JavaScript

Умовні і циклічні конструкції

План

1. Умовна конструкція if … else.
2. Тернарний оператор.
3. Оператор багатозначного вибору.
4. Цикли.
5. Масиви.
6. Функції.

Умовні конструкції

- Умовний оператор **if … else**
- Тернарний оператор **… ? … : …**
- Оператор багатозначного вибору **switch … case**

Тернарний оператор

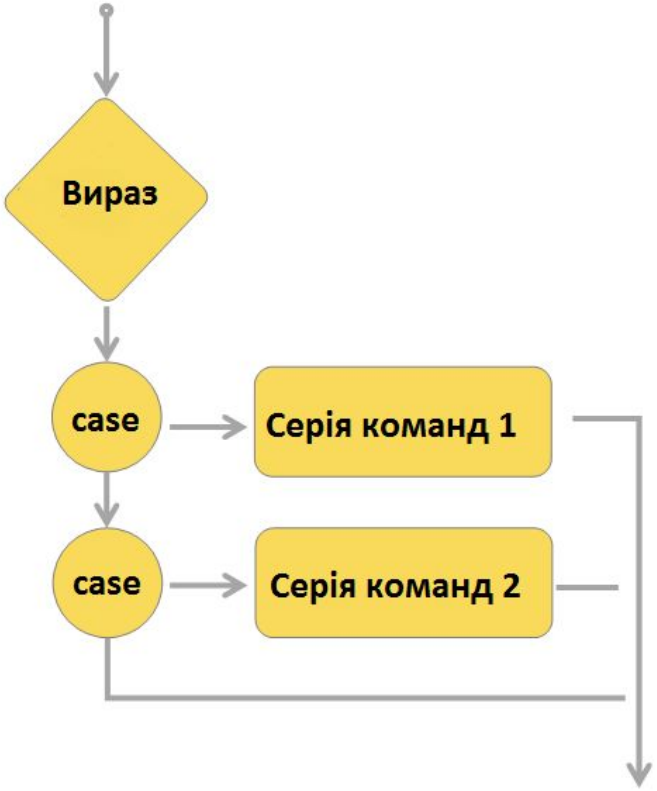
- Тернарний оператор - операція, яка повертає свій другий або третій операнд в залежності від значення логічного виразу, заданого першим операндом


```
var a = 10;
var msg = (a == 10) ? "a = 10" : "a != 10 ";
alert(msg);
```

switch-case

Конструкція switch служить для порівняння значення на рівність різними варіантами.

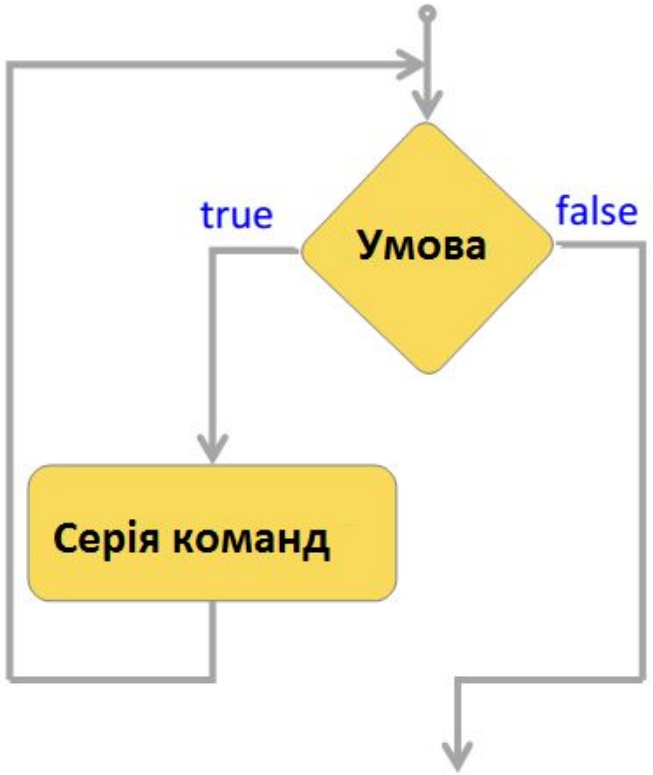
```
var day = "10";
switch (day) {
  case "10":
   {
    alert("a = 10");
   }
   break;
  case "11":
   {
     alert("a = 11");
   }
   break;
};
```



Цикл з передумовою

**Цикл, з передумовою while - це цикл, який виконується до тих
поки умова задовольняє істинності.**

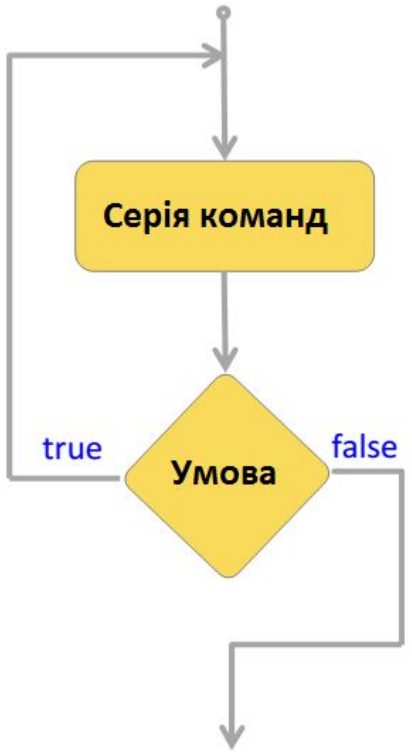
```
var counter = 0;

while (counter < 10) {
  counter++;
  document.write(counter + "<br />");
```



Цикл з постумовою

Цикл з постумовою do-while - це цикл, в якому умова перевіряється після виконання тіла циклу. Тіло циклу do-while виконується хоча б один раз.

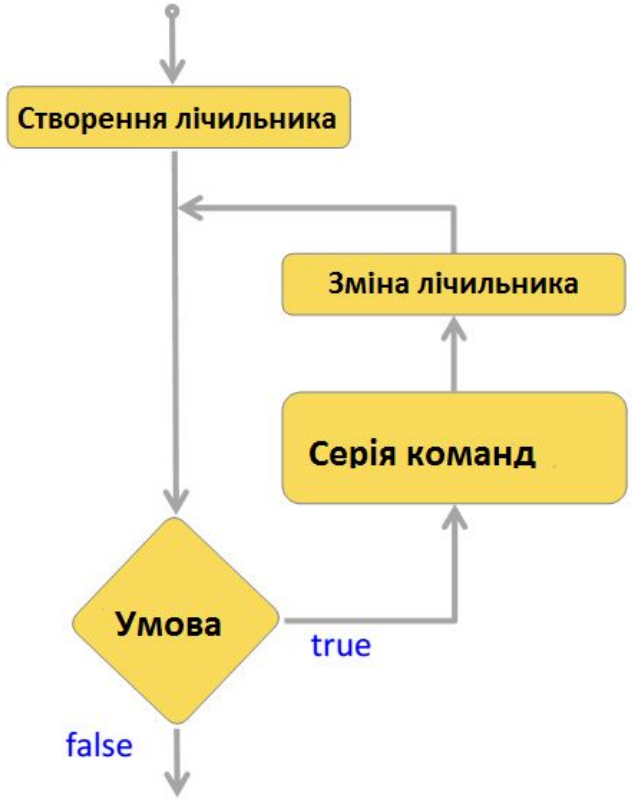
```
var counter = 0;
do {
  counter++;
  document.write(counter + "<br />");
}
while (counter < 10);
```



Цикл з лічильником

Цикл з лічильником for - це цикл, в якому змінна - лічильник ітерацій циклу, з певним кроком, змінює своє значення до заданого кінцевого значення.

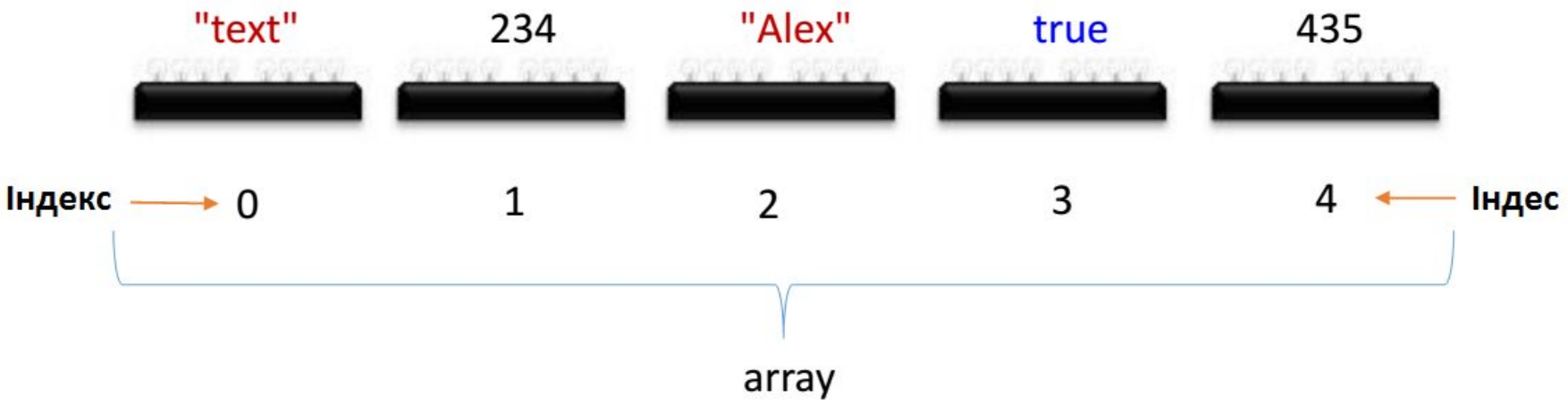
```
for (var i = 0; i < 10; i++) {
   document.write(i + "<br />");
}
```



Array

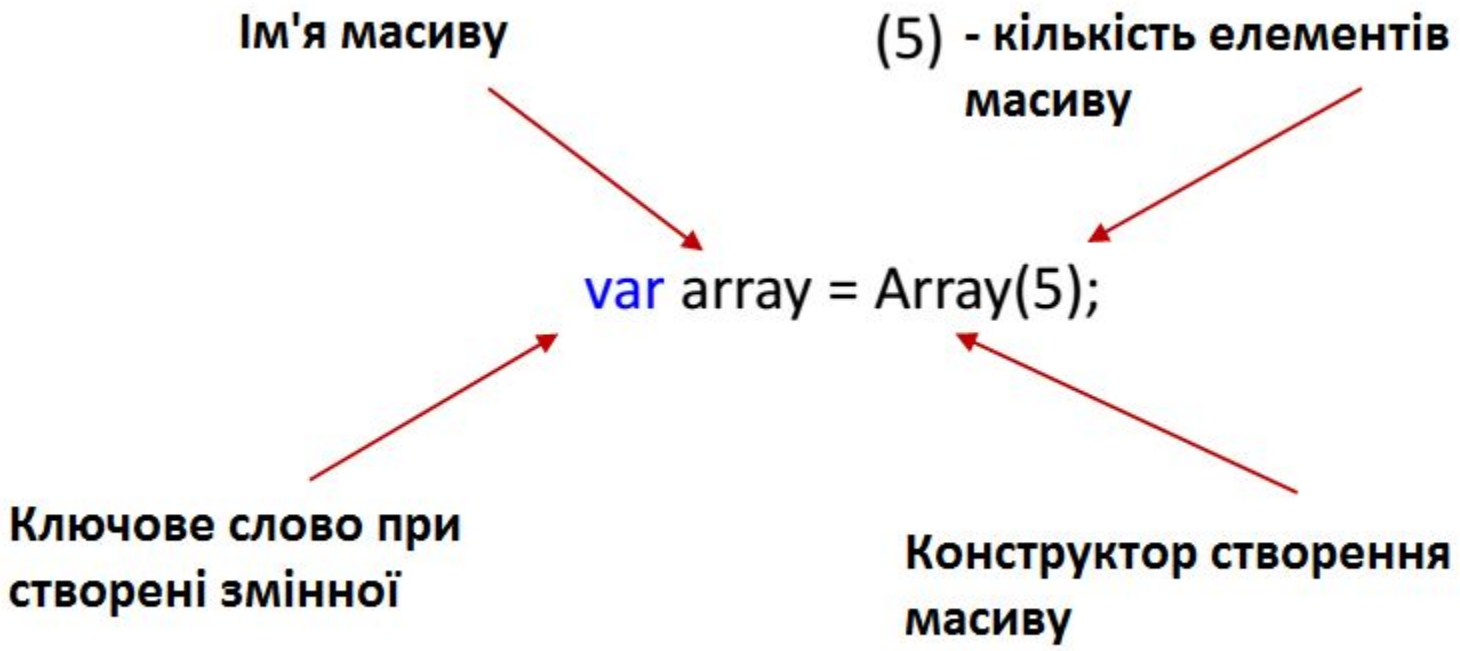
Масив в JavaScript - іменований набір не строго типізованих змінних, розташованих в пам'яті безпосередньо один за одним, доступ до яких здійснюється за індексом.

```
var array = ["text", 234, "Alex", true, 435];
```



Одновимірний

- **Одновимірний масив - масив, що містить один індекс.**



Способи створення одновимірних масивів

- `var array = Array(5);`
- `var array = Array(1, 2, 3, 4, 5, 6);`
- `var array = [1, 2, 3, 4, 6];`

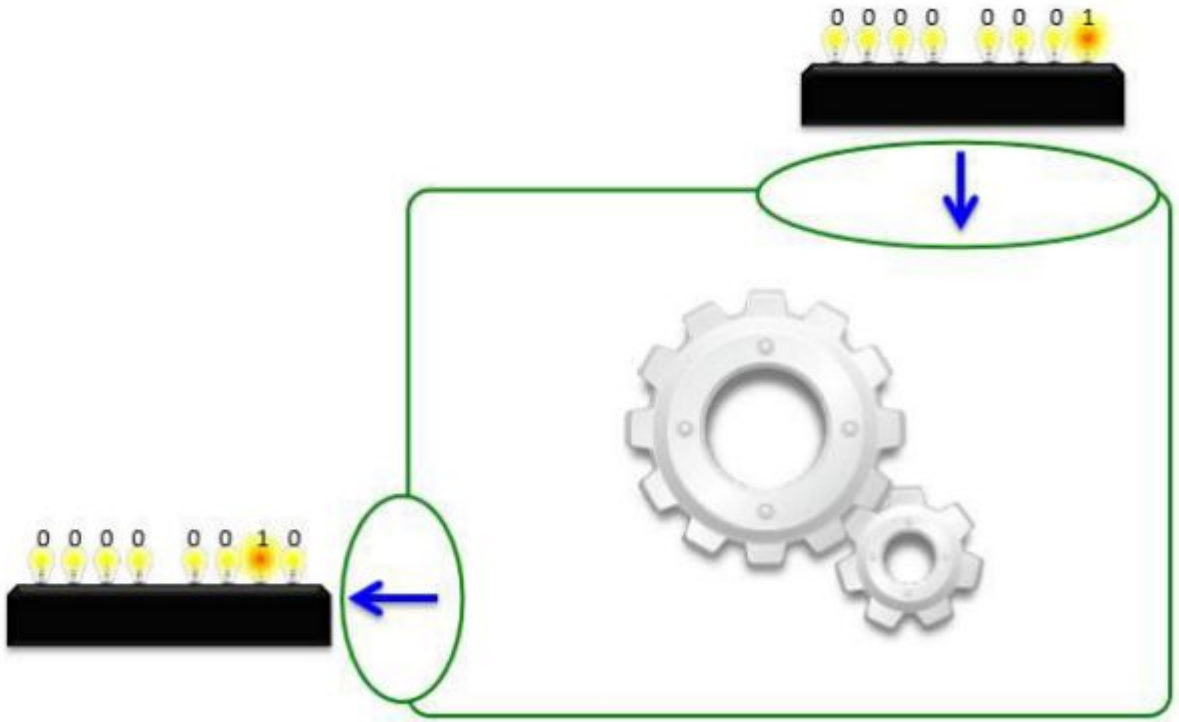
Багатовимірні масиви

- var array = new Array(5);

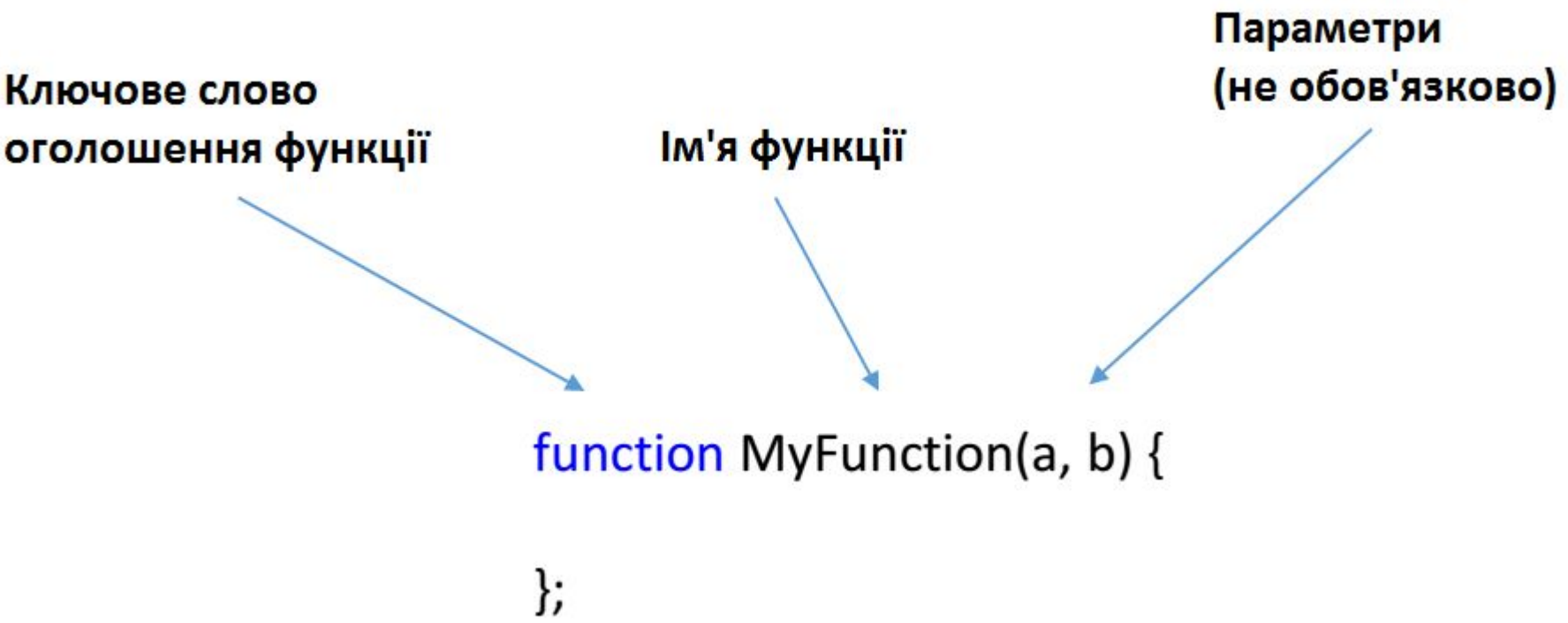
- array[0] = new Array(5);
- array[1] = new Array(5);
- array[2] = new Array(5);
- array[3] = new Array(5);
- array[4] = new Array(5);

Функція

Функція - це блок коду, який можна багаторазово виконувати.
Функція може приймати аргументи і повертати значення.



Створення функції



Значення, що повертаються функцією

Ключове слово return – завершує виконання функції і повертає деяке значення (за замовчуванням - undefined).

```
function pow (x) {
    return x * x;
};
```

Вкладені функції

- В JavaScript допускається вкладені визначення функцій в інші функції.

```
function hypotenuse(a, b) {
    function pow(x) {
        return x * x;
    };
   return Math.sqrt(pow(a) + pow(b));
};
```

Функціональні літерали

- JavaScript дозволяє визначати функції у вигляді функціональних літералів.

```
var f = function(a, b) {
   document.write(a + b);
};

f(4, 6);
```

Глобальні і локальні змінні

- Змінні, створені в функції, є локальними і не доступні за межами тіла функції.

```
var f = function () {
   var local_a = 4; // локальна
       local_b = 5;   // глобальна
};
```

Аргументи функції

- Функції в JavaScript можуть викликатися з довільним числом аргументів.

```
var f = function (a, b) {
    document.writeln(a);
    document.writeln(b);
};

f(12);
f(12, 54, 32);
```

Об’єкт

- **Об’єкти – складений тип даних, вони об’єднують множину значень в єдиний модуль і дозволяють зберігати і видобувати їх значення за іменами.**

Властивості – характеристики

Ім’я: Хвостик.
Порода: Двіртер’єр.
Колір: В плямочку.
Вік: 3 роки.
Вага: 5.5 кг.

Методи - поведінка

Сидіти

Лежати

Їсти

Приклад створення об’єкту в JavaScript

```
var dog = {

   name:  "Tail",
   breed: "Yardterrier",
   color: "in blot",
    age: 3,
   weight: 5.5,
   seat: function () {},
   lie: function () {},
   eat: function () {}
  }
cat.seat();
```

Способи створення обєктів

- **Функції які знаходяться в об'єкті, називається методами.**

Створення об’єкту через блок ініціалізації.

```
var dog = {

name  :"Tail",
breed :"Yardterrier",

run: function () { }
};
```

Створення об’єкту через конструктор.

```
var dog = new Object()

dog.name  = "Tail";
dog.breed = "Yardterrier";

dog.run = function () {};
```

Java Script Object Notation (JSON)

- **Для доступу до значень властивостей об'єкта використовується оператор '.'**

Ключове слово this

- В JavaScript ключове слово this завжди відноситься до «власника» виконуваної функції.

```
var Person1 = {
name: “Joe“
};

var Person2 = {
name: “Jim“
};

var say = function() {
document.write(“Hello, I am " + this.name);      };

Person1.sayHi = say;
Person2.sayHi = say;

Person1.sayHi();

Person2.sayHi();
```