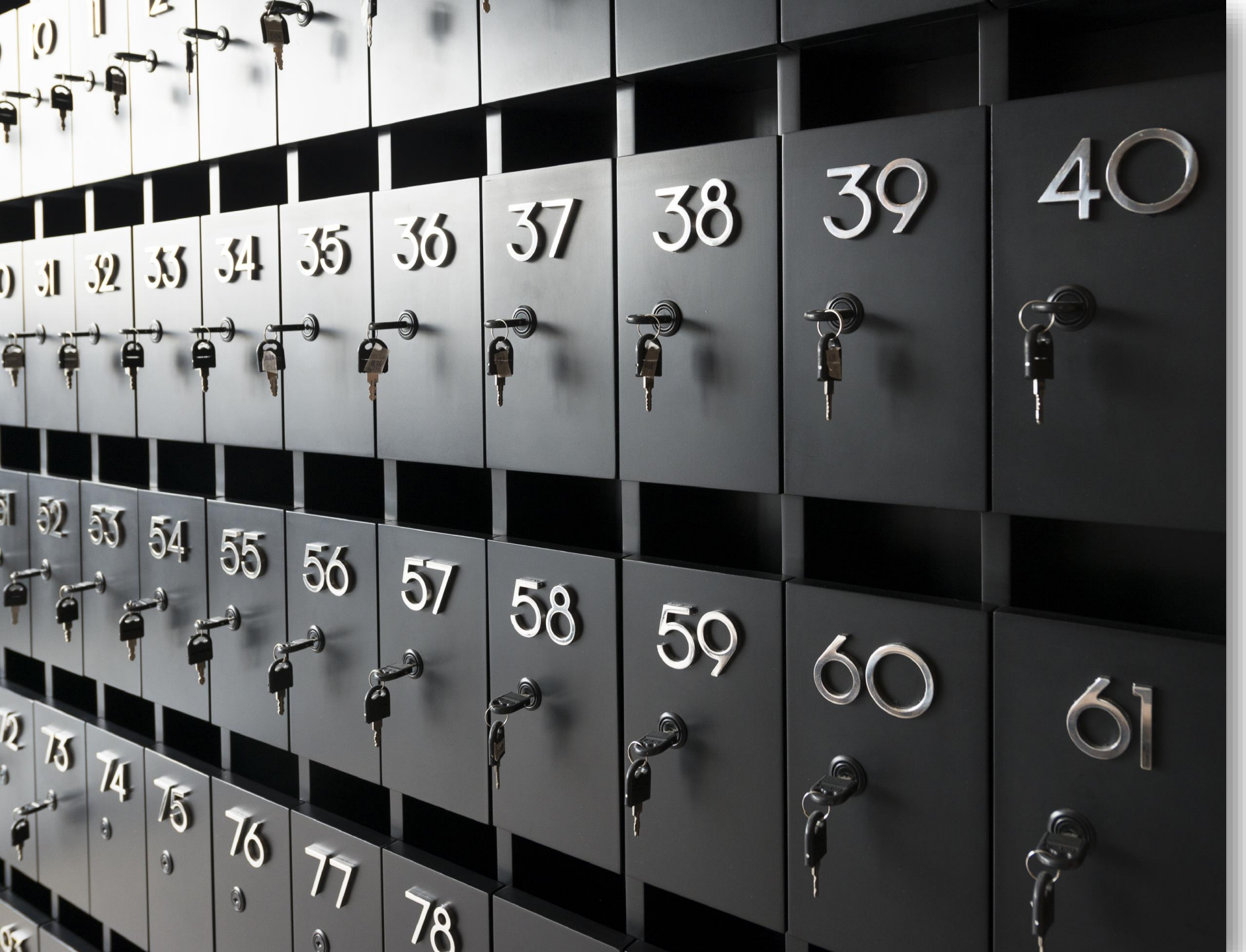
JavaScript:
Arrays

dimon.work/kurs.html

1. Arrays



Arrays

Array – to numerowany zestaw wartości (zasadniczo wiele zmiennych z liczbami w ramach jednej „dużej” zmiennej).

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
```

Array – to ponumerowany zestaw wartości (zasadniczo wiele zmiennych z numerami w ramach jednej „dużej” zmiennej).

```
let array = ['Jhon', 'Maria', 'Anna', 'Peter'];

console.log('Initial Length:', array.length); //4

array.shift();

console.log('Length After Shift:', array.length); //3

array.pop();

console.log('Length After Pop:', array.length); //2

array.push('Jane');

console.log('Length After Push:', array.length); //3

array.unshift('Alice');

console.log('Length After Unshify:', array.length); //4

console.log(array);
```

Arrays

Array w **JavaScript** jest dynamiczną strukturą danych. Tablica ma pojęcie długości (którą można sprawdzić w dowolnym momencie za pomocą właściwości **array.length**). Możesz dodawać do nich elementy za pomocą metod **array.push()** i **array.unshift()** lub usuwać je za pomocą **array.pop()** i **array.shift().**

2. Cykl for i arrays

Cykl for i przeszukiwanie array

```
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
```

Cykl **for** jest wygodny w przypadkach, gdy z góry wiadomo (lub można obliczyć na podstawie już dostępnych danych), ile razy należy powtórzyć określoną czynność. Na przykład: **wyszukiwanie i przetwarzanie elementów**.

3. Cykl for-of

Cykl for-of – przeszukiwanie elementów array

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

Cykl **for-of** jest wygodny w przypadkach, gdy musimy przejrzeć wszystkie elementy tablicy, a ich numeracja nie jest dla nas ważna.

4. Algorytmy przetwarzania zbioru danych

Trochę praktyki: podstawowe algorytmy pracy z danymi

```
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.
        7, 24.4, 25.25, 26.98, 25.51, 24.49];
5
```

Szablon za linkiem:
dimon.work/kurs/les20.7z

Podstawowe algorytmy pracy z danymi

1. Wyszukiwanie maksymalnego (minimalnego) elementu, wartości średniej;
2. Stworzenie nowego zestawu danych na podstawie istniejącego;

...

W niektórych zadaniach wstawienie elementów może wpłynąć na wynik, a w niektórych nie.

Podstawowe algorytmy pracy z danymi

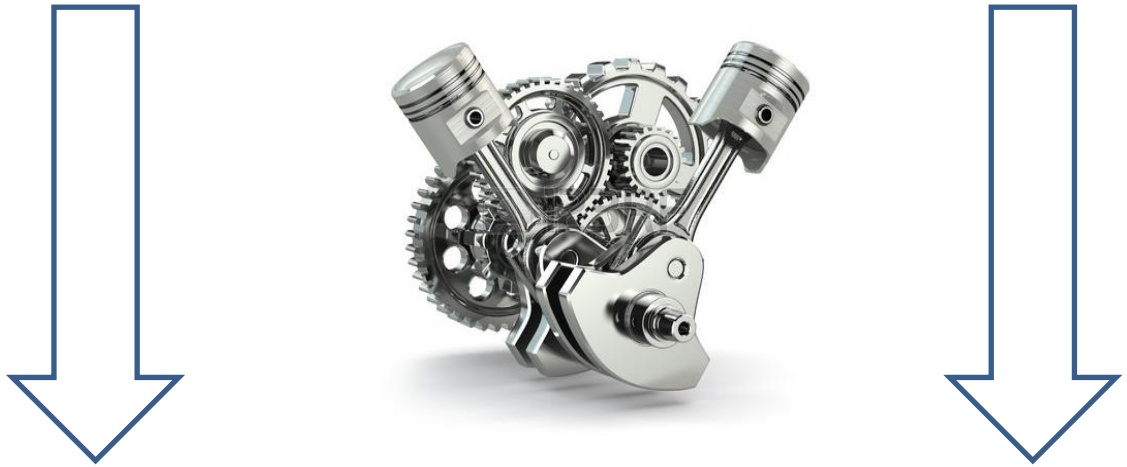
Wyszukiwanie elementu maksymalnego (minimalnego) i średniej arytmetycznej

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

Podstawowe algorytmy pracy z danymi

Tworzenie nowego zbioru danych na podstawie istniejącego

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



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

Zadanie

Zadanie „ przymrozek” #B1

```
2
3       //Массив хранит значение температуры за 90 дней зимы
4
5       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];
6
```

Zadanie: tablica zawiera dane o dziennych temperaturach w miesiącach zimowych. Konieczne jest obliczenie, **ile było przymrozków** w tym okresie (przymrozki występują, gdy temperatura jest dodatnia pewnego dnia i ujemna następnego dnia).

Trudne zadanie : Określ **najdłuższy** okres mrozu (ile dni z rzędu temperatura wynosiła poniżej zera).

Szablon z danymi:

dimon.work/kurs/les20.7z

JS: funkcje i zdarzenia