Расширенный JavaScript

- Определение функций
- Замыкание и области действия переменных
- Оформление функций JavaScript в виде модулей
- Создание прототипов
- Создание взаимодействий вида drag-and-drop на JavaScript
- Создание таймеров и интервалов JavaScript для анимации HTML
- Использование холстов HTML5 для рисования на страницах
- Практики: Создание холста, интервалов, drag-and-drop и реализация жестов мыши

Область видимости

```
var scope = "global";              // Глобальная переменная
function checkscope( ) {
    var scope = "local";           // Локальная переменная
                                   // с тем же именем
    document.write(scope);         // Используется локальная
}
checkscope( );                     // Выводит "local"
```

```
scope = "global";                  // Глобальная переменная
function checkscope( ) {
    scope = "local";               // Изменили глобальную!
    console.log(scope);            // Глобальная
    myscope = "local";             // Неявное объявление
                                   // глобальной переменной
    console.log(myscope);          // Глобальная
}
checkscope( );                     // Выводит "locallocal"
console.log(scope);                // Выводит "local"
console.log(myscope);              // Выводит "local"
```

Область видимости

Объявленная с помощью var переменная действует во всей функции

```
function test(o) {
    var i = 0;              // i действует во всей функции
    if (typeof o == "object") {
        var j = 0;          // j действует не только в блоке
        for(var k = 0; k < 10; k++) {
            console.log(k);
        }
        console.log(k);     // k определена, выводит 10
    }
    console.log(j);         // j определена, возможно undefined
}
```

Блочная видимость (ES6)

- var - действует во всей функции

```
var apples = 5;

if (true) {
  var apples = 10; // повторное объявление
  alert(apples);   // 10 (внутри блока)
}

alert(apples);     // 10 (снаружи блока то же самое)
```

- let - действует в пределах блока

```
let apples = 5;

if (true) {
  let apples = 10;// новая переменная
  alert(apples);  // 10 (внутри блока)
}

alert(apples);    // 5 (снаружи блока значение не менялось)
```

Блочная видимость (ES6)

- var – видна во всей функции

```
alert(a); // undefined

var a = 5;
```

- let - видна только после объявления

```
alert(a); // ошибка, нет такой переменной

let a = 5;
```

- Повторное объявление не разрешается

```
let x;
let x; // ошибка: переменная x уже объявлена
```

Strict mode (ES5)

```
"use strict";
var arguments = 3.14;
```

```
function myFunction() {
    "use strict";
    y = 3.14;
}
```

Цепочка видимости (scope chain)

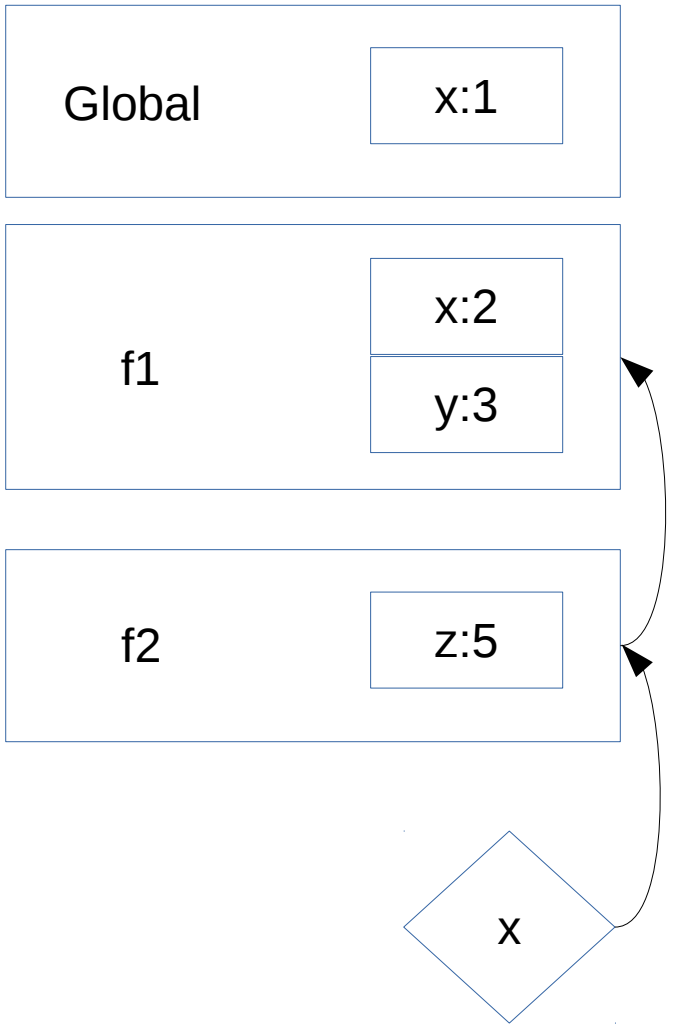
```
var x=1;

function f1(){

   var x=2,y=3;

   function f2(){
      var z=5;
      alert(x);
   }

}
```



Константы (ES6)

Аналогично let, но нельзя менять значение

```
const apple = 5;
apple = 10; // ошибка
```

```
const user = {
  name: "Вася"
};

user.name = "Петя"; // допустимо
user = 5;           // нельзя, будет ошибка
```

Вопрос

- Что будет выведено?

```
var scope = "global";
function f( ) {
    alert(scope);
    var scope = "local";
    alert(scope);
}
f( );
```

- Что будет, если var заменить на let?
- Что будет, если var заменить на const?

Расширенный JavaScript

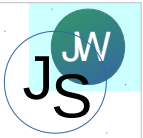
- Определение функций
- Замыкание и области действия переменных
- Оформление функций JavaScript в виде модулей
- Создание прототипов
- Создание взаимодействий вида drag-and-drop на JavaScript
- Создание таймеров и интервалов JavaScript для анимации HTML
- Использование холстов HTML5 для рисования на страницах
- Практики: Создание холста, интервалов, drag-and-drop и реализация жестов мыши

Область видимости

```
var scope = "global";           // Глобальная переменная
function checkscope( ) {
    var scope = "local";        // Локальная переменная
                                // с тем же именем
    document.write(scope);      // Используется локальная
}
checkscope( );                  // Выводит "local"
```

```
scope = "global";                 // Глобальная переменная
function checkscope( ) {
    scope = "local";              // Изменили глобальную!
    console.log(scope);           // Глобальная
    myscope = "local";            // Неявное объявление
                                  // глобальной переменной
    console.log(myscope);         // Глобальная
}
checkscope( );                    // Выводит "locallocal"
console.log(scope);               // Выводит "local"
console.log(myscope);             // Выводит "local"
```

Область видимости



Объявленная с помощью var переменная действует во всей функции

```
function test(o) {
    var i = 0;            // i действует во всей функции
    if (typeof o == "object") {
        var j = 0;        // j действует не только в блоке
        for(var k = 0; k < 10; k++) {
            console.log(k);
        }
        console.log(k);   // k определена, выводит 10
    }
    console.log(j);       // j определена, возможно undefined
}
```

Блочная видимость (ES6)

- var - действует во всей функции

```
var apples = 5;

if (true) {
  var apples = 10; // повторное объявление
  alert(apples);   // 10 (внутри блока)
}

alert(apples);     // 10 (снаружи блока то же самое)
```

- let - действует в пределах блока

```
let apples = 5;

if (true) {
  let apples = 10;// новая переменная
  alert(apples);  // 10 (внутри блока)
}

alert(apples);    // 5 (снаружи блока значение не менялось)
```

Блочная видимость (ES6)

- var – видна во всей функции

```
alert(a); // undefined

var a = 5;
```

- let - видна только после объявления

```
alert(a); // ошибка, нет такой переменной

let a = 5;
```

- Повторное объявление не разрешается

```
let x;
let x; // ошибка: переменная x уже объявлена
```

Strict mode (ES5)


```
"use strict";
var arguments = 3.14;
```

```
function myFunction() {
    "use strict";
    y = 3.14;
}
```

Цепочка видимости (scope chain)

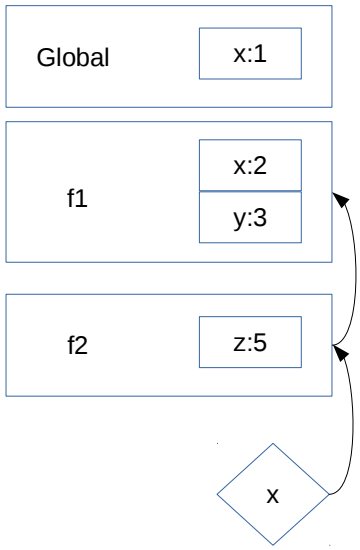
```
var x=1;

function f1(){

   var x=2,y=3;

   function f2(){
      var z=5;
      alert(x);
   }

}
```



Константы (ES6)

Аналогично let, но нельзя менять значение

```
const apple = 5;
apple = 10; // ошибка
```

```
const user = {
  name: "Вася"
};

user.name = "Петя"; // допустимо
user = 5;           // нельзя, будет ошибка
```

Вопрос

- Что будет выведено?

```
var scope = "global";
function f( ) {
    alert(scope);
    var scope = "local";
    alert(scope);
}
f( );
```

- Что будет, если var заменить на let?
- Что будет, если var заменить на const?