

ZAJĘCIA SPECJALISTYCZNE

JavaScript



TEMAT 1-2: Zmienne, typy i operatory.

Autor dokumentu: Wojciech Galiński

poniedziałek, 9 lutego 2015 r.

312[01]/T,SP/MENIS/2004.06.14

ŹRÓDŁA WIEDZY: <http://pl.wikipedia.org/>, <http://webmaster.helion.pl/index.php/kurs-javascript>,
<http://krook.org/jsdom/>, <http://www.dynamicdrive.com/>, <http://www.w3schools.com>.

Zagadnienia obowiązkowe

1. Standardowe typy danych – wyróżniamy:

- TYP LICZBOWY – służy do reprezentowania liczb bez rozróżniania na typy całkowite i zmiennoprzecinkowe np. **123**, **-456** (dziesiętne); **067**, **-051** (ósemkowe); **0xAB**, **-0xC** (szesnastkowe); **0.1**, **-0.2**; **3.4E5**, **1E-2** (zmiennoprzecinkowe);

→ PRZYKŁAD:

```
document.write( 31 + ' ' + (-255) + ' ' + 037 + ' ' + (-0377) + ' ' +  
                + 0x1F + ' ' + (-0xFF) + ' ' +  
                + 3.14 + ' ' + (-0.2) + ' ' + 7E9 + ' ' + 7.07E-1 );
```

- TYP ŁAŃCUCHOWY – służy do reprezentowania ciągów znaków. Ciągi takie powinny być ujęte w cudzysłowy lub w apostrofy (warto trzymać się tu konwencji języka C/C++).

Jeżeli tekst chcemy rozciągnąć na kilka linii, apostrofy muszą się znajdować w każdej linii. Ciągi te należy połączyć operatorem „+”.

Ciąg taki może zawierać też znaki specjalne, np.: `'\n'`, `'\t'`, `'\"'`, `'\''`, `'\\'`;

PRZYKŁAD:

```
document.write( '<pre>\t<b>Program \"Informacja\"</b>\n' +  
                'Plik programu: \'C:\\plik.exe\\'\n' +  
                'Wersja: 1.0 z dnia 21 września 2012 r.\n' +  
                'Autor: Wojciech Galiński</pre>' );
```

- TYP LOGICZNY – pozwala na określenie dwóch wartości logicznych: prawda (true) i fałsz (false);

PRZYKŁAD: `document.write(true + ' ' + false);`

- TYP OBIEKTOWY – służy do reprezentacji obiektów (obiekty będą omawiane w innym temacie).

2. Specjalne typy danych – wyróżniamy:

- TYP NULL – określa wartość pustą;

- TYP UNDEFINED – określa wartość niezdefiniowaną (obiekty mogą mieć taką wartość z jednego z dwóch powodów: obiekt nie istnieje albo wartość została celowo przypisana).

PRZYKŁAD: `document.write(null + ' ' + undefined);`

3. Zmienne – służą do przechowywania różnego typu pojedynczych wartości. Uzyskujemy do nich dostęp poprzez ich nazwy, które muszą spełniać następujące warunki:

- muszą zaczynać się od litery lub znaku podkreślenia;
- mogą zawierać: **lity** (wielkość liter jest rozróżniana), **cyfry**, znaki dolara „\$” i znaki podkreślenia „_”;
- mogą zawierać znaki narodowe (np. polskie litery: ą, ć, ę, ł, ń, ó, ś, ź, ż).

4. Definiowanie zmiennej – zmienną w języku JavaScript definiujemy zwykle poprzez przypisanie do niej wartości (ale nie jest to konieczne – 3 przypadek w poniższej tabeli).

Poniższa tabela pokazuje możliwe sposoby definiowania zmiennych:

Rodzaj zmiennej	Przypisanie wartości do zmiennej	Typ zmiennej
GLOBALNA	<code>nazwa_zmiennej = wartość_zmiennej;</code>	Taki sam, jak typ przypisanej wartości
LOKALNA	<code>var nazwa_zmiennej = wartość_zmiennej;</code>	Taki sam, jak typ przypisanej wartości
	<code>var nazwa_zmiennej;</code>	undefined

5. **Typ zmiennej** – zależy od wartości przypisanej do tej zmiennej i nie jest stały. Każde przypisanie wartości może spowodować zmianę typu zmiennej, o ile przypisywana wartość jest innego typu, niż poprzednia wartość przechowywana przez zmienną, np.

```
var zmienna; // typ "undefined"
zmienna = 5; // typ liczbowy
zmienna = 'Nysa' lub zmienna = "Pi"; // typ łańcuchowy
zmienna = true; // typ logiczny
```

6. **Operatory arytmetyczne** – dzielą się na:

➔ JEDNOARGUMENTOWE – wyróżniamy:

- ✓ dodatni znak liczby „+” (np. +5);
- ✓ ujemny znak liczby „-” (np. -3);
- ✓ inkrementacja „++” – preinkrementacja (np. ++k), postinkrementacja (np. k++);
- ✓ dekrementacja „--” – predekrementacja (np. --k), postdekrementacja (np. k--);

PRZYKŁAD: `document.write((5+3)+' '+ (5-3)+' '+ (5*3)+' '+ (5/3));`

➔ DUUARGUMENTOWE – wyróżniamy:

- ✓ dodawanie lub łączenie wartości zmiennych „+” (np. 200+45, "PI: " + 3.14 + ' '),
- ✓ odejmowanie „-” (np. 200 - 45),
- ✓ mnożenie „*” (np. 5*7),
- ✓ dzielenie „/” (np. 3/4),
- ✓ reszta z dzielenia „%” (np. 7%3).

7. **Debugowanie skryptów JavaScript** – w JavaScript brak jest kompilatora, więc jak sprawdzić poprawność skryptu? Z pomocą przychodzą przeglądarki internetowe, które posiadają okno typu „Konsola błędów” lub podobne dostępne za pomocą skrótu klawiszowego: Firefox (skrót klawiszowy: **[Ctrl][Shift][J]**), Opera – (skrót klawiszowy: **[Ctrl][Shift][O]**), Chrome (skrót klawiszowy: **[Ctrl][Shift][J]**), Safari – (skrót klawiszowy: **[Lewy Alt][Mac][I]** – działa po zainstalowaniu pakietu „Developer Tools for Safari”), MS IE – (skrót klawiszowy: **[F12]**). Aby podglądać wartości zmiennych (będzie o nich mowa później), wystarczy wyświetlić ich wartości w przeglądarce, np. za pomocą funkcji: `„document.write(zmienna);”`.

Zadania

- Najpierw utwórz tylko zmienne (wybierz dla nich intuicyjne nazwy). Następnie przypisz do nich następujące wartości: **liczbę całkowitą**, $\frac{1}{4}$, $7 \cdot 10^7$, **znak „!”**, **odpowiedź na pytanie: „czy świeci słońce?”**, **tekst „C:\TEMP”**, **„null”**. Wyświetl na ekranie ich nazwy i wartości.
- Sprawdź, czy da się wyświetlić na ekranie wartość zmiennej, do której nic nie przypisano.
- Z pomocą funkcji **„prompt()”** wczytaj 2 liczby i wykonaj do nich operacje wykorzystując poznane operatory jedno- i dwuargumentowe. Korzystając z funkcji **„document.write”**. Wyświetl te operacje i wyniki na ekranie, np. $3 + 2 = 5$, $3 - 2 = 1$, itd.
- Zdefiniować odpowiednio zmienną **„i”** i zmodyfikować instrukcję **„document.write(i);”** w taki sposób, żeby po 1000-krotnym skopiowaniu wyświetlały się kolejne liczby od 1 do 1000.
- Wykonać poprzednie zadanie w odwrotnej kolejności (od 1000 do 1).