

# ***TABLICE w JS przydatne strony***

[http://www.w3schools.com/js/js\\_arrays.asp](http://www.w3schools.com/js/js_arrays.asp)

<http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-2000>

<http://kursjs.pl/kurs/slice.html> <http://kursjs.pl/kurs/hierarchia/hierarchia.html>

[http://www.w3schools.com/jsref/jsref\\_obj\\_array.asp](http://www.w3schools.com/jsref/jsref_obj_array.asp)

13//  
C

# **TABLICE w JS**

**Tablica w JS to struktura danych, która przechowuje wiele elementów niekoniecznie tego samego typu. Np., gdy potrzebujemy przechowywać 1000 liczb, to możemy stworzyć tysiąc zmiennych (szkoda czasu), albo jedną tablicę składającą się z 1000 komórek, do których będziemy odwoływać się przez indeks.**

**Tablice w języku JavaScript działają na podobnych zasadach jak w innych językach programowania, jednak tutaj programista może dużo więcej. Tablice to w JS bardzo rozbudowany mechanizm. Korzyści jakie niesie za sobą stosowanie z tablic w JS w porównaniu do innych języków to:**

- nie trzeba na początku przewidywać rozmiaru tablicy**
- dane przechowywane w komórkach tablicy nie muszą być tego samego typu,**
- tablice można w dowolnej chwili rozszerzać i zmniejszać**
- dostęp do indeksu wykraczającego poza tablicę (przy zapisie) nie powoduje błędu, a automatyczne rozszerzenie tej tablicy.**

# Tworzenie nowej tablicy

```
var NazwaTablicy = new Array()  
//lub  
var NazwaTablicy = []
```

Konstruktor

literal

1. utworzenie pustej tablicy:

```
var tablica = new Array();
```

2. utworzenie tablicy i podanie jej rozmiaru (np. 10 elementów):

```
var tablica = new Array(10);
```

3. utworzenie tablicy i podanie listy elementów które mają się w niej znaleźć:

```
var tablica = new Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
```

• utworzenie pustej tablicy:

```
var tablica = [ ];
```

• utworzenie tablicy zawierającej jeden element:

```
var tablica = [ 10 ];
```

• utworzenie tablicy zawierającej więcej elementów:

```
var tablica = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ];
```

```
var NazwaTablicy = new Array('Marcin', 'Ania', 'Agnieszka')
```

```
//lub
```

```
var NazwaTablicy = ['Marcin', 'Ania', 'Agnieszka']
```

# Właściwości i metody klasy Array

- Właściwości

- length – ilość elementów w tablicy

```
var tab=['jeden', 'dwa', 'trzy', 'cztery', 'pięć'];  
var n1=tab.length;  
for(i=0; i<n1;i++){  
document.write(i+"--- "+tab[i]+"<br>");  
}
```

0--- jeden  
1--- dwa  
2--- trzy  
3--- cztery  
4--- pięć

```
var tab=['jeden', 'dwa', 'trzy', 'cztery', 'pięć'];  
for(klucz in tab){ document.write(klucz+"---"+tab[klucz]+"<br>");}
```

0---jeden  
1---dwa  
2---trzy  
3---cztery  
4---pięć

- toString() – zamienia tablicę na napis

```
var tab=['jeden', 'dwa', 'trzy', 'cztery', 'pięć'];  
var strTab=tab.toString();  
document.write(strTab+"<br>");
```

jeden,dwa,trzy,cztery,pięć

# Właściwości i metody klasy Array

```
var tab10=[];  
for (i=0;i<5;i++){  
  tab10[i]=prompt("podaj element tablicy","");  
}  
var strTab10=tab10.toString();  
document.write(strTab10+"<br>");
```

# Właściwości i metody klasy Array

## • Metody

- concat(tab2, tab3, ... tabN)

```
var str2=tabSlice.toString();  
var tab=['jeden', 'dwa', 'trzy','cztery','pięć'];  
var tab2=[6, 7, 8, 9];  
var tab3=tab.concat(tab2,["ala", "ola"]);  
var strTab3=tab3.toString();  
document.write(strTab3+"<br>");
```

jeden,dwa,trzy,cztery,pięć,6,7,8,9,ala,ola

Następujący kod łączy dwie tablice:

```
1 alpha = new Array("a", "b", "c");  
2 numeric = new Array(1, 2, 3);  
3 alphaNumeric = alpha.concat(numeric); // tworzy tablicę ["a", "b", "c", 1, 2, 3]
```

Następujący kod łączy trzy tablice:

```
1 num1 = [1, 2, 3];  
2 num2 = [4, 5, 6];  
3 num3 = [7, 8, 9];  
4 nums = num1.concat(num2,num3) // tworzy tablicę [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Następujący kod łączy trzy wartości z tablicą:

```
1 alpha = ['a', 'b', 'c'];  
2 alphaNumeric = alpha.concat(1, 2, 3); // tworzy tablicę ["a", "b", "c", 1, 2, 3]
```

# Właściwości i metody klasy Array

join(separator) – wszystkie elementy tablicy są łączone w jeden napis; elementy są pooddzielane separatorem (domyślnie ",")

```
var tab=['jeden', 'dwa', 'trzy','cztery','pięć'];  
var str1=tab.join('+');  
document.write(str1+"<br>");
```

jeden+dwa+trzy+cztery+pięć

# Właściwości i metody klasy Array

- `pop()` – usuwa i zwraca ostatni element tablicy  

```
colours = ["red", "black", "white"];  
kolorek = colours.pop(); // "white"
```
- `push(elem1,...,elemN)` – wstawia elementy na koniec tablicy i zwraca nowy rozmiar  

```
colours = ["red", "black"];  
ile = colours.push("white"); // 3
```
- `reverse()` – odwraca kolejność elementów  

```
a = new Array("Jeden", "Dwa", "Trzy")  
a.reverse() // ["Trzy", "Dwa", "Jeden"]
```
- `shift()` – usuwa i zwraca pierwszy element tablicy  

```
colours = ["red", "black", "white"];  
kolorek = colours.pop(); // "red"
```
- `unshift(element1,..., elementN)` – wstawia elementy na początek tablicy  

```
colours = ["red", "black"];  
ile = colours.unshift("white"); // 3
```



# Właściwości i metody klasy Array

## Tworzenie nowej tablicy za pomocą metody slice()

Metoda **slice(od,ile)** "tnie" naszą tablicę i tworzy w ten sposób nową.

Pierwszy parametr **od** wskazuje na index (może być ujemny - wtedy będzie liczony od końca tablicy), od którego ma "wyciąć" elementy, a parametr **ile** wskazuje ile elementów ma być "wyciętych" (licząc od początku tablicy):

```
var Tablica = newArray('Marcin', 'Ania', 'Agnieszka');
```

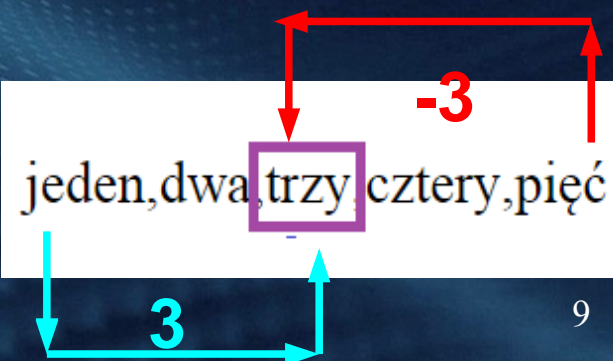
```
var Tablica_2 = Tablica.slice(0,1)
document.write(Tablica_2.join()); //wypisze się
"Marcin"
```

```
var Tablica_2 = Tablica.slice(0,2)
document.write(Tablica_2.join()); //wypisze się
"Marcin,Ania"
```

```
var tabSlice3=tab.slice(-3,3);
var str3=tabSlice3.toString();
document.write(str3+"<br>");
```

```
var tab=['jeden', 'dwa', 'trzy', 'cztery', 'pięć'];
var tabSlice=tab.slice(1,3);
var str2=tabSlice.toString();
document.write(str2+"<br>");
```

dwa,trzy



# Właściwości i metody klasy Array

The `slice()` method returns the selected elements in an array, as a new array object.

The `slice()` method selects the elements starting at the given *start* argument, and ends at, *but does not include*, the given *end* argument.

**Note:** The original array will not be changed.

```
array.slice(start, end)
```

## Parameter Description

<i>start</i>	Optional. An integer that specifies where to start the selection (The first element has an index of 0). Use negative numbers to select from the end of an array. If omitted, it acts like "0"
<i>end</i>	Optional. An integer that specifies where to end the selection. If omitted, all elements from the start position and to the end of the array will be selected. Use negative numbers to select from the end of an array

```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];  
var citrus = fruits.slice(1, 3);
```

The result of *citrus* will be:

Orange, Lemon

```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];  
var myBest = fruits.slice(-3, -1);
```

The result of *myBest* will be:

Lemon, Apple

# Właściwości i metody klasy Array

## Sortowanie tablic za pomocą metody sort()

```
1 var Tablica = new Array('Marcin', 'Ania', 'Piotrek', 'Grześ')
2 Tablica.sort()
3 document.write( Tablica.join() ); //wypisze się "Ania,Grześ,Marcin,Piotrek"
4
```

# Operator in

## Pętle wykonywane po tablicach

Kolejną bardzo pomocną instrukcją jest **in**, która pozwala wykonać dany kod dla każdej składowej danego obiektu w tablicy (podobne działanie ma funkcja `foreach` w php):

```
var t = ['pierwszy', 'drugi', 'trzeci'];  
t[1000] = 'dwudziesty';  
  
for (var obj in a) {  
    document.write(a[obj]); //dzięki temu pętla  
    wykona się 4 razy a nie 1000  
}
```

# Funkcje operujące na napisach -cd

## Napisy

JS udostępnia m.in. następujące funkcje do obsługi napisów:

- `napis = "ab" + "ca"` // łączenie (konkatenacja) napisów
- `napis == 'abca'` // porównanie true
- `napis[3]` // czwarta litera napisu, czyli a
- `napis.length` // długość napisu, czyli 4
- `napis.indexOf('b')` // pozycja pierwszego wystąpienia  
// 'b' w napisie, czyli 1
- `napis.indexOf('e')` // 'e' nie występuje w napisie,  
// więc zwracane jest -1
- `napis.lastIndexOf('a')` // ostatnie wystąpienie 'a'  
// w napisie 3
- `napis.substr(1,2);` // od pozycji 1 (czyli drugiej  
// litery) zwróć dwie litery,  
// czyli 'bc'