

TypeScript

Tablice tworzymy tak samo jak w JS.

```
//Różne sposoby tworzenia tablic
```

```
let tab:string[] = [];
```

```
tab.push("Anna"); //dodanie na końcu tablicy nowego elementu
```

```
let auta = new Array("Volvo","Polonez","Fiat");
```

```
let imiona = ["jan","maria","karolina"];
```

Na każdej tablicy można wykonać różne operacje:

Metoda	Opis
concat(nazwa_dodłączanej_tablicy)	Po użyciu metody zostanie zwrócona nowa tablica, która powstaje w wyniku połączenia tablicy z tą podaną w argumencie (można łączyć ze sobą kilka tablic)
join(separator)	Zwracany jest ciąg tekstowy wszystkich wartości przechowywanych w tablicy; poszczególne elementy są od siebie oddzielone separatorem
pop()	Metoda usuwa i zwraca ostatni element w tablicy
push(element)	Metoda dodaje określony element na końcu tablicy
reverse()	Metoda zwraca tablicę, której elementy są posortowane w kolejności malejącej
shift()	Metoda usuwa i zwraca pierwszy element w tablicy
slice(start, stop)	Metoda zwraca fragment tablicy; start to indeks początku, stop zaś to indeks końca
sort()	Metoda zwraca tablicę, której elementy są posortowane w kolejności rosnącej
splice(indeks, liczba)	Metoda usuwa podaną liczbę elementów od wskazanego indeksu
unshift(element)	Metoda powoduje wstawienie nowego elementu na początku tablicy

Można też sprawdzić czy dana zmienna jest tablicą:

```
let imiona = ["jan","maria","karolina"];
```

```
Array.isArray(imiona);
```

Zadania

Do zadania 2.5 – operacje wykonujemy po naciśnięciu odpowiedniego przycisku (usuń, dodaj itp)

Zadanie 2.5.

Utwórz tablicę zawierającą owoce: jabłko, arbuz, banan, gruszka.

- a. Zwróć długość tablicy.
- b. Zwróć posortowaną tablicę.
- c. Dodaj na koniec tablicy nową wartość ananas.
- d. Usuń pierwszy element tablicy.
- e. Usuń ostatni element tablicy.
- f. Dodaj jako pierwszy element tablicy nową wartość ananas.
- g. Odwróć kolejność elementów tablicy.

Zadanie 2.6.

Za pomocą pętli for wyświetl zawartość tablicy z zadania 2.5.

Do szybkiego łączenia ze sobą tablic możemy użyć **operatora rozwinięcia** (ang. *spread operator*), który ma postać trzech kropek (...).

Użycie operatora **spread** jest pokazane na listingu 2.20.

Listing 2.20. Operator rozwinięcia

```
let koloryCiepłe: string[] = ['czerwony', 'pomarańczowy', 'żółty'];
let koloryZimne: string[] = ['zielony', 'niebieski', 'szary'];

let kolory = [...koloryCiepłe, ...koloryZimne];
```

Zadanie 2.7.

Wykorzystaj tablicę z zadania 2.5 i dodatkowo utwórz kolejną, zawierającą warzywa: marchew, burak, pietruszka, kalafior.

- a. Połącz obie tablice z użyciem metody `.concat()`.
- b. Sprawdź, czy tablice można do siebie dodać za pomocą znaku dodawania.
- c. Połącz obie tablice z użyciem operatora rozwinięcia.

Na celujący:

Zamiana wyrazów w tablicy w taki sposób aby co druga literka w wyrazie była pisana z dużej np.:
jabłko → JaBłKo