Lekcja 5

Temat: Tablice

Tworzenie tablic

Tablice to struktury danych pozwalające na przechowywanie uporządkowanego zbioru elementów. Najprostszą tablicę można sobie wyobrazić jako zbiór elementów, taki jak zaprezentowany na rys. Poszczególne pola danych nazywamy komórkami tablicy, a każda komórka ma identyfikujący ją indeks (mówimy wtedy o **tablicach indeksowanych numerycznie**). Pierwsza komórka (zawierająca *wartość 1*) ma indeks 0, druga — indeks 1, trzecia — indeks 2 itd.

Tablicę można utworzyć za pomocą

1) **literału tablicowego []**, korzystając z konstrukcji o następującej postaci:

```
var nazwa_tablicy = [element1, element2, ..., elementN];
```

W ten sposób powstaje *N*-elementowa tablica, w której w kolejnych komórkach zostały zapisane poszczególne elementy wymienione w nawiasie kwadratowym. Wartość każdej komórki może być dowolnego typu. Można utworzyć pustą tablicę, do której w dalszej części skryptu będą zapisywane dane. Wystarczy użyć instrukcji:

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

Składnia z nawiasem kwadratowym dopuszcza pominięcie w definicji niektórych elementów. Powstają wtedy komórki o wartości undefined (niezdefiniowane). Pusty element zostanie utworzony, jeśli pomiędzy dwoma przecinkami nie umieści się żadnej wartości. I tak instrukcja:

```
var tablica = [1,,3,,5];
```

spowoduje utworzenie tablicy o strukturze:

1	undefined	3	undefined	5

2) **konstruktora** new Array() - w obecnych wersjach JavaScriptu tablice są obiektami, do ich konstrukcji oprócz przedstawionego wyżej literału można również używać formalnego wywołania konstruktora typu `Array`.

Możliwe są trzy typy takiego wywołania:

- `new Array()` — powstanie pusta tablica; jest to odpowiednik zapisu [].
- `new Array(liczba_elementów)` — powstanie tablica o określonej liczbie pustych elementów (zawierających wartość undefined); jest to odpowiednik zapisu z przecinkami, np. new Array(2) to to samo co [,,]
- `new Array(element1, element2,…,elementN)` — powstanie tablica zawierająca określone elementy; jest to odpowiednik zapisu [*element1*, *element2*,…,*elementN*].

Należy przy tym zwrócić szczególną uwagę na drugi sposób tworzenia i nie mylić go z trzecim w przypadku tworzenia tablicy z jednym elementem. Zapis:

```
var tablica = new Array(2);
```

oznacza utworzenie tablicy o dwóch pustych elementach. Z kolei zapis:

```
var tablica = new Array(1,2);
```

oznacza powstanie tablicy o dwóch elementach, z których pierwszy będzie miał wartość 1, a drugi — wartość 2.

UWAGA: Wykorzystanie literału jest łatwiejsze i zalecane.

Odczyt i zapis tablic

Aby odczytać zawartość tablicy możesz zastosować zwykłe wyświetlenie jej jako zmiennej:

Np. : `console.log(tablica)`

Odczyt kolejnych danych z tablicy jest równie prosty. Wystarczy podać indeks żądanego elementu w nawiasie kwadratowym:

```
var zmienna = tablica[indeks];
```

Należy przy tym pamiętać, że indeksowanie rozpoczyna się od 0.

Aby zmienić zawartość istniejącej komórki lub też utworzyć nową komórkę, należy użyć zwykłego operatora przypisania:

```
tablica[indeks] = wartość;
```

Jeżeli komórka o indeksie wskazanym przez parametr *indeks* istnieje, jej wartość zostanie zmieniona, jeżeli zaś nie istnieje, zostanie utworzona.

Każda tablica posiada właściwość **length**, która pozwala określić liczbę komórek (inaczej: długość tablicy, rozmiar tablicy). To znaczące ułatwienie, np. pisząc:

```
var zmienna = nazwa_tablicy.length;
```

można uzyskać liczbę jej elementów.

Przykład 1. Przygotuj strony html wykorzystujące poniższy kod. Zapisz przykłady jako *l5p1.html*

Sprawdź w konsoli efekt działania skryptu, rozwiń zawartość tablic strzałką . Zwróć uwagę na właściwość **length.**

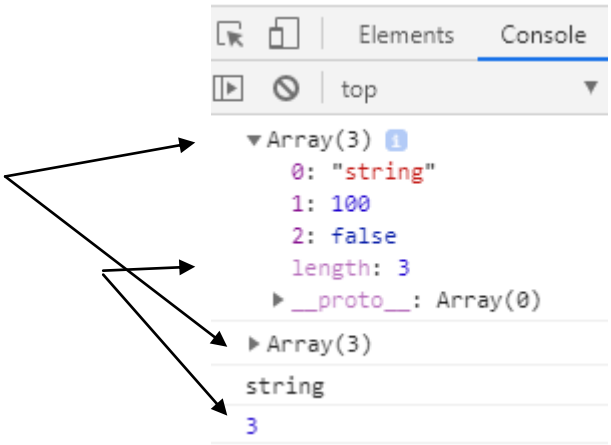
```
<!DOCTYPE HTML>
<html lang="pl">
<head>
    <meta charset="utf-8"/>
    <title>t5pl</title>
</head>
<body>
    <script>
        var myArray = ["string",100,false]; // literał tablicowy
        var myArray3 = []; // pusta tablica

        console.log(myArray);

        var myArray1= new Array("string",100,false); // konstruktor
        var myArray2 = new Array(); // pusta tablica
        console.log(myArray1);

        console.log(myArray[0]); // wyświetlenie pierwszego elementu
        console.log(myArray.length);// długość tablicy
    </script>
</body>

</html>
```



Przykład 2. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p2.html.*

```
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="utf-8">
    <title>Tablice - t5p2</title>

</head>
<body>
    <script>
        //różne typy danych w tablicach
        var numbers = [1, 2, 3];
        var strings = ["czerwony", "zielony", "niebieski"];
        var mix = [1, 2,, "czerwony",, "niebieski", false];
        console.log("liczby: " + numbers[0] + ", " + numbers[1] + ", " + numbers[2]);
        console.log("napisy: " + strings[0] + ", " + strings[1] + ", " + strings[2]);
        console.log("miks: " + mix[0] + ", " + mix[1] + ", " + mix[2] + ", " + mix[3] +
        ", " + mix[4] + ", " + mix[5]+", "+ mix[6]);
        console.log(numbers);
        console.log(strings);
        console.log(mix);
    </script>
</body>
</html>
```

Przykład 3. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p3.html.*

```
<script>
    var names = ["Rafał", "Joanna", "Piotr", "Kuba", "Agata"];
    console.log(names[0]);
    console.log(names[1]);
    console.log(names);
    //zmiana zawartości istniejącej komórki
    names[1]="Szymon";
    console.log(names[0]);
    console.log(names[1]);
    console.log(names);
    //dopisanie kolejnej wartości
    names[5]="Tomasz";
    console.log(names);
    // wstawienie odległego elementu
    names[100]="Anna";
    console.log(names);// zwróć uwagę na długość tablicy
</script>
```

Przykład 4. Wykorzystanie właściwości **length** . Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p4.html.*

```
<script>
    // wykorzystanie właściwości length
    var task = ["wynieś śmieci","przeczytaj książkę","napisz javascript","idź do pracy","śpij"];
    var progress = ["gotowe","w toku","zrobione"];
    console.log(task);
    console.log(task[(task.length-1)]);//wyświetlenie ostatniego elementu tablicy

    task[task.length] = "oglądaj TV";// dodanie elementu na końcu tablicy
    console.log(task);

    var question1 = prompt("które zadanie 1-"+task.length+"?");
    var question2 = prompt("status 1 = gotowe 2 = w toku 3 = zrobione?");

    console.log("zadanie na dziś: " + task[(question1-1)] +
    " status zadania: " + progress[(question2-1)]);
</script>
```

Pamiętaj, że wartości indeksu tablicy pochodzą z przedziału od 0 do wartości o jeden mniejszej od wartości zwracanej przez właściwość `length` tablicy. A właściwość `length` zwraca liczbę wszystkich elementów tablicy.

Wyświetlanie zawartości tablicy

Przykład 5. **wyświetlenie zawartości tablicy w pętli**. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p5.html.*

```
<script>
    var names = ["Rafał", "Joanna", "Piotr", "Kuba", "Agata"];
    //wyświetlenie  zawartości tablicy w pętli
    for(var i = 0; i < 5; i++){
    console.log( names[i]);
    }
    // drugi sposób- wykorzystanie length
    console.log("Drugi sposób");
    for(var i = 0; i < names.length; i++){
      console.log(names[i]);
    }

</script>
```

Przykład 6. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p6.html.*

```
<body>
    <ul id="num1">
    </ul>
    <script>
        var numbers = [];// utworzenie pustej tablicy
        // wypełnienie jej danymi
        for(var i = 0; i < 20; i++){
          numbers[i] = 100 - 3*i;
        }
        // wyświetlenie  zawartość tablicy
        var listA=document.getElementById("num1");
        for(var i = 0; i < numbers.length; i++){
          listA.innerHTML+=("<li>"+numbers[i] + "</li> ");
        }
    </script>
</body>
```

Przykład 7. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p7.html.*

```
<body>
    <p id="output">
    </p>
    <script>
    var numbers = [];
    /* poniższa pętla wyświetla okno dialogowe i zapamiętuje podawane
    przez użytkownika liczby w tablicy*/
    for(var i = 1; i <= 5; i++){
        numbers[i-1]=parseFloat(prompt("Podaj 5 liczb, liczba nr " +i+" :"));
      }
    // pętla wyświetlająca zawartość tablicy
    var out=document.getElementById("output");
    var i = 0;
    while(i < numbers.length) {
        out.innerHTML+=(numbers[i] + ",  ");
        i++;
    }
    </script>
</body>
```

Przykład 8. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p8.html.*

```
<body>
    <p id="output">
    </p>
    <script>
    /* skrypt losujący 10 liczb z zakresu 1 do 10 i zapisujący
    wylosowane liczby w tablicy. */
    var numbers = [];
    for(var i = 0; i <10; i++){
        numbers[i]=Math.floor((Math.random()*10)+1);
      }
    var out=document.getElementById("output");
    for(var i = 0; i < numbers.length; i++){
        out.innerHTML+=(numbers[i] + ",  ");
    }
    </script>
</body>
```

Przykład 9. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p9.html.*

```
<body>
    <script>
    // instrukcje continue i  break
    var jan = ['Jan', 'Kowalski', 1990, 'kurier', false, 'niebieski'];
    console.log("Pierwsza:");
    for (var i = 0; i < jan.length; i++) {
        if (typeof jan[i] !== 'string') continue;
        console.log(jan[i]);
    }
    console.log("Druga:");
    for (var i = 0; i < jan.length; i++) {
        if (typeof jan[i] !== 'string') break;
        console.log(jan[i]);
    }
    console.log("Trzecia:");
    // pętla odwrotna
    for (var i = jan.length - 1; i >= 0; i--) {
        console.log(jan[i]);
    }
    </script>
</body>
```

Wybrane metody obiektu array

Do wstawienia elementu na końcu tablicy użyliśmy w przykładzie 4 właściwości length. Możemy w tym celu zastosować jedną z metod obsługujących tablice.

Właściwość	Krótki opis
concat()	Metoda ta umożliwia łączenie elementów dwóch lub większej liczby tablic. Jako parametr podajemy nazwę tablicy, której elementy chcemy dołączyć do tablicy, dla której metoda jest wywoływana.
fill()	Wypełnia wszystkie elementy tablicy stałą wartością podaną jako parametr.
indexOf()	Przeszukuje tablicę dla elementu podanego jako parametr i zwraca jego pozycję.
join()	Metoda ta służy do łączenia elementów tablicy w jeden tekst. Jako parametr możemy podać separator, który oddziela poszczególne elementy.

lastIndexOf()	Przeszukuje tablicę dla elementu podanego jako parametr, zaczynając od końca i zwraca jego pozycję.
pop()	Usuwa ostatni element tablicy i zwraca jego wartość.
push()	Metoda pozwala wstawić elementy na końcu tablicy. Przykład: zwierzaki.push("kot","pies") spowoduje dodanie do tablicy zwierzaki dwóch elementów o wartościach "kot" i "pies".
reverse()	Metoda ta służy do odwrócenia kolejności występowania elementów tablicy.
shift()	Metoda ta usuwa pierwszy element tablicy i go zwraca.
slice()	Metoda ta umożliwia usuwanie elementów tablicy od indeksu a do indeksu b, podawanych jako parametry - *tablica.slice(1,3)* usunie elementy o indeksach od 1 do 3.
sort()	Metoda *sort()* umożliwia sortowanie tablicy. Jako parametr możemy podać własną funkcję sortująca elementy tablicy. Domyślnie przyjmowane jest sortowanie leksykalne (jak w słowniku).
splice()	Metoda ta umożliwia usunięcie elementów tablicy od indeksu *a*, który podajemy jako parametr. Drugim parametrem jest liczba elementów do wycięcia. Jeżeli podamy kolejne parametry, to będą to wartości jakie powinny zastąpić usuwane elementy tablicy. Np. *tablica.splice(2,1,"kot")* usunie 1 element o indeksie 2 i zastąpi go elementem *"kot"*. Uwaga, jeżeli drugi parametr jest równy zeru, to metoda pozwala wstawić element tablicy w dowolnym miejscu, bez usuwania czegokolwiek.
toString()	Konwertuje tablicę na tekst (string) i zwraca jego wartość.
unshift()	Metoda ta wstawia nowe elementy tablicy podane jako parametry metody i zwraca nową długość tablicy.

Przykład 10. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p10.html.*

```
<script>
    var animals1 = ["kot","pies","królik","ptak","ryba","zebra"];
    animals1.push("owca"); // dodaje element na końcu tablicy
    console.log(animals1);

    var animals2 = animals1;
    var remove1 = animals2.pop(); // usuwa ostatni element z końca tablicy
    console.log(animals2);

    var animals3=animals2;
    animals3.shift(); //usuwa pierwszy element
    console.log(animals3);

    var animals4=animals3;
    animals4.unshift("koń");//dodaje pierwszy element do tablicy
    console.log(animals4);

    var animals5=animals4;
    animals5.splice(2,3) ; // usuwa 3 elementy od pozycji 2
    console.log(animals5);

    var animals = ["kot","pies","królik","ptak","ryba","zebra"];
    var startVal = animals.length;
    console.log(startVal);
    animals[15] = "koń";
    animals.fill("mysz",startVal , (animals.length-1));
    // wypełnia danymi brakujące elementy tablicy
    console.log(animals);
</script>
```

Przykład 11. **indexOF**. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p11.html.*

```
<script>
    //indexOf
    var jan = ['Jan', 'Kowalski', 1990, 'kurier', false,
    'niebieski'];
    console.log(jan);
    console.log(jan.indexOf(1990));// pozycja parametru: 2
    console.log(jan.indexOf('kurier'));
    console.log(jan.indexOf(23));// brak parametru; pozycja: -1
    // wykorzystanie 1:
    var isProgrammer = jan.indexOf('programista') === -1 ? 'Jan
    nie jest programistą' : 'Jan jest programistą';
    console.log(isProgrammer);
    // wykorzystanie 2:
    var q = prompt("Co chcesz usunąć?");
    var finder = jan.indexOf(q);

    var v = (finder > -1) ? jan.splice(finder,1) : false;
    console.log(v);
    console.log(jan);
</script>
```

Przykład 12. **Sortowanie**. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p12.html.*

```
<script>
    var animals = ["kot","ryba","zebra","pies","królik","ptak"];
    console.log(animals);
    var animals2=animals.sort(); // sortuje elementy według
    wartości
    console.log(animals2);
    var animals3=animals2.reverse(); // odwraca kolejność elementów
    console.log(animals3);
    var animals4=animals.sort().reverse();// łączenie metod
    console.log(animals4);
    // sortowanie elementów różnego typu
    var arr = [3,4,5,"pies",77,33,22,4,1,55,2,"kot"];
    console.log(arr);
    var arr2=arr.sort();
    console.log(arr2);
</script>
```

Przykład 13. **Łączenie elementów tablicy./łączenie tablic** Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p13.html.*

```
<script>
    //Łączenie elementów tablicy
    var animals = ["kot","ryba","zebra","pies","królik","ptak"];
    console.log(animals.join());// bez separatora
    console.log(animals.join(" - ")); // z separatorem
    console.log(animals.join(" ; "));
    //łączenie dwóch tablic
    var animals1 = ["kot","ryba","zebra"];
    var animals2 = ["królik","ptak"];
    console.log(animals1);
    console.log(animals2);
    animals3 = animals1.concat(animals2);
    console.log(animals3);
</script>
```

Tablice wielowymiarowe

W języku JavaScript wartością komórek tablicy mogą być inne tablice. Możemy utworzyć tablice zawierające w komórkach inne tablice, które mogą zawierać kolejne tablice itd.

Dostęp do elementów w takich tablicach jest następujący: tablica[x][y] – Z tablicy pobierz wartość komórki oznaczonej indeksem x a z niej wartość komórki oznaczonej y

Przykład 14. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p14.html.*

```
<script>
    var arr = [["kot","pies","ryba"],[4,2,5,6,6,4],"ok", 3];// tablica zawierająca tablice
    console.log(arr);// rozwiń kolejne tablice w konsoli
    console.log(arr[0]);// pierwszy element talicy- tablica
    console.log(arr[0][2]);// trzeci element z pierwszej tablicy
    console.log(arr[2])

    var ourFriends = [["Ania","Tomek","Piotr"],[1980,1990,2016]];
    var q =prompt("który przyjaciel?");
    console.log("spotkałeś " + ourFriends[0][q] + " w " + ourFriends[1][q]);

</script>
```

Przykład 15. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako *l5p15.html.*

```
<body>
    <p id="ekranik"></p>
    <script>
        //deklaracja zmiennych
        var iloscKolumn = 10;
        var iloscWierszy = 10;
        var napis = '';
        var tablica = [];
        //deklaracja drugiego wymiaru tablicy
        for (var i = 0; i < iloscWierszy; i++) {
            tablica[i] = [];
        }
        //wypełnianie tablicy wartościami
        for (var i = 0; i < iloscWierszy; i++) {
            for (var j = 0; j < iloscKolumn; j++) {
                tablica[i][j] = i + ', ' + j;
            }
        }
        //odczytywanie tablicy i generowanie napisu
        for (var i = 0; i < iloscWierszy; i++) {
            for (var j = 0; j < iloscKolumn; j++) {
                napis += tablica[i][j] + " || ";
            }
            napis += '<br>';
        }
        //wyświetlenie napisu na stronie
        document.getElementById("ekranik").innerHTML = napis;
    </script>
</body>
```

Ćwiczenia

Ćwiczenie 1 Napisz skrypt, w którym wykonasz w tablicy spis artykułów spożywczych, które musisz zakupić, np.: *Mleko*, *Jajka*, *Płatki śniadaniowe*, *Salami* i *Sok*. Wyświetl następnie tę listę w postaci listy numerowanej i opatrz tytułem: *Lista zakupów*.

Zapisz skrypt pod nazwą l5cw1.html

Ćwiczenie 2 Przygotuj stronę**,** w której utworzysz tablicę, zawierającą wartości: BMW, Toyota, Opel i Skoda. Wyświetl wartości tablicy w postaci zdania:

Lubię samochody marki Toyota i Opel, nie lubię BMW i Skoda.

Dodaj do tablicy po jednym elemencie na początku i końcu tablicy. Użyj do tego odpowiednich metod. Po dodaniu elementów wypisz długość tablicy oraz całą tablicę w konsoli.

Zapisz skrypt pod nazwą l5cw2.html

Ćwiczenie 3

a) Przygotuj stronę**,** w której utworzysz tablicę, zawierającą następujące dane o Tobie lub dowolnej osobie: imię, nazwisko, wiek, wzrost, datę urodzenia, klasę. Wyświetl wartości tablicy w jednym wierszu tabeli z obramowaniem.
Zapisz skrypt pod nazwą l5cw3a.html

b) Zmodyfikuj skrypt tak, aby użytkownik podawał powyższe dane w oknie dialogowym. Dane powinny zostać następnie zapisywane w tablicy i wyświetlone na stronie w tabelce.
Wskazówka: Wykorzystaj dwie tablice: z danymi: imię, nazwisko, wiek, wzrost, data urodzenia, klasa; wczytywanymi następnie do okien dialogowych oraz tablicę na zapisywanie podawanych przez użytkownika wartości.

Zapisz skrypt pod nazwą l5cw3b.html

Ćwiczenie 4

a) Napisz skrypt wczytujący 5 liczb podawanych przez użytkownika do tablicy, a następnie wyświetlający je w kolejności odwrotnej do kolejności podawania.

Zapisz skrypt pod nazwą l5cw4a.html

b) Zmodyfikuj skrypt tak, aby użytkownik mógł decydować ile liczb będzie chciał podać.
Zapisz skrypt pod nazwą l5cw4b.html

Ćwiczenie 5 Napisz skrypt, który wczyta 5 liczb podawanych przez użytkownika do tablicy, a następnie (po wczytaniu wszystkich liczb) obliczy sumę tych liczb.

Zapisz skrypt pod nazwą l5cw5.html

Ćwiczenie 6 Zmodyfikuj skrypt z ćwiczenia 1 w taki sposób, aby to użytkownik podawał kolejne dane do listy zakupów w oknach dialogowych. Skrypt powinien następnie wprowadzone dane posortować alfabetycznie i wyświetlić w postaci listy numerowanej.

Przygotuj instrukcję sprawdzającą czy użytkownik nie zapomniał o potrzebie zakupu chleba. Wyświetl komunikat, np.: „Dodaj chleb" lub „ Ok, zapisałeś chleb"

Zapisz skrypt pod nazwą l5cw6.html

Ćwiczenie 7 Przygotuj dwie tablice. Jedną zawierającą 5 początkowych liczb parzystych, drugą 6 początkowych liczb nieparzystych. Złącz następnie obie tablice w jedną, posegreguj malejąco i wyświetl wartości oddzielając je od siebie znakami //.

Zapisz skrypt pod nazwą l5cw7.html

Ćwiczenie 8 Przygotuj tablicę zawierającą trzy podtablice, w których umieścisz dane trzech osób w układzie: imię, nazwisko, wiek, datę urodzenia, klasę.

Wyświetl następnie te dane na stronie w kolejnych wierszach, oddzielając dane dla każdej osoby spacjami.

Zapisz skrypt pod nazwą l5cw8.html