Składnia języka JavaScript

Składnia języka jest ściśle określonym zbiorem reguł, których należy przestrzegać. W języku JavaScript stosuje się następujące kategorie pojęć składniowych: wyrażenia, zmienne i literaly. Do elementów języka należy również zbiór predefiniowanych obiektów i funkcji (tablice, data, funkcje matematyczne).

**Instrukcje**

Instrukcje w języku JavaScript mogą być pisane pojedynczo, w jednym wierszu lub w kilku wierszach. Średnik odgrywa rolę ogranicznika instrukcji, ale nie jest wymagany, jeżeli po instrukcji wystąpi znak nowego wiersza. Jeżeli kilka instrukcji zostanie zapisanych w jednym wierszu, muszą być one oddzielone średnikiem.

**Wielkość liter**

W programie JavaScript wielkość liter przeważnie ma znaczenie. Stosowane są następujące zasady:

* Słowa kluczowe pisane są małymi literami, np. for.
* Nazwy obiektów wbudowanych pisane są od dużej litery, pozostałe litery są małe, np. Date.
* Nazwy obiektów DOM pisane są małymi literami, ale w nazwach metod tych obiektów dopuszczane są małe i duże litery, np. toLowerCase.
* Definiowanym zmiennym, obiektom, funkcjom można nadawać nazwy pisane w dowolny sposób. Oprócz liter w nazwach można stosować cyfry i znak podkreślenia. Nazwa może zaczynać się literą lub znakiem podkreślenia. W nazwie nie można używać słów zarezerwowanych.
* Białe znaki, typu spacja, znak tabulacji itp., są nieistotne.

**Komentarze**

Komentarze są ignorowane w trakcie przetwarzania kodu. Mogą być umieszczane w dowolnym miejscu kodu. Komentarzem jest każdy wiersz rozpoczynający się od znaków**//**. Komentarz rozpoczynający się od // można także umieszczać po instrukcji np:

a = b + c // suma wartości

Jeżeli komentarz zawiera wiele wierszy, można użyć ograniczników**/\*** (do rozpoczęcia komentarza) i**\*/** (do zakończenia komentarza).Tak zapisane komentarze dopuszczalne są tylko wewnątrz znaczników script i /script

**Zmienne**

Zmienne służą do przechowywania danych i wyników w celu dalszego ich wykorzystania. W języku JayaScript zmienne deklaruje się za pomocą słowa kluczowego **var** poprzedzającego nazwę zmiennej. Nazwa zmiennej:

* może zawierać litery, cyfry i znak podkreślenia
* nie może zawierać znaków narodowych
* wielkość liter używanych w nazwach ma znaczenie
* nazwa zmiennej nie może rozpoczynać się od cyfry.

Tworzone zmienne nie mają określonego typu. Typ jest przypisywany do zmiennej po nadaniu jej wartości. Typ danych nie jest przypisany do zmiennej na stałe i może ulegać zmianie.

**Skalarne typy danych**

W języku JavaScript dopuszczane są następujące typy danych:

* **typ liczbowy,**
* **typ łańcuchowy,**
* **typ logiczny,**
* **typ null.**

**Typ liczbowy**

Służy do zapisywania liczb. Można zapisywać liczby w formatach wykładniczym, dziesiętnym, ósemkowym i szesnastkowym. Jeżeli liczba:

* zostanie poprzedzona cyfrą zero (prefiks **0**), to jest traktowana jako wartość **ósemkowa** (np. 042).
* zostanie poprzedzona ciągiem znaków **0x** lub **OX**, to jest traktowana jako wartość **szesnastkowa** (heksadecymalna) (np. 0x23A).
* nie jest poprzedzona żadnym znakiem lub jest poprzedzona znakiem +, jest to wartość **dodatnia**.
* jest poprzedzona znakiem -, jest to wartość **ujemna**.

Wartości liczbowe mogą być zapisywane w notacji **wykładniczej** (np. 3E-2)

Przykład deklaracji zmiennych typu liczbowego:

X=123;

Y=12.3545;

Z=0xAB56;

T=1.4E-5

**Typ łańcuchowy string**

Zawiera ciągi znaków o dowolnej długości. Ciąg znaków musi być umieszczony w ogranicznikach typu **cudzysłów** lub **apostrof**.

Nazwa=’Warszawa’;

Imie=”Alicja”;

**Typ logiczny boolean**

Przyjmuje jedną z dwóch wartości **prawda** (**true**) lub **fałsz** (**false**). Wartości typu logicznego są wykorzystywane przy budowaniu wyrażeń logicznych, porównywaniu danych, określaniu, czy wykonywana operacja zakończyła się sukcesem.

Zm=true;

S=false;

**Typ null**

Jest to typ specjalny określający **wartość pustą** (**null**). Nie przechowuje żadnej wartości.

**Typ undefined**

Jest to typ zawierający zmienne, którym nie została nadana żadna wartość.

**Złożone typy danych**

W języku JavaScript zostały zdefiniowane również złożone typy danych. Oznacza to, że wewnątrz nich można przechowywać więcej niż jedną wartość. Złożonymi typami danych są:

* **typ object**
* **typ array**

**Typ object**

Służy do reprezentacji obiektów. Wykorzystywane są obiekty wbudowane oraz udostępniane przez przeglądarkę.

**Typ array**

Typ array jest typem tablicowym, który służy do przechowywania wielu wartości.

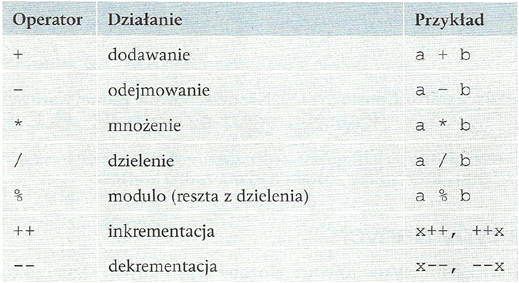
**Operatory**

Operatory w języku JavaScript zostały podzielone na następujące grupy:

* **arytmetyczne**
* **porównania**
* **bitowe**
* **logiczne**
* **przypisania**
* **pozostałe**

**Operatory arytmetyczne**

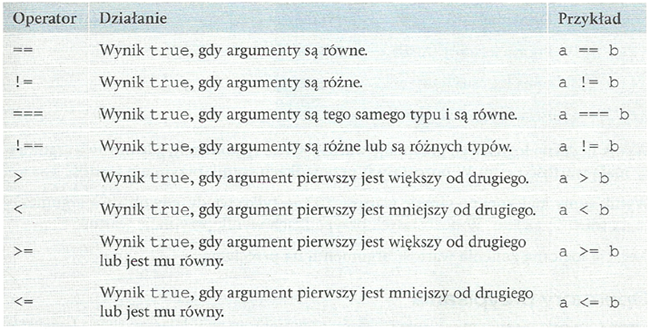
Służą do wykonywania operacji arytmetycznych. W tej grupie znajdują się też operatory inkrementacj i (zwiększania) i dekrementacj i (zmniejszania). Operatory arytmetyczne są dwuargumentowe, natomiast operatory inkrementacji i dekrementacji są jednoargumentowe



**Operator inkrementacji** powoduje zwiększenie wartości o jeden. Może występować w postaci **przedrostkowej** (++x) lub **przyrostkowej** (x++). Operacja x++ zwiększa wartość zmiennej po jej wykorzystaniu, natomiast ++x przed jej wykorzystaniem. Operator dekrementacji działa analogicznie, tylko zamiast zwiększać wartości zmiennych, zmniejsza je.

**Operatory porównania**

Operatory porównania porównują argumenty. Ich wynikiem jest wartość logiczna true (prawda) lub false (fałsz)

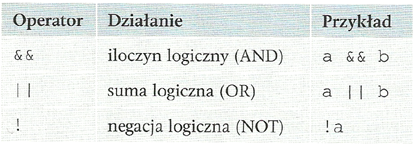


**Operatory bitowe**

Operatory bitowe umożliwiają wykonanie operacji na poszczególnych bitach liczb

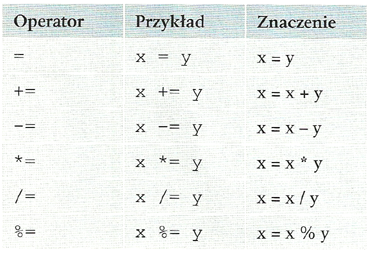
**Operatory logiczne**

Operatory logiczne wykonują operacje na argumentach, które posiadają wartość logiczną (true lub false)



**Operatory przypisania**

Operatory przypisania przypisują wartości argumentom znajdującym się po lewej stronie operatora. Oprócz prostej operacji przypisania pozwalają na połączenie operacji przypisania z inną operacją, np. dodawania.  
Zapis i += 7 oznacza w praktyce to samo co zapis i = i + 7. Stosowanie takich skróconych zapisów upraszcza tworzenie bardziej rozbudowanych skryptów.



Instrukcja warunkowa

Instrukcja warunkowa służy do sprawdzenia w programie warunku i w zależności od tego, czy wynik będzie prawdą, czy fałszem, dalszego wykonywania instrukcji. Do takiego sprawdzania służy instrukcja

**if ... else**

 uwaga: przed else nie ma średnika

 if (warunek)

{instrukcje, gdy warunek prawdziwy}

else

{instrukcje, gdy warunek fałszywy};

Przykładowy skrypt z instrukcją warunkową:

<script>

x=prompt('podaj pierwszą liczbę');

y=prompt('podaj drugą liczbę');

if (x>y)

{document.write('pierwsza liczba jest większa');}

else

{document.write('pierwsza liczba nie jest większa');}; </script>

**Zadania**

1. Napisz skrypt sprawdzający która z dwóch liczb podanych z klawiatury jest większa.
2. Napisz skrypt sprawdzający czy użytkownik komputera jest pełnoletni.
3. Napisz skrypt sprawdzający czy podana liczba jest ujemna, dodatnia lub równa zero.
4. Napisz skrypt sprawdzający czy podany rok jest przestępny.
5. Napisz skrypt zamieniający nr miesiąca na jego nazwę.
6. Napisz skrypt obliczający i sprawdzający współczynnik BMI.