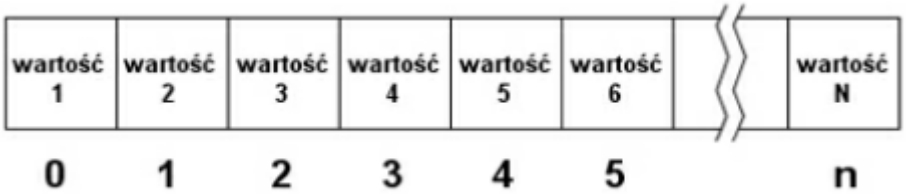
Lekcja 9

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. : `dokument.write(tablica)` lub `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. Zakres wartości indeksu w typowych implementacjach zawiera się w przedziale 0 – 232-1.

Przykład 1. Przygotuj strony html wykorzystujące poniższy kod.

```
<body>
    <p id="output"></p>
    <script>
        var imie = ["Rafał", "Joanna", "Piotr", "Kuba", "Agata"];
        document.write("<h3> Cała tablica</h3>");
        document.write(imie);
        console.log(imie);
        document.write("<h3> Poszczególne wartości:</h3>");
        document.write(imie[0]+"<br>");
        document.write(imie[1]+"<br>");
        document.write(imie[2]+"<br>");
        document.write(imie[3]+"<br>");
        document.write(imie[4]+"<br>");

        for(i=0; i<5; i++){
            console.log(`${imie[i]},`);
        }

        let output = document.getElementById("output");
        let contents='Wyświetlenie tablicy w akapicie: ';
        for (i = 0; i <5; i++) {

                contents+=`${imie[i]}, `;

        }
        output.innerHTML = contents;

    </script>
</body>
```

```
<body>
    <script>
        var imie = new Array("Rafał", "Joanna", "Piotr", "Kuba", "Agata");
        console.log(imie[0]+"<br>");
        document.write(imie[1]+"<br>");
    </script>
</body>
```

Przykład 2. Przygotuj stronę html wykorzystującą poniższy kod.

```
<body>
    <script>
        //różne typy danych w tablicach
        var liczby = [1, 2, 3];
        var napisy = ["czerwony", "zielony", "niebieski"];
        var mix = [1, 2,, "czerwony",, "niebieski"];
        document.write("liczby: " + liczby[0] + ", " + liczby[1] + ", " + liczby[2]);
        document.write("<br>");
        document.write("napisy: " + napisy[0] + ", " + napisy[1] + ", " + napisy[2]);
        document.write("<br>");
        document.write("miks: " + mix[0] + ", " + mix[1] + ", " + mix[2] + ", " + mix[3] + ", " + mix[4] + ", " + mix[5]);
    </script>
</body>
```

Lub

```
<body>
    <script>
        //różne typy danych w tablicach
        let numbers = [1, 2, 3];
        let strings = ["czerwony", "zielony", "niebieski"];
        let 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>
```

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

```
let zmienna = nazwa_tablicy.length;
```

można uzyskać liczbę elementów.

Przykład 3. Przygotuj stronę html wykorzystującą poniższy kod.

```
<body>
    <script>
        var imie = ["Rafał", "Joanna", "Piotr", "Kuba", "Agata"];
        //wyświetlenie  zawartości tablicy w pętli
        document.write("<h3>Imiona:</h3><br>");
        for(var i = 0; i < 5; i++){
        document.write( imie[i]+"<br>");
        }
        // drugi sposób- wykorzystanie length
        document.write("<h3>Imiona - drugi sposób:</h3><br>");
        for(var i = 0; i < imie.length; i++){
          document.write(imie[i]+"<br>");
        }

    </script>
</body>
```

Przykład 4. Przygotuj stronę html wykorzystującą poniższy kod.

```
<body>
    <script>
        var nameS = ["Rafał", "Joanna", "Piotr", "Kuba", "Agata"];
        console.log(nameS[0]);
        console.log(nameS[1]);
        //zmiana zawartości istniejącej komórki
        nameS[1]="Szymon";
        console.log("Imiona po zmianie:");
        console.log(nameS[0]);
        console.log(nameS[1]);
        //dopisanie kolejnej wartości
        nameS[5]="Marcin";
        console.log("Imiona po dopisaniu:");
        for(i=0; i<nameS.length; i++){
            console.log(`${nameS[i]},`);
        }
    </script>
</body>
```

Należy zwróć uwagę, że zakres pętli zaczyna się od 0 i kończy się na wartości o jeden mniejszej niż pełna długość tablicy (zwracana przez właściwość `length`). Jest tak dlatego, 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!

Przykład 5. Przygotuj stronę html wykorzystującą poniższy kod.

```
<body>
    <script>
        var liczby = [];// utworzenie pustej tablicy
        // wypełnienie jej danymi
        for(var i = 0; i < 10; i++){
        liczby[i] = i + 1;
        }
        // wyświetlenie zawartość tablicy
        for(var i = 0; i < liczby.length; i++){
          document.write(liczby[i] + ", ");
        }
        // zmiana zawartości, w kolejnych komórkach liczby od 100 do 81
        for(var i = 0; i < 20; i++){
          liczby[i] = 100 - i;
        }
        // wyświetlenie nowej zawartość tablicy
        document.write("<br>");
        for(var i = 0; i < liczby.length; i++){
          document.write(liczby[i] + ", ");
        }
    </script>
</body>
```

Przykład 6. Przygotuj stronę html wykorzystującą poniższy kod.

Zapis wartości parzystych i nieparzystych. Użycie pętli for do utworzenia tablicy, w której znajdą się wartości od 1 do 10 ułożone w taki sposób, aby najpierw umieszczone były liczby parzyste, a dopiero za nimi nieparzyste.

```
<body>
    <p id="output"></p>
    <script>
        let numbers1 = [];
        let ile = 10;
        let indP = 0, indN = 5;

        for(var i = 1; i <= ile; i++){
          if(i % 2 == 0){
            numbers1[indP++] = i;
          }
          else{
            numbers1[indN++] = i;
          }
        }

        let output = document.getElementById("output");
        let contents='Wyświetlenie tablicy w akapicie: ';
        for (i = 0; i <numbers1.length; i++) {

                contents+=`${numbers1[i]}, `;
        }
        output.innerHTML = contents;
    </script>
</body>
```

Indeks, od którego mają być wstawiane wartości parzyste, jest zapisany w zmiennej indP, a indeks dla wartości nieparzystych — w zmiennej indN. Początkowa wartość indP to zawsze 0, natomiast początkowa wartość indN wynika z podzielenia całkowitej liczby wartości przez 2. W tym przypadku, ponieważ wartości ma być 10, wartość zapisana w indN wynosi 5. Wewnątrz pętli for badane jest, czy bieżąca wartość jest podzielna przez 2, czyli czy wynikiem działania i % 2 jest 0. Jeśli tak, jest zapisywana w tablicy liczby pod indeksem wynikającym z wartości zmiennej indP, a wartość tej zmiennej jest zwiększana o 1 za pomocą przyrostkowej wersji operatora++. Dla wartości nieparzystych wykonywane jest analogiczne działanie z użyciem zmiennej indN.

Przykład 7. Przygotuj stronę html wykorzystującą poniższy kod.

```
<body>
    <p id="output"></p>
    <script>
        const numbers1 = [];
        /* poniższa pętla wyświetla okno dialogowe i zapamiętuje podawane przez
        użytkownika liczby w tablicy*/
        for(let i = 1; i <= 5; i++){
            numbers1[i-1]=parseFloat(prompt("Podaj liczbę nr " +i+" :"));
          }
          console.log(numbers1);
        // pętla wyświetlająca zawartość tablicy
        const output = document.getElementById("output");
        let contents='Wyświetlenie tablicy w akapicie: ';
        for (i = 0; i<numbers1.length; i++) {

                contents+=`${numbers1[i]}, `;
        }
        output.innerHTML = contents;
    </script>
</body>
```