

# JAVASCRIPT

## 1. Podstawy

- każda instrukcja (nie blok instrukcji) kończy się średnikiem
- wielkość liter ma znaczenie
- nazwy zmiennych, funkcji mogą się składać z liter i cyfr
- nazwy zmiennych, funkcji nie mogą się zaczynać od cyfry
- nazwy zmiennych, funkcji nie mogą zawierać spacji
- nazwy zmiennych, funkcji mogą zawierać znak podkreślenia
- nazwą zmiennej, funkcji nie może być słowo kluczowe

## 2. Dodawanie skryptu JavaScript w kodzie HTML

```
<script type="text/javascript">  
// tu wstaw kod javascript  
</script>
```

## 3. Dołączanie zewnętrznego pliku ze skryptem JavaScript w kodzie HTML

```
<script type="text/javascript" src="plik_z_kodem.js"></script>
```

## 4. Komentarz jednolinijkowy

```
// ta linia zostanie zignorowana
```

## 5. Komentarz wielolinijkowy

```
/* te  
   linie  
   zostaną  
   zignorowane */
```

## 6. Blok instrukcji

```
{  
    instrukcja1;  
    instrukcja2;  
    instrukcja3;  
}
```

## 7. Wyświetlanie informacji

```
document.write("tekst");// umieszcza tekst na stronie  
console.log("tekst"); // wypisuje tekst w konsoli przeglądarki
```

## 8. Komunikacja z użytkownikiem

```
alert("komunikat"); // wyświetla okno dialogowe z komunikatem
confirm("komunikat");// wyświetla okno z przyciskami OK i ANULUJ
                        // zwraca true lub false

prompt("komunikat", "odp_domyslna")
// okno z polem do wpisywania tekstu
// zwraca tekst wpisany przez użytkownika
```

## 9. Wartości

```
5           // liczba całkowita (typ number)
3.14        // liczby zmiennoprzecinkowa (typ number)
true, false // wartość logiczna (typ boolean)
"tekst"     // łańcuch znaków (typ string)
null        // wartość "pusta"
undefined   // wartość nieokreślona
```

## 10. Zmienne

```
var a; // deklaracja zmiennej o nazwie a
        // zmienna a przechowuje wartość undefined

var b = "tekst"; // zmienna b przechowuje łańcuch znaków
                 // tzw. string o wartości "tekst"

var c = 5; // zmienna c przechowuje liczbę 5

var d = true; // zmienna d przechowuje wartość logiczną "prawda"
var e = false; // zmienna e przechowuje wartość logiczną "fałsz"

var f = 5 + 10; // zmienną f przechowuje liczbę 15 (sumę 5 i 10)
var g = 5 + "5"; // zmienna g przechowuje łańcuch znaków: "55"
```

## 11. Operatory arytmetyczne

```
a = 8 + 4; // dodawanie, wynik 12
a = 8 - 4; // odejmowanie, wynik 4
a = 8 * 4; // mnożenie, wynik 32
a = 8 / 4; // dzielenie, wynik 2
a = 5 % 2; // reszta z dzielenia, wynik 1
a = (b + c) * d; // grupowanie (nawias)
                 // najpierw dodaj, potem pomnóż
a++; // inkrementacja, zwiększ a o 1 (to samo: a = a + 1)
b--; // dekrementacja, zmniejsz a o 1 (to samo: a = a - 1)
```

## 12. Operatory przypisania

```
a = 5; // przypisz do a wartość 5
a += b; // a = a + b; przypisz do a sumę a + b
```

```
a -= b; // a = a - b; przypisz do a różnicę a - b
a *= b; // a = a * b; przypisz do a iloczyn a * b
a /= b; // a = a / b; przypisz do a iloraz a / b
a %= b; // a = a % b; przypisz do a resztę z dzielenia a / b
```

### 13. Operator łączenia łańcuchów znaków (konkatenacja)

```
a = "To jest ";
b = "tekst";
c = a + b; // zmienna c zawiera tekst: "To jest tekst"
```

### 14. Operatory porówniania

```
4 == 4; // == czy równe, zwróci prawdę (true)
4 == 5; // == czy równe, zwróci fałsz (false)
5 != 4; // != czy różne, zwróci prawdę (true)
5 > 4; // > czy większe, zwróci prawdę (true)
5 < 4; // < czy mniejsze, zwróci fałsz (false)
5 >= 4; // >= czy większe lub równe, zwróci prawdę (true)
5 <= 5; // <= czy mniejsze lub równe, zwróci prawdę (true)
```

### 15. Operatory logiczne

```
a = true;
b = false;

a && b; // logiczne I, wynik fałsz (false)
b || c; // logiczne LUB, wynik prawda (true)
! a; // logiczne NIE, wynik fałsz (false)
```

### 16. Instrukcja if

```
if (wiek >= 18) { // jeżeli wiek większy lub równy 18
    odp = "pełnoletni"; // wykonaj, jeżeli warunek został spełniony
}
```

### 17. Instrukcja if else

```
if (wiek >= 18) { // jeżeli wiek większy lub równy 18
    odp = "pełnoletni"; // wykonaj, jeżeli warunek został spełniony
}
else {
    odp = "niepełnoletni"; // wykonaj, jeżeli warunek niespełniony
}
```

### 18. Instrukcja switch

```
switch(liczba) { // sprawdź zmienną liczba
    case 1: // jeżeli liczba == 1 to wykonaj:
```

```

        wynik = "Liczba 1";
        break; // wyjdź z instrukcji switch
    case 2: // jeżeli liczba == 2 to wykonaj:
        wynik = "Liczba 2";
        break; // wyjdź z instrukcji switch
    case 3: // jeżeli liczba == 3 to wykonaj:
        wynik = "Liczba 3";
        break; // wyjdź z instrukcji switch
    default: // jeżeli liczba różna od 1,2,3:
        wynik = "Inna liczba";
}

```

### 19. Pętla while

```

var i = 1; // przypisz do i wartość 1

while (i < 100) { // jeżeli warunek spełniony wejdź do pętli
    document.write(i); // wypisz zawartość i
    i++; // zwiększ i o 1
}

```

### 20. Pętla do while

```

var i = 1; // przypisz do i wartość 1

do {
    document.write(i); // wypisz wartość i
    i++; // zwiększ i o 1
} while(i < 100) // jeżeli spełniony warunek powtórz pętlę

```

### 21. Pętla for

```

// szablon pętli for:
// for(licznik; warunek pętli; modyfikacja licznika) {
//     instrukcje_pętli;
// }

// działa tak, jak w punkcie 19. pętla while
for(var i = 1; i < 100; i++) {
    document.write(i);
}

```

### 22. Słowo kluczowe break

```

for(var i = 1; i < 100; i++) {
    document.write(i);
    if (i == 5) { // jeżeli i równe 5
        break; // przerwij pętlę
    }
}

```

### 23. Słowo kluczowe **continue**

```
for(var i = 1; i < 100; i++) {  
    if (i == 5) { // jeżeli i równe 5  
        continue; // opuść instrukcje poniżej  
    } // skocz na początek pętli  
    document.write(i);  
  
}
```

### 24. Funkcje, słowo kluczowe **function**

```
function hello() { // funkcja bezargumentowa  
    alert("Cześć"); // ciało funkcji  
}  
hello(); // wywołanie funkcji  
  
function dodaj(a, b) { // funkcja, która przyjmuje dwa argumenty  
    return a + b; // funkcja zwraca sumę a i b  
}  
var z = dodaj(5, 10); // zmienna z zawiera wartość 15
```

### 25. Obiekty (sposób 1.)

```
// definicja obiektu osoba  
var osoba = {  
    imie: "Piotr", // właściwość imie  
    nazwisko: "Kowalski", // właściwość nazwisko  
    wiek: 45, // właściwość wiek  
    pelnaNazwa: function () { // metoda pelnaNazwa  
        return this.imie + " " + this.nazwisko;  
    }  
};  
  
osoba.wiek = 46; // zmiana wartości właściwości wiek  
alert(osoba.pelnaNazwa()); // wywołanie metody pelnaNazwa
```

### 26. Obiekty (sposób 2. - konstruktor)

```
// definicja konstruktora obiektów typu Osoba  
function Osoba(imie, naz, wiek) {  
    this.imie = imie;  
    this.nazwisko = naz;  
    this.wiek = wiek;  
    this.pelnaNazwa = function() {  
        return this.imie + " " + this.nazwisko;  
    }  
}
```

```
// utwórz obiekt osoba1 typu Osoba
var osoba1 = new Osoba("Piotr", "Kowalski", 45);
osoba1.wiek = 46; // zmiana wartości właściwości wiek
alert(osoba1.pelnaNazwa()); // wywołanie metody pelnaNazwa
```