

Typy wartości - proste

Liczby	Dowolna wartość numeryczna
String	
Boolean	Wartość logiczna, dwie wartości TRUE i FALSE
Function	wartość zwracana przez funkcję
Null	posiada tylko jedną wartość: NULL . Oznacza ona <i>brak wartości</i> lub <i>brak obiektu</i> . Przypisanie tej wartości do zmiennej powoduje wyczyszczenie jej wartości, ale nie powoduje usunięcia zmiennej.
Undefined	posiada tylko jedną wartość: undefined

Operator `typeof()` zwraca string z nazwą typu jaki ma przekazany parametr (np. zmienna). Może on zwrócić jedną z podanych wartości: *number*, *string*, *boolean*, *object*, *function* lub *undefined*.

Typy obiektowe

Object	obiekt, typ złożony, może przechowywać inne typy danych oraz funkcje
Function	wartość zwracana przez funkcję
Array	tablica, grupowanie danych w strukturę, gdzie każdemu elementowi przypisany jest określony indeks
Symbol	symbol
[]	obiekt
{}	obiekt

Operatory arytmetyczne

$x + y$	Sumuje argumenty x i y
$x - y$	Różnica y od x
$x * y$	Mnożenie $x * y$
x / y	Dzielenie x przez y
$x \% y$	modulo - reszta z dzielenia x przez y
$x++$, $+$	inkrementacja wartości o 1 (<i>przed</i> , <i>po</i>)
$x--$, $+$	dekrementacja wartości o 1 (<i>przed</i> , <i>po</i>)
$-x$	zmiana znaku wartości x

W przypadku *SUMY* gdy jedna wartość ma typ **NUMBER**, a druga **STRING** wynik jest typu **STRING**

Porównania

$x == y$	TRUE jeśli $x = y$, niezależnie od typu
$x === y$	TRUE jeśli $x = y$ łącznie z typami
$x != y$	TRUE jeśli x jest różne od y
$x !== y$	TRUE jeśli x i y nie są identyczne
$x > y$	TRUE jeśli x jest większe od y
$x >= y$	TRUE jeśli x jest większe lub równe y
$x \&\& y$	TRUE jeśli obie wartości są TRUE
$x y$	TRUE jeśli co najmniej jedna z wartości jest TRUE
$x \wedge y$	TRUE jeśli jedna z wartości jest TRUE
$!x$	TRUE jeśli x jest FALSE

Przypisania

$x = y$	Ustawia x wartość y
$x += y$	skrótowa postać $x = x + y$
$x -= y$	skrótowa postać $x = x - y$
$x *= y$	skrótowa postać $x = x * y$
$x /= y$	skrótowa postać $x = x / y$
$x \% = y$	skrótowa postać $x = x \% y$

Obsługa tablic

<code>push()</code>	Dokłada element na koniec tablicy
<code>pop()</code>	Usuwa ostatni element tablicy
<code>unshift()</code>	Dokłada element na początku tablicy
<code>shift()</code>	Usuwa pierwszy element tablicy
<code>delete</code>	usuwa element, np. po indeksie
<code>concat()</code>	Służy do łączenia tabel
<code>join()</code>	zwraca STRING połączonych elementów, parametr jako delimiter
<code>length()</code>	zwraca długość tablicy
<code>splice()</code>	Umożliwia zmianę lub usunięcie kilku elementów
<code>sort()</code>	Sortuje elementy typu STRING

Obsługa tablic (cont)

```
reverse()   Odwraca kolejność tablicy
```

```
var owoce = ["Li mon ka", " Man - go"];
owoce.push("K iwi "); //Dokłada "K iwi" na końcu
owoce.pop(); //Usuwa element "K iwi" z owoce
owoce.unshift("K iwi "); //wkłada "K iwi" do owoce
owoce.shift(); //Usuwa element "Li mon ka" z owoce
```

Sortowanie numeryczne można wykonać:

```
var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return a-b});
```

lub w odwrotnej kolejności - malejąco:

```
points.sort(function(a, b){return b-a});
```

Instrukcja warunkowa if .. else

```
if (warunek) {
    rób coś;
} else if (warunek 2) {
    rób coś innego;
} else {
    rób coś jeszcze innego;
}
```

Instrukcja warunkowa switch .. case

```
switch(zmienna)
case 0:
    ale rt( " zmienna ma wartość zero");
    break;
case 1:
    ale rt( " zmienna ma wartość jeden");
    break;
default:
    ale rt( " zmienna posiada inną wartość");
    break;
}
```

