Rodzaje zmiennych w JS

Typ zmiennej określa rodzaj przechowywanych w niej informacji.

W języku JavaSript wyróżnia się sześć podstawowych typów danych:

liczby,

łańcuchy,

wartości logiczne,

obiekty,

funkcje

wartości niezdefiniowane.

Wszystkie liczby w języku JavaScript składają się z 64 bitów.
Liczby ułamkowe zapisuje się z użyciem kropki. `9.81`
Do wyrażania bardzo dużych i bardzo małych liczb można też używać notacji naukowej. W tym celu należy dodać literę `e` i wykładnik potęgi: `2.998e8`
Powyższy zapis jest równoważny z : 2.998 * = 299800000.

Rodzaje zmiennych w JS

Logiczne (boolean) - zawierające wartość będącą prawdą lub fałszem

```
var zmienna = true;
var zmienna = false;
```

Znakowe - zawierające wartość będącą ciągiem znaków (stringiem)

```
var zmienna = "abcd    sdfgh";
var zmienna = "Jakiś tekst";
```

null - zawierające wartość pustą czyli *null*. Wartość **null** nie jest ani **0**, ani pustym ciągiem znaków **''**. Null to nic - po prostu nic.

JavaScript rozróżnia wielkość liter, tak więc **null** nie jest równoznaczne z Null czy NULL

undefined - zmienna niezadeklarowana. Ten typ zmiennej służy do sprawdzania, czy zmienna w ogóle istnieje.

Typy i zmienne

JavaScript jest językiem typowanym dynamicznie

Zmienne

Deklarowanie zmiennej:

```
var zmienna = 10;
var zmienna = "to jest tekst";
var ob = document.Formularz.poleTekstowe;
```

Dla uniknięcia przypadkowych błędów, deklarację zmiennych poprzedzamy słowem **var** (nie jest wymagane).
Zmienne dzielą się na **globalne** i **lokalne**. Globalne są deklarowane poza funkcjami, a lokalne wewnątrz funkcji. Lokalne zmienne są dostępne tylko dla danej funkcji, a globalne są dostępne dla całego skryptu.

Przykład:

```
var zmienna = 69;
zmienna = "nowa wartość" // tutaj nie będzie
  błędu
x = "x = "+40; // zwraca "x = 40"
y = 69-9;    // zwraca 60
z = "69"+9;    // zwraca 699
```

Zmienne i stałe

Typy danych

Javascript udostępnia metodę **typeof()**, dzięki której możesz sprawdzać typ danych

```
var n = 3;
document.write( typeof(n) ) //wypisze się "number"

var s = "napis";
document.write( typeof(s) ) //wypisze się "string"
```

- Deklaracje zmiennych
  - przez przypisanie wartości

    `x=5;`
  - przez słowo **`var`**
  - jeśli zmiennej nie zostanie przypisana wartość to przyjmuje wartość **`undefined`**
- Deklaracja stałych
  - stała nie może zmieniać wartości lub być przedeklarowana
  - **`const wroclaw = "071";`**

Rodzaje zmiennych:

Liczbowe - czyli zawierające wartość liczbową

```
var zmienna = 10.5;
var zmienna = 100;
```

Logiczne (boolean) - zawierające wartość będącą prawdą lub fałszem

```
var zmienna = true;
var zmienna = false;
```

Znakowe - zawierające wartość będącą ciągiem znaków (stringiem)

```
var zmienna = "Moje imie to Grzegorz";
var zmienna = "Jakiś tekst";
```

undefined - zmienna niezadeklarowana. Ten typ zmiennej służy do sprawdzania, czy zmienna w ogóle istnieje.

Undefined and Null

The value of a variable with no value is **undefined**.

Variables can be emptied by setting the value to **null**.

Example

```
var cars;              // Value is undefined
var person = null;     // Value is null
```

obiekt Number

Javascript udostępnia obiekt **Number**, który zawiera takie wartości jak:

- **MAX_VALUE** - maksymalna wartość
- **MIN_VALUE** - minimalna wartość
- **NaN** - nie liczba
- **NEGATIVE_INFINITY** - specjalna wartosć nieskończoności (zwracana w przypadku overflow)
- **POSITIVE_INFINITY** - specjalna ujemna wartosć nieskończoności (zwracana w przypadku overflow)

JavaScript numbers can be written with, or without decimals:

Example

```
var x = 34.00;     // A number with decimals
var y = 34;        // A number without decimals
```

Extra large or extra small numbers can be written with scientific (exponent) notation:

Example

```
var x = 123e5;     // 12300000
var y = 123e-5;    // 0.00123
```

POSITIVE_INFINITY Property

Example

Return positive infinity:

```
Number.POSITIVE_INFINITY;
```

The result will be:

```
Infinity
```

Always Use Number.POSITIVE_INFINITY

POSITIVE_INFINITY a static property of the JavaScript Number object.

You can only use it as Number.POSITIVE_INFINITY.

Using x.POSITIVE_INFINITY, where x is a number or a Number object, will return undefined:

Example

```
var x = 100;
x.NEGATIVE_INFINITY;
```

The value of x will be:

```
undefined
```

Infinity

Infinity (or -Infinity) is the value JavaScript will return if you calculate a number outside the largest possible number.

Example

```
var myNumber = 2;
while (myNumber != Infinity) {          // Execute until Infinity
    myNumber = myNumber * myNumber;
}
```

Division by 0 (zero) also generates Infinity:

Example

```
var x =  2 / 0;          // x will be Infinity
var y = -2 / 0;          // y will be -Infinity
```

```
<script>
    var x = 2/0;
    var y = -2/0;
    document.write(x+"<br>");
    document.write(y+"<br>");
</script>
```

Infinity
-Infinity

Infinity is a number: typeOf Infinity returns number.

Example

```
typeof Infinity;        // returns "number"
```

NaN - Not a Number

NaN is a JavaScript reserved word indicating that a value is not a number.

Trying to do arithmetic with a non-numeric string will result in NaN (Not a Number):

Example

```
var x = 100 / "Apple";  // x will be NaN (Not a Number)
```

However, if the string contains a numeric value , the result will be a number:

Example

```
var x = 100 / "10";     // x will be 10
```

You can use the global JavaScript function isNaN() to find out if a value is a number.

Example

```
var x = 100 / "Apple";
isNaN(x);               // returns true because x is Not a Number
```

Watch out for NaN. If you use NaN in a mathematical operation, the result will also be NaN.

Example

```
var x = NaN;
var y = 5;
var z = x + y;          // z will be NaN
```

NaN is a number: typeOf NaN returns number.

Example

```
typeof NaN;             // returns "number"
```

Konwersja danych

JavaScript nie wymaga od ciebie abyś deklarował typ zmiennych. Przykładowo możesz utworzyć zmienną typu liczbowego o nazwie np. zmienna:

```
var zmienna = "to jest napis " + 20; //zwróci "to jest napis 20"
```

```
var zmienna = "20" + 1; //zwróci "201"
```

Gdy od zmiennej typu znakowego w której skład wchodzą tylko znaki cyfr odejmiemy zmienną typu liczbowego wówczas wykonamy normalne równanie:

```
var zmienna = "21" - 1; //zwróci 20
```

```
var zmienna = "20a" - 1; //zwróci NaN
```

Gdy od zmiennej typu znakowego w której skład wchodzą nie tylko znaki cyfr ale i liczb odejmiemy zmienną typu liczbowego wówczas otrzymanym wynikiem będzie NaN (Not-A-Number)

Możemy też wymusić konwersję poprzez rzutowanie wartości na dany typ danych za pomocą funkcji: **parseInt()**, **parseFloat()**, **String()**

Operatory

- **Operatory arytmetyczne:**

Operator	Opis	Przykład	Wynik
+	Dodawanie	x=3 x=x+4	7
-	Odejmowanie	x=4 x=6-x	2
*	Mnożenie	x=3 x=x*5	15
/	Dzielenie	10/5 9/2	2 4.5
%	Modulo (reszta z dzielenia)	4%3 12%8 8%2	1 4 0
++	Inkrementacja (zwiększanie o 1)		x=3
--	dekrementacja(zmniejszanie o 1)	x=4 x--	x=3

Wyrażenia i operatory

- **Operatory przypisania**

Operator	Przykład	Równoważne z
=	x=y	
+=	x+=7	x=x+7
-=	x-=3	x=x-3
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

- **Operatory relacyjne**

Operator	Opis	Przykład
==	jest równe	2==3 wynik:fałsz
!=	nie jest równe	2!=3 wynik:prawda
>	jest większe	25>3 wynik:prawda
<	jest mniejsze	2<3 wynik:prawda
>=	większe lub równe	25>=3 wynik:prawda
<=	mniejsze lub równe	2<=3 wynik:prawda

Wyrażenia i operatory

- Operator === jest operatorem identyczności, działa jak ==, ale bardziej rygorystycznie oprócz sprawdzania wartości operandów sprawdzany jest również ich typ, nie są wykonywane żadne konwersje typów.

```
"1" == true //true
"1" === true //false
1.0 == 1 //true
1.0 === 1 //true - ten sam typ Number
```

- **Operatory logiczne**

Operator	Opis	Przykład
&&	i	x=3 y=4 (x < 9 && y > 2) wynik:prawda
||	lub	x=3 y=4 (x==8 || y==6) wynik:fałsz
!	zaprzeczenie	x=3 y=4 !(x==y) wynik:prawda

Wyrażenia i operatory

- **Operatory bitowe:**

Operator	Description	Example	Same as	Result	Decimal
&	AND -bitowa koniunkcja	x = 5 & 1	0101 & 0001	0001	1
|	OR -bitowa alternatywa	x = 5 | 1	0101 | 0001	0101	5
~	NOT	x = ~ 5	~0101	1010	10
^	XOR-bitowa alternatywa wyłaczajaca	x = 5 ^ 1	0101 ^ 0001	0100	4
<<	Left shift – przesunięcie bitowe w lewo o podaną liczbę miejsc	x = 5 << 1	0101 << 1	1010	10
>>	Right shift - przesunięcie bitowe w prawo o podaną liczbę miejsc	x = 5 >> 1	0101 >> 1	0010	2
a>>>b	Zero-fill right shift -				
	Przesuwa bity w *a* o *b* bitów w prawo usuwając nadmiarowe bity z prawej strony i dodając zera z lewej				

- **Operatory specjalne:**

+ konkatenacja - łączenie łańcuchów znaków,
delete – usuwanie obiektu, własności obiektu, bądź elementu tablicy,
new – tworzenie obiektu,
this słowo kluczowe umożliwiające odwołanie się do obiektu wywołującego metodę,
typeof – typ obiektu (boolean, string, number, object,
void -nie zwraca wartości,

Priorytety operatorów

W tabeli poniżej przedstawiono operatory i ich ważność im wyższa pozycja w tabeli tym ważniejszy operator. Operatory znajdujące się w jednym wierszu maja taki sam priorytet.

L.p	Operator	Symbole
1	indeks tablicy, wywołanie funkcji	[], ()
2	inkrementacja i dekrementacja, ustalenie znaku, negacja bitowa i logiczna, utworzenie obiektu, ustalenie typu zmiennej, usunięcie składowej	++, --, +, -, ~, !, new,typeof, delete
3	mnożenie, dzielenie, reszta z dzielenia	*, /, %
4	dodawanie, odejmowanie	+, -
5	przesunięcie bitowe, w lewo, w prawo, w prawo z wypełnieniem zerami	<<, >>, >>>
6	mniejsze, większe, mniejsze lub równe, większe lub równe, porównanie typów	<, >, <=, >=
7	równe, różne	==, !=
8	iloczyn bitowy	&
9	bitowa różnica symetryczna	^
10	suma bitowa	|
11	iloczyn logiczny	&&
12	suma logiczna	||
13	warunkowy	? :
14	operatory przypisania	=, +=, -=, *=, /=, %=, &=, ^=,|=, <<=, >>=, >>>=
15	rozdzielanie wyrażeń	,

Operator warunkowy

Operator warunkowy pozwala na ustalenie wartości wyrażenia w zależności od prawdziwości danego warunku. Ma on postać:

warunek ? wartość1 : wartość2

Użycie operatora warunkowego

```
<script type="text/javascript">
var liczba = 100;
var wynik = (liczba < 0) ? -1 : 1;
document.write(wynik);
</script>
```

operator warunkowy ma mniejszy priorytet niż operatory relacyjne. W związku z tym nawias okrągły, wprowadzony do wyrażenia w celu zwiększenia czytelności zapisu, może zostać pominięty i może mieć ono również postać:

```
liczba < 0 ? -1 : 1;
```

Operator warunkowy

Operator warunkowy pozwala na ustalenie wartości wyrażenia w zależności od prawdziwości danego warunku. Ma on postać:

warunek ? wartość1 : wartość2

Użycie operatora warunkowego

```
<script type="text/javascript">
var liczba = 100;
var wynik = (liczba < 0) ? -1 : 1;
document.write(wynik);
</script>
```

operator warunkowy ma mniejszy priorytet niż operatory relacyjne. W związku z tym nawias okrągły, wprowadzony do wyrażenia w celu zwiększenia czytelności zapisu, może zostać pominięty i może mieć ono również postać:

```
liczba < 0 ? -1 : 1;
```

Zadania

Liczbę 1 otrzymamy jako wynik działania (javaScript)

a) 7 / 2
b) (7+ 2)/4
c) 7 % 2
d) 7 * 2

Wynikiem działania programu:

```
var a=7, b=3, w=1;
w+=(a++)+(++b);
document.write( w+ "   " +a);
```

jest wyświetlenie liczb:
a) 11 i 8 b) 12 i 7 c) 12 i 8 d) 13 i 8 /odp c

W języku programowania JavaScript błędną nazwą zmiennej jest zapis

a) _7liczb
b) Archiwum_nr_321
c) 12Liczb
d) Int_ /odp c

Zadania

```
(zadanie z egzaminu teoretycznego czerwiec 2014)
Które wielkości będą kolejno wypisane w wyniku działania
przedstawionego skryptu?
  <SCRIPT language=”JavaScript”>
 var x=1;
var y;
/*0*/   ++y;
/*1*/ document.write(++x);
/*2*/ document.write("  ");
/*3*/ document.write(x--);
/*4*/ document.write("  ");
/*5*/ document.write(x);
          </SCRIPT>
A.   2  2  1   B.  2  1  1     C.  1  2  1     D.  1  2  2

odp A
```

(zadanie z egzaminu teoretycznego czerwiec 2014)
Typ zmiennej w języku JavaScript
A. nie występuje B. jest tylko jeden C. następuje poprzez przypisanie wartości
D. musi być zadeklarowany na początku skryptu

Zadania

```
(zadanie z egzaminu teoretycznego czerwiec 2014)
Które wielkości będą kolejno wypisane w wyniku działania
przedstawionego skryptu?
  <SCRIPT language=”JavaScript”>
 var x=1;
var y;
/*0*/   ++y;
/*1*/ document.write(++x);
/*2*/ document.write("  ");
/*3*/ document.write(x--);
/*4*/ document.write("  ");
/*5*/ document.write(x);
        </SCRIPT>
A.   2  2  1   B.  2  1  1     C.  1  2  1     D.  1  2  2

odp A
```

(zadanie z egzaminu teoretycznego czerwiec 2014)
Typ zmiennej w języku JavaScript
A. nie występuje B. jest tylko jeden C. następuje poprzez przypisanie wartości
D. musi być zadeklarowany na początku skryptu

cd. wykładu:

Plik- z8_algorytm_js_instrukcjeWarunkowe_instrukcjaWyboru