

TypeScript

FUNKCJE:

Ten zapis jest bardzo podobny do metod z Javy.

```
//Funkcja przyjmuje dwa parametry i zwraca wartość
function suma(a:number, b:number) : number{
    return a+b;
}
console.log(suma(6,7)); //Użycie funkcji
```

```
//Funkcja jako wyrażenie
const sumaw = function (a:number, b:number) : number{
    return a+b;
}
//Użycie:
console.log(sumaw(6,7));
```

```
//Ustawienie wartości domyślnej parametru funkcji:
function poleKola(r:number, pi:number = 3.14):number{
    return r*r*pi;
}
//Użycie:
console.log(poleKola(10));
//lub:
console.log(poleKola(10,3.1415)); //większa dokładność ale uwaga na wynik!!!
```

```
//PARAMETR OPCJONALNY
function daneOsobowe(s:string,n?:number):string{ //ZNAK ZAPYTANIA!!!
    return `Witaj ${s} masz ${n} lat!`; //UWAGA apostrof przy tyldzie
}
//Użycie:
console.log(daneOsobowe("Janek"));
//lub
console.log(daneOsobowe("Janek",30));
```

```
//Wiele parametrów
//trzy kropki - wtedy parametry jako tablica
function spisOsob(...imiona:string[]):string{
    return `Witajcie: ${imiona}`;
}
//Użycie:
console.log(spisOsob("Janek","Ryszard","Karolina")); //dowolna ilość parametrów

//parametr resztowy
//... umieszczamy jako ostatni parametr funkcji
//funkcja nie zwraca wartości
function spisDanych(imie: string, nazwisko: string, ...dane:string[]){
    let temp = imie+` `+nazwisko+` -> `+dane.join(" | ");
    console.log(temp);
}
//użycie
spisDanych("Anna","Mucha","studentka","3 rok","rak");
```

Funkcja anonimowa

W programowaniu **funkcja anonimowa** to funkcja, która nie ma przypisanej nazwy. Funkcje anonimowe są często **argumentami przekazywanymi innym funkcjom**. Najczęściej są używane do operacji, które nie powtarzają się w wielu miejscach, przez co nie chcemy definiować osobnej funkcji.

Przyjrzyjmy się programowi, którego kod jest pokazany na listingu 2.33.

Listing 2.33. Funkcja jako argument

```
function powitanie(imie: string): string {
    return `Witaj, ${imie}`;
}

const bazaImion: string[] = ["Jan", "Paweł", "Mariusz"];
console.log(bazaImion.map(powitanie));
```

Funkcja powitanie ma jeden parametr, imie, który jest zwracany po jej wywołaniu wraz z napisem witaj. Ta funkcja jest argumentem metody map() wykonanej na tablicy bazaImion. Takie wywołanie spowoduje wykonanie funkcji na każdym elemencie tablicy (bądź przekazanie każdego elementu tablicy bazaImion jako argumentu funkcji powitanie).

Wynikiem wywołania kodu jest wyświetlenie:

```
[ 'Witaj, Jan', 'Witaj, Paweł', 'Witaj, Mariusz' ]
```

Ten sam efekt uzyskamy za pomocą kodu przedstawionego na listingu 2.34.

Listing 2.34. Funkcja anonimowa

```
const bazaImion: string[] = ["Jan", "Paweł", "Mariusz"];
console.log(bazaImion.map(function (imie: string): string { return `Witaj, ${imie}`; }));
```

Funkcja jest definiowana bezpośrednio w metodzie map(). Funkcja ta nie ma nazwy, dlatego jest określana jako anonimowa.

```
//Funkcja strzałkowa
const sumas = (a:number, b:number) => (a+b);
/*
Powyższe to to samo co niżej
const sumas = sumaf(a:number, b:number) : number{
    return a+b;
}
*/
//Użycie:
console.log(`Suma liczb ${sumas(9,10)}`);
```

```
//Funkcja strzałkowa oraz użycie metody map
const liczby: number[] = [2,4,6];
const potega = (a:number) => (a**2);
console.log(`Liczby: ${liczby} do potęgi: ${liczby.map(potega)}`);
```

Metoda **map** pobiera wszystkie elementy z tablicy i wstawia po kolei do funkcji, następnie zwraca wynik jako tablicę.

W powyższym przykładzie:

Bierze 2 wstawia do funkcji **potega** funkcja podnosi ją do potęgi drugiej a następnie zapisuje do nowej tablicy.

I tak z każdą liczbę po kolei. Na końcu otrzymuje tablicę z liczbami podniesionymi do potęgi.

ZADANIA na następnej stronie.

Zadania

Zadanie 2.8.

Użyj funkcji strzałkowej!

Napisz funkcję, która obliczy i wyświetli w konsoli różnicę, iloczyn i iloraz dwóch liczb podanych przez użytkownika.

Zadanie 2.9.

Napisz funkcję, która wyliczy i wyświetli wartość bezwzględną z zadanej liczby.

Zadanie 2.11.

Sprawdź, w jaki sposób działa metoda `.filter()`. Masz daną tablicę: `const numery = [2, 4, 7, 11, 14, 19, 21, 100]`. Napisz funkcję, która wypisze w konsoli dwie tablice: pierwszą — z liczbami parzystymi, a drugą — z liczbami nieparzystymi.

Zadanie 2.12.

Napisz funkcję, do której jako argument przekażesz tablicę zawierającą listę osób. Zadaniem funkcji jest wylosowanie osoby. Tablica do wywołania funkcji to: `const osoby = ["Marcin", "Jan", "Beata", "Tadeusz", "Teresa"]`.

Zadanie 2.14:

Stwórz tablicę składającą się ze 100 losowych liczb z zakresu 1 – 100 użyj funkcji zwracającej tę tablicę, jako parametr funkcja przyjmuje ile liczb ma wylosować, domyślnie funkcja losuje 100 liczb.

Stwórz funkcję, która jako parametr przyjmuje tablicę, w której wyszukuje liczby podzielne przez 5. Wynik działania funkcji wyszukującej to:

Otrzymałem tablicę z 20 liczbami, gdzie najmniejsza liczba to 4 a największa 80 w tym liczby podzielne przez 5 to: 10,20,80 czyli 15% wszystkich liczb.

Ilość zrobionych zadań to ocena. Ostatnie zadanie jest traktowane jako dwie oceny.