

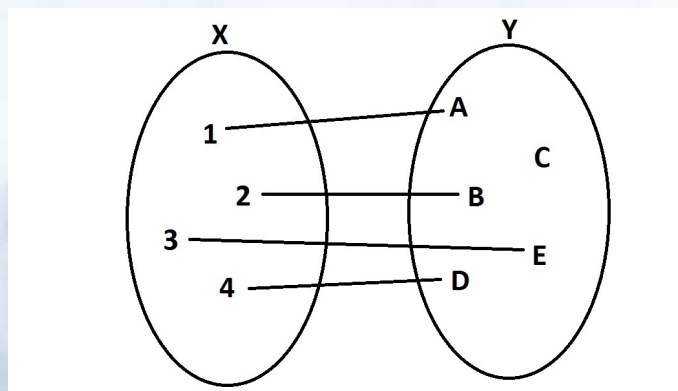
A misty forest landscape with tall evergreen trees and low-lying clouds. The scene is atmospheric and serene, with the trees partially obscured by the fog. The overall color palette is muted, with various shades of blue, grey, and green.

# Funkcje

Małgorzata Tokarek

# Pojęcie i nazwa funkcji

Funkcja czyli odpowiednik matematycznej funkcji jest to obiekt, który dla tego samego argumentu zawsze zwraca tą samą wartość. Każda funkcja ma swoją nazwę i listę argumentów w nawiasach okrągłych.



$f(x)$

$mF(x)$

# Wartość funkcji

Odpowiednikiem wartości  $y$  gdy funkcja przyjmuje wartość  $x$  będzie w programowaniu wartość zwraca przez funkcję:

$$y=f(x)$$

let  $y= mF(x)$

# Wartość funkcji cd.

Wartość przyjęta/zwraca przez funkcję może być używana przez inną funkcję/metodę. W poniższych przykładach:

- wartość funkcji  $f$  jest argumentem funkcji  $g$

$$y = g(2 * f(x))$$

- Wartość zwracana przez funkcję  $mF1$  jest argumentem metody(funkcji)

`document.write`

`document.write(mF(x))`



# Przykłady definicji funkcji

➤ W matematyce:

$$f(x)=x+2 \quad x \in R$$

➤ W JavaScript:

```
function fM(x){  
    return x+2;  
}
```

# Liczba wartości zwracane przez funkcję

W JavaScript funkcja może nie zwracać żadnej wartości:

```
<script>  
    function funkcja1(){  
        document.write("Ala ma kota");  
    }
```

# Wywołanie funkcji w JS

Poniższy kod pokazuje 3 sposoby wywołania funkcji (wydania polecenia wykonania się funkcji)

```
<button onclick="funkcja1()">Funkcja 1</button>
<br><br>
<input type="button" onclick="funkcja1()" value="Funkcja 1">
<br><br>
<script>
  function funkcja1(){
    document.write("Ala ma kota");
  }
  funkcja1();
```

Funkcja 1

Funkcja 1

Ala ma kota

# Funkcje zwracająca wartość

```
function funkcja1(){  
    return "Ala ma kota";  
}  
function funkcja2() {  
    let a=Number(prompt("Podaj a: ") );  
    return 2*a;  
}  
document.write("Funkcja1 zwraca: "+funkcja1());  
document.write("<br>");  
document.write("Funkcja2 zwraca: "+funkcja2());
```

Funkcja1 zwraca: Ala ma kota  
Funkcja2 zwraca: 12



# Pytania

Pytanie 1: Jakim słowem kluczowym rozpoczynamy pisanie definicji funkcji.

Pytanie 2: W jaki sposób wywołujemy funkcję, która nic nie zwraca

Pytanie 3: W jaki sposób wywołujemy funkcję, która coś zwraca

Pytanie 4: Podaj za pomocą jakiego atrybutu

# Zadania

Do każdego z poniższych zadań utwórz przycisk z napisem Zadanie x, gdzie x jest numerem zadania. Przyśnięcie przycisku uruchamia funkcję, która jest realizacją danego zadania.

Zadanie 1: Napisz funkcję, która wyświetla na ekranie Twoje imię

Zadanie 2: Napisz funkcję, która wczytuje z okna dialogowego dwie liczby i wypisuje na ekranie ich sumę.

Zadanie 3: Napisz funkcję, która wczytuje z okna dialogowego dwie liczby, wykonuje na nich podstawowe działania arytmetyczne, a ich wyniki wypisuje na ekranie. Uwaga, jeśli się nie da to nie próbować dzielenia należy tylko wyświetlić na ekranie stosowany komunikat.

# Argumenty z formularza - *getElementById*

W zestawach egzaminacyjnych z JS, na stronie z przydatnymi metodami, pierwsza tabelka zawiera interesujące nas metody obiektu document.

Do pobierania informacji z formularza przydaje się metoda *document.getElementById*

## Wybrane pola i metody modelu DOM języka JavaScript

### Wyszukiwanie elementów

```
document.getElementById(id)  
document.getElementsByTagName(TagName)  
document.getElementsByClassName(ClassName)  
document.getElementsByName(ElementName)
```

### Zmiana elementów

```
element.innerHTML = "nowa zawartość"  
element.attribute_name = "nowa  
wartość"  
element.setAttribute(atrybut,  
wartosc)  
element.style.property_name = "nowa  
wartość"
```



# *document.getElementById(„identyfikator”)*

Jak nazwa wskazuje metoda ta odszukuje w dokumencie i daje dostęp do elementu o podanym id.

Ale tylko do elementu.

Jeśli chcemy coś zrobić z tym elementem to musimy, to musimy to odpowiedzieć. Np. gdy chcemy uzyskać dostęp do wartości danego elementu to piszemy *document.getElementById(„identyfikator”).value*

Ponadto pobraną wartość z elementu możemy poddać „obróbce”

```
let b=Number(document.getElementById("liczba2").value);
```



# Pobieranie wartości z formularza

Można też rozbić to na dwie zmienne, co dla niektórych może być bardziej czytelne.

```
let element=document.getElementById("liczba1");  
let a=Number(element.value);
```

Tak wprowadzoną zmienną możemy używać w dalszej części skryptu.

# Przekazywanie informacji ze skryptu do strony

Zależy nam aby efekty pracy skryptu były widoczne w konkretnym miejscu na stronie:

*Iloczyn liczb 6 i 0 wynosi 0*

# Przekazywanie informacji ze skryptu do HTML

**innerHTML** – jest to właściwość elementu umożliwia odczyt, zmodyfikowanie, wstawienie, albo usunięcie kodu HTML Jaki jest we wnętrzu danego elementu DOM.

Umiemy już uzyskiwać dostęp do elementów mających id, a zatem w połączeniu z tą właściwością możemy przesyłać do HTML

# Wstawienie informacji ze skryptu do HTML

Pod formularzem wstawiono pusty akapit (`<p>` - paragraf) mający *id*, do którego zawartość wstawimy w JS korzystając z *innerHTML*

```
<form>
  <input type="number" id="liczba1">
  <input type="number" id="liczba2">
  <br><br>
  <input type="button" value=" + " onclick="dodawanie()">
  <input type="button" value=" - " onclick="odejmowanie()">
  <input type="button" value=" * " onclick="mnozenie()">
  <input type="button" value=" / " onclick="dzielenie()">
</form>
<p id="idWynik"></p>
```



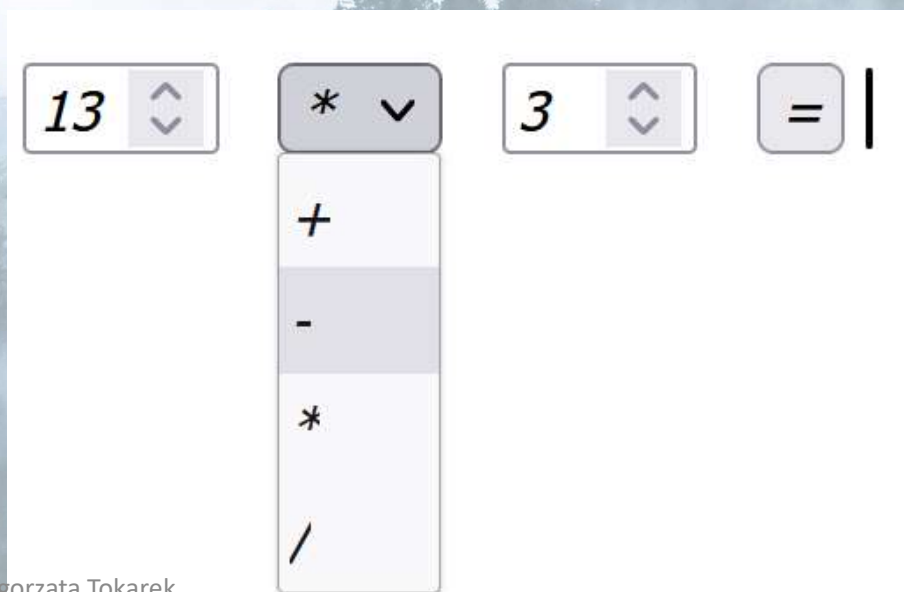
# Wstawienie informacji ze skryptu do HTML

W skrypcie użyto

```
function dodawanie(){  
    let element1=document.getElementById("liczba1");  
    let a=Number(element1.value);  
    let element2=document.getElementById("liczba2");  
    let b=Number(element2.value);  
    // wynik to element o id wynoszącym "idWynik"  
    let wynik=document.getElementById("idWynik");  
    wynik.innerHTML="Suma liczb "+a+" i "+b+" wynosi "+ (a+b);  
}
```

# Lista rozwijalna czyli pole *select* w formularzu

Jeśli go chcemy przekazać do formularza to jak innym polom musimy mu nadać *id*. Odpowiednio kolejne opcje muszą mieć *value*, które będzie pobierane w JS



The image shows a calculator interface with a dropdown menu for the multiplication operator. The interface includes two input fields with values 13 and 3, a dropdown menu with options +, -, \*, and /, and an equals sign button. The dropdown menu is currently open, showing the multiplication operator selected.

# Lista rozwijalna - implementacja

```
<form>
  <input type="number" id="liczba1" class="doLiczba">
  <select id="znak" class="doPrzycisku">
    <option value="+> + </option>
    <option value="-> - </option>
    <option value="*> * </option>
    <option value="/> / </option>
  </select>
  <input type="number" id="liczba2" class="doLiczba">
  <input type="button" value=" = " onclick="oblicz()" class="doPrzycisku">
  <span id="idWynik"></span>
</form>
```

# Wczytanie wartości z listy rozwijalnej

Odbywa się ono jak dla innych pól. Potrzebujemy znać *id* danego pola i odczytywać jego *value*:

```
let elemZnak=document.getElementById("znak");  
let znak=elemZnak.value;
```



# Instrukcja wyboru

W zależności od wartości zmiennej znak w różny sposób modyfikujemy zawartość elementu „*idWynik*”

```
switch (znak){  
    case "+":  
        wynik.innerHTML=(a+b);  
        break;  
    case "-":  
        wynik.innerHTML=(a-b);  
        break;  
}
```

*document.getElementById(„ „).style*

W JS możemy też modyfikować wygląd wskazanego elementu.

A visual representation of a calculator interface. It consists of a sequence of input fields and operators. The first field contains the number '13' and has up/down arrow icons to its right. This is followed by a minus sign '-' and a dropdown arrow 'v'. The next field contains the number '3' and also has up/down arrow icons. This is followed by an equals sign '='. The final field contains the result '10'.

Tutaj zmodyfikowano wygląd elementu *idWynik*

```
// wynik to element o id wynoszącym "idWynik"  
let wynik=document.getElementById("idWynik");  
wynik.style.padding='2px';
```

# Zadania

Zadanie 1: Napisać prosty kalkulator, wybierając znak działania z listy rozwijalnej

Zadanie 2: Napisz skrypt do działania kantoru, podający cenę zakupu przez klienta waluty w kantorze. Z formularza na stronie skrypt pobiera

- liczbę (ile waluty chcemy zakupić) z pola numerycznego
- rodzaj waluty z listy rozwijalnej (euro, dolar, funt, frank)

Uwaga! Pamiętaj, że kupujący musi patrzeć na cenę sprzedaży.

Waluta	Nazwa	Kod waluty	Kupno	Sprzedaż
 CHF PLN	Frank szwajcarski	1 CHF	↓ 4,7212	↓ 4,7426
 EUR PLN	Euro	1 EUR	↓ 4,5146	↓ 4,5371
 GBP PLN	Funt brytyjski	1 GBP	↓ 5,2323	↓ 5,2572
 USD PLN	Dolar amerykański	1 USD	↓ 4,2450	↓ 4,2667

Małgorzata Tokarek