содержимое может меняться**), за какое-то количество дней. Скрипт должен **рассчитать какую максимальную прибыль** (и сообщить её пользователю) можно получить если сначала купить “биткоин”, а затем продать его (именно в такой последовательности, продать раньше чем купить нельзя). Можно совершить только одну покупку и одну продажу. **Цикл внутри цикла** поможем справиться с этой задачей.

**К следующему  
занятию...**

# О Функциях



Предварительные знания – лучший помощник в обучении, поэтому к следующему занятию жду, что **посмотрите небольшой ролик о функциях.**

<https://youtu.be/g4F2DLtPVv4>