

Typy danych w JavaScript

Do dyspozycji mamy liczby - podajemy liczby bez cudzysłówów, łańcuchy - tekst, liczby umieszczone w cudzysłowach oraz wartości logiczne - true i false.

W języku JavaScript można wykonywać operacje nie tylko na liczbach, ale także na innych typach wartości. Pełna lista tych typów jest następująca (w nawiasach są nazwy angielskie):

- liczbowy (Number);
- łańcuchowy, ciągi znaków (String);
- logiczny (Boolean);
- obiektowy (Object);
- tablicowy (Array);
- typ specjalny: Null i Undefined.

Typ liczbowy

Typ liczbowy służy do reprezentacji liczb, przy czym nie ma występującego w klasycznych językach programowania rozróżnienia na typy całkowitoliczbowe i rzeczywiste (zmiennopozycyjne). Liczby zapisywane są za pomocą literałów (inaczej stałych napisowych, z ang. string constant, literal constant) liczbowych. Obowiązują przy tym następujące zasady:

- Jeżeli ciąg cyfr nie jest poprzedzony żadnym znakiem lub jest poprzedzony znakiem +, reprezentuje on wartość dodatnią; jeżeli natomiast jest poprzedzony znakiem –, reprezentuje wartość ujemną.
- Jeżeli literał rozpoczyna się od cyfry zero, jest traktowany jako wartość ósemkowa.
- Jeżeli literał rozpoczyna się od ciągu znaków 0x, jest traktowany jako wartość szesnastkowa (heksadecymalna). W zapisie wartości szesnastkowych mogą być wykorzystywane zarówno małe, jak i duże litery alfabetu od A do F.
- Literały mogą być zapisywane w notacji wykładniczej, w postaci X.YeZ, gdzie X to część całkowita, Y część dziesiętna, natomiast Z to wykładnik potęgi liczby 10. Zapis taki oznacza to samo, co $X.Y * 10^Z$.

Przykłady literałów:

123	dodatnia całkowita wartość dziesiętna 123
-123	ujemna całkowita wartość dziesiętna –123
012	dodatnia całkowita wartość ósemkowa równa 10 dziesiętnie
-024	ujemna całkowita wartość ósemkowa równa 20 dziesiętnie
0xFF	dodatnia całkowita wartość szesnastkowa równa 255 dziesiętnie
-0x0f	ujemna całkowita wartość szesnastkowa równa –15 dziesiętnie
1.1	dodatnia wartość rzeczywista 1.1
-1.1	ujemna wartość rzeczywista 1.1
0.1E2	dodatnia wartość rzeczywista równa 10
1.0E-2	dodatnia wartość rzeczywista równa 0.01

Typ łańcuchowy

Typ łańcuchowy pozwala na reprezentację ciągów znaków. Ciągi te powinny być ujęte w znaki cudzysłowu, aczkolwiek dopuszczalne jest również użycie znaków apostrofu. Przykładowy literał napisowy:

"abcdefg"

Typ logiczny

Typ logiczny (boolean) pozwala na określenie dwóch wartości logicznych: prawda i fałsz. Wartość prawda jest w języku JavaScript reprezentowana przez słowo **true**, natomiast wartość fałsz — przez słowo **false**.

Typ obiektowy

Typ obiektowy służy do reprezentacji obiektów. Nie ma specjalnego słowa kluczowego oznaczającego ten typ. Najczęściej wykorzystuje się obiekty wbudowane oraz udostępniane przez przeglądarkę.

Typ tablicowy

Następny dostępny typ to tablice. Służą one do grupowania danych w strukturę, gdzie każdemu elementowi przypisany jest określony indeks. Opiszę je dokładniej w jednej z kolejnych lekcji. Tu wspomnę tylko jeszcze że tablice, podobnie jak ciągi znaków, też są obiektem (Array).

Typy specjalne

Wyróżniamy dwa rodzaje typów specjalnych: **null** i **undefined**.

Typ **null** określa wartość pustą („brak wartości” lub „brak obiektu”). Przypisanie tej wartości do zmiennej powoduje wyczyszczenie jej wartości, ale nie powoduje usunięcia zmiennej.

Typ **Undefined** również posiada tylko jedną wartość: **undefined**. Typ undefined określa wartość niezdefiniowaną. Jest ona przypisana do zmiennych, które zostały zadeklarowane, ale nie zostały zainicjalizowane (nie posiadają przypisanej wartości), lub gdy podane pole (właściwość) obiektu nie istnieje. Typ Undefined również posiada tylko jedną wartość: **undefined**.

Wartość ta ma także specjalną właściwość - próba porównania jej z samą sobą zawsze daje wynik **false** (fałsz). Dlatego też, aby sprawdzić czy coś posiada wartość undefined, należy użyć operatora **typeof**.

KONWERSJA TYPÓW ZMIENNYCH

parseFloat(zmienna) - konwersja zmiennej do typu float

parseInt(zmienna) - konwersja zmiennej do typu int (liczbowego)

toString(zmienna) - konwersja zmiennej do typu łańcuchowego

isFinite(zmienna) - zwraca true jeżeli zmienna ma wartość nieskończoność

isNaN(zmienna) - zwraca true jeżeli zmienna nie jest liczbą ("Not a Number")

Number(zmienna) - sprawdza czy zmienna jest typu liczbowego (zwraca NaN jeżeli nie, zwraca wartość jeżeli tak)