Wprowadzenie do JavaScript

Język JavaScript jest obiektowym skryptowym językiem programowania nawiązującym do języka C. Ponadto jest językiem interpretowanym, czyli bez konieczności kompilowania kodu możemy oglądać efekty jego działania. JavaScript należy do grupy technologii frond-endowych, to znaczy kod jest wykonywany po stronie użytkownika w środowisku przeglądarki internetowej.

Implementacja JS w HTML

Kod źródłowy JS może być umieszczony wewnątrz dokumentu HTML między znacznikami <script></script>

Przykładowe umiejscowienie kodu JS w HTML:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script>
   //komentarz w Javascript
  document.write("Hello world! - wstawienie Js w <head>");
  </script>
</head>
<body>
  <script>
   //komentarz w Javascript
  document.write("Hello world! - wstawienie Js w dowolnym miejscu w <body>");
  </script>
```

```
</body>

  <script>
   //komentarz w Javascript
  document.write("Hello world! - wstawienie Js  po zakończeniu <body>");
  </script>

</html>
```

Znaczniki skryptu mogą być wstawiane w dowolnym miejscu dokumentu, najczęściej umieszczane są jak w przykładzie powyżej i tak naprawdę od rodzaju skryptu zależeć będzie, gdzie będzie najlepiej działać.

Skrypt umieszczony w zewnętrznym pliku script.js

Dobrą praktyką jest wielokrotne wykorzystanie raz napisanego kodu. Możemy kod umieścić w pliku o rozszerzeniu *.js, a następnie podłączyć go do wielu plików HTML-a:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src=”script.js”> </script>
</head>
```

Wtedy w pliku script.js umieszczamy już tylko kod JavaScript, np.:

```
document.write("Hello world");
```

Deklaracja zmiennych

Zmienną możemy zadeklarować za pomocą słowa kluczowego **var i wpisując nazwę zmiennej** - (nazwa może zawierać litery, cyfry i znak podkreślenia, **nie może zawierać polskich znaków**) przykład:

```
<script>
 //utworzenie zmiennej 'imie' typu string
var imie = "Ania";

//utworzenie zmiennej 'x' typu liczbowego
var x = 10;

// utworzenie zmiennej 'info' typu logicznego
var info = true;

// utworzenie zmiennej 'pojemnik' typu null
var pojemnik = null;

// jeśli zmiennej 'y' nie przypiszemy wartości to otrzymuje ona typ undefned
var y;

</script>
```

W języku javascript nie musimy sami definiować typu tworzonej zmiennej jak np. w jezyku C++, typ zmiennej jest przypisywany do niej po nadaniu jej wartości. Typ ten może ulec zmianie po ponownym przypisaniu wartości.

Operatory

1. Operatory arytmetyczne:

Służą do wykonywania operacji arytmetycznych.

Operator	Działanie	Przykład
+	Dodawanie	a + b
-	Odejmowanie	a - b
*	Mnożenie	a * b
/	Dzielenie	a / b
%	Modulo (reszta z dzielenia)	a % b
++	Inkrementacja	x++, ++x
--	Dekrementacja	x--, --x

Przykład:

```
x = 5;
y = x + 2;      //y = 7
y = x - 1;      //y = 4
y = x * 3;      //y = 15
y = x / 2;      //y = 2.5
y = x % 2;      //1 bo % oznacza resztę z dzielenia
x--;            //to to samo co x = x - 1
x++;            //to to samo co x = x + 1
```

2. Operatory porównania

Porównują argumenty. Ich wynikiem jest wartość logiczna true/false

Operator	Działanie	Przykład
==	Wynik ***true***, gdy argumenty są równe	a == b
!=	Wynik ***true***, gdy argumenty są różne	a != b
===	Wynik ***true***, gdy argumenty są tego samego typu i są równe	a === b
!==	Wynik ***true***, gdy argumenty są tego różnych typów lub są różne	a !== b
>	Wynik ***true***, gdy argument pierwszy jest większy od drugiego	a > b
<	Wynik ***true***, gdy argument pierwszy jest mniejszy od drugiego	a < b
>=	Wynik ***true***, gdy argument pierwszy jest większy od drugiego lub jest mu równy	a > b
<=	Wynik ***true***, gdy argument pierwszy jest mniejszy od drugiego lub jest mu równy	a < b

Przykład:

Dla zmiennej: var x = 8;

```
if (x === 10) {
   //ten kawałek kodu się nie wykona
}

if (x <= 10) {
   //ten kawałek kodu się wykona
}

if (x !== 8) {
   //ten kawałek kodu się nie wykona
}
```

3. Operatory logiczne

Służą do wykonania operacji na argumentach, które posiadają wartość logiczną true/false

Operator	Działanie	Przykład
&&	Iloczyn logiczny (AND)	a && b
||	Suma logiczna (OR)	a || b
!	Negacja logiczna (NOT)	!a

Przykład:

```
var x  = 8;
var y = 15;

if (x === 8 && y === 10) {
   //ten kawałek kodu się nie wykona bo mamy "i", oba muszą być spełnione, a nie są
}

if (x === 8 || y === 8) {
   //ten kawałek się wykona bo mamy "lub" - jeden z warunków jest poprawny
}

if (!(x === 8)) {
   //ten kawałek się nie wykona, bo mamy negację!
   //powyższy warunek jest jednoznaczny z x!== 8
}
```

4. Operatory bitowe

Umożliwiają wykonanie operacji na poszczególnych bitach liczb

Operator	Działanie	Przykład
&	Iloczyn bitowy (AND)	a & b
|	Suma bitowa (OR)	a | b
~	Negacja bitowa (NOT)	~ a
^	Bitowa różnica symetryczna	a ^ b
>>	Przesunięcie bitowe w prawo	a >> n
<<	Przesunięcie bitowe w lewo	a << n
>>>	Przesunięcie bitowe w prawo z wypełnieniem zerami	a >> > n

5. Operatory przypisania

Przypisują wartości argumentom znajdującym się po lewej stronie operatora

Operator	Przykład	Znaczenie
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Przykład:

```
var x = 10;

x*=5;


var text = "Przykładowy tekst";

text += "króry nie zmieścił się";

text += "w jednej linii";
```