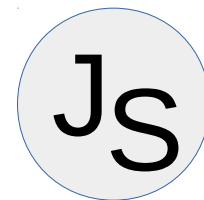
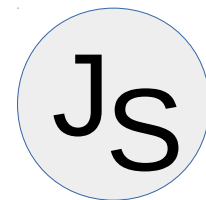


# Инструкции



- Инструкция-выражение
- Блок
- Условная инструкция
- Инструкция switch
- Инструкция while
- Инструкция do/while
- Инструкция for
- Инструкция for/in
- Инструкция break
- Инструкция continue
- Инструкции var и let

# Инструкция-выражение. Блок.



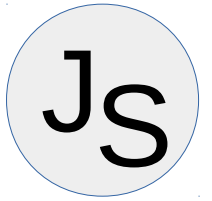
## Выражение

```
5;  
s = "строка";  
counter++;  
delete point.z;  
alert("Hello, world");  
window.close();
```

## Блок

```
{  
    x = Math.PI;  
    cx = Math.cos(x);  
    alert("cos(" + x + ") = " + cx);  
}
```

# Условная инструкция



```
if (выражение)  
    инструкция1
```

```
if (выражение)  
    инструкция1  
else  
    инструкция2
```

```
if (username != null) {  
    alert("Hello " + username);  
}  
else {  
    username = prompt("What is your name?");  
    alert("Hello " + username);  
}
```

# Условная инструкция



```
if (i == j)
    if (j == k) document.write("i equals k");
else
    document.write("i doesn't equal j");
```

```
if (i == j) {
    if (j == k) {
        document.write("i equals k");
    }
    else {
        document.write("i doesn't equal j");
    }
}
```

# Инструкция switch



```
switch (выражение) {  
  case выражение1:  
    инструкции1  
  case выражение2:  
    инструкции2  
  default:  
    инструкции  
}
```

- Выражения любого типа
- Проверка на идентичность

```
switch(p) {  
  case 0:  
  case 1:  
    break;  
  case 2:  
    n = n * n;  
    break;  
  default:  
    n = 2 * n;  
}
```

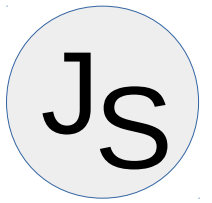
# Инструкция switch



```
case 60*60*24:
case Math.cos(1):
case names[0]:
```

```
function convert(x) {
  switch(typeof x) {
    case 'number':
      return x.toString(16);
    case 'string':
      return '"' + x + '"';
    case 'boolean':
      return x.toString().toUpperCase();
    default:
      return x.toString();
  }
}
```

# Инструкции while и do while



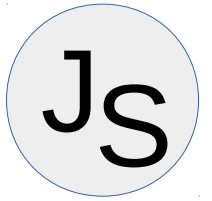
```
while (выражение)  
    инструкция
```

```
do  
    инструкция  
while (выражение);
```

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

```
var i = 0;  
do {  
    document.write(a[i] + "<br>");  
} while (++i < a.length);
```

# Инструкция for



```
for (инициализация; проверка; инкремент)  
    инструкция
```

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

```
for(i = 0, j = 10 ; i < 10 ; i++, j--){  
    sum += i * j;  
}
```



# Инструкция for in

JS

```
for (переменная in объект)  
    инструкция
```

```
for (var prop in my_object) {  
    document.write(  
        "name: " + prop +  
        " value: " + my_object[prop] +  
        "<br>"  
    );  
}
```

```
var o = {x:1, y:2, z:3};  
var a = new Array( );  
var i = 0;  
for(a[i++] in o) ; // пустое тело цикла  
for(i in a) alert(i);
```

# Инструкция break



```
break;  
break метка;
```

```
for(i = 0; i < a.length; i++) {  
    if (a[i] % 2 == 0) {  
        found=true;  
        break;  
    }  
}
```

# Инструкция break



```
outerloop:
  for(var i = 0; i < 10; i++) {
    innerloop:
      for(var j = 0; j < 10; j++) {
        if (j > 3) break;
        if (i == 2) break innerloop;
        if (i == 4) break outerloop;
        document.write(
          "i = " + i + " j = " + j + "<br>"
        );
      }
    }
  }
document.write(
  "FINAL i = " + i + " j = " + j + "<br>"
);
```

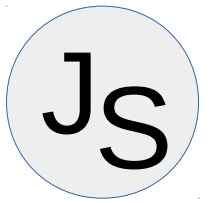
# Инструкция continue



```
continue;  
continue метка;
```

```
for(i = 0; i < data.length; i++) {  
    if (data[i] == null){  
        continue;  
    }  
    total += data[i];  
}
```

# Инструкции var и let



```
var name_1[=value_1] [ , ..., name_n[=value_n] ];  
let name_1[=value_1] [ , ..., name_n[=value_n] ];
```

```
var i;  
var j = 0;  
var p, q;  
var greeting = "hello" + name;  
var x = 2.34, y = Math.cos(0.75), r, theta;
```

```
for(let i = 0, j=10; i < 10; i++,j--){  
    document.write(i*j, "<br>");  
}  
for(let i in o){  
    document.write(i, "<br>");  
}
```

# Пустая инструкция



```
;
```

## Инициализация массива

```
for(i=0; i < a.length; a[i++] = 0) ;
```

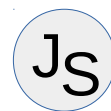
```
if ( a == 0 || b == 0 );  
    o = null;           // выполнится всегда
```

## Инструкции



- Инструкция-выражение
- Блок
- Условная инструкция
- Инструкция switch
- Инструкция while
- Инструкция do/while
- Инструкция for
- Инструкция for/in
- Инструкция break
- Инструкция continue
- Инструкции var и let

## Инструкция-выражение. Блок.



### Выражение

```
5;  
s = "строка";  
counter++;  
delete point.z;  
alert("Hello, world");  
window.close();
```

### Блок

```
{  
    x = Math.PI;  
    cx = Math.cos(x);  
    alert("cos(" + x + ") = " + cx);  
}
```



## Условная инструкция



```
if (выражение)  
    инструкция1
```

```
if (выражение)  
    инструкция1  
else  
    инструкция2
```

```
if (username != null){  
    alert("Hello " + username);  
}  
else {  
    username = prompt("What is your name?");  
    alert("Hello " + username);  
}
```

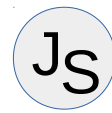
## Условная инструкция



```
if (i == j)
  if (j == k) document.write("i equals k");
else
  document.write("i doesn't equal j");
```

```
if (i == j) {
  if (j == k) {
    document.write("i equals k");
  }
  else {
    document.write("i doesn't equal j");
  }
}
```

## Инструкция switch



```
switch (выражение) {  
  case выражение1:  
    инструкции1  
  case выражение2:  
    инструкции2  
  default:  
    инструкции  
}
```

- Выражения любого типа
- Проверка на идентичность

```
switch(p) {  
  case 0:  
  case 1:  
    break;  
  case 2:  
    n = n * n;  
    break;  
  default:  
    n = 2 * n;  
}
```

## Инструкция switch



```
case 60*60*24:  
case Math.cos(1):  
case names[0]:
```

```
function convert(x) {  
  switch(typeof x) {  
    case 'number':  
      return x.toString(16);  
    case 'string':  
      return '"' + x + '"';  
    case 'boolean':  
      return x.toString().toUpperCase();  
    default:  
      return x.toString();  
  }  
}
```

# Инструкции while и do while



```
while (выражение)  
    инструкция
```

```
do  
    инструкция  
while (выражение);
```

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

```
var i = 0;  
do {  
    document.write(a[i] + "<br>");  
} while (++i < a.length);
```

## Инструкция for

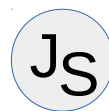


```
for (инициализация; проверка; инкремент)  
    инструкция
```

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

```
for(i = 0, j = 10 ; i < 10 ; i++, j--){  
    sum += i * j;  
}
```

## Инструкция for in

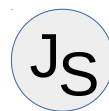


```
for(переменная in объект)  
    инструкция
```

```
for (var prop in my_object) {  
    document.write(  
        "name: " + prop +  
        " value: " + my_object[prop] +  
        "<br>"  
    );  
}
```

```
var o = {x:1, y:2, z:3};  
var a = new Array( );  
var i = 0;  
for(a[i++] in o) ;           // пустое тело цикла  
for(i in a) alert(i);
```

## Инструкция break



```
break;  
break метка;
```

```
for(i = 0; i < a.length; i++) {  
    if (a[i] % 2 ==0){  
        found=true;  
        break;  
    }  
}
```



## Инструкция break



```
outerloop:
  for(var i = 0; i < 10; i++) {
    innerloop:
      for(var j = 0; j < 10; j++) {
        if (j > 3) break;
        if (i == 2) break innerloop;
        if (i == 4) break outerloop;
        document.write(
          "i = " + i + " j = " + j + "<br>"
        );
      }
    }
  document.write(
    "FINAL i = " + i + " j = " + j + "<br>"
  );
```

# Инструкция continue



```
continue;  
continue метка;
```

```
for(i = 0; i < data.length; i++) {  
    if (data[i] == null){  
        continue;  
    }  
    total += data[i];  
}
```

## Инструкции var и let



```
var name_1[=value_1] [ ,..., name_n[=value_n]];
let name_1[=value_1] [ ,..., name_n[=value_n]];
```

```
var i;
var j = 0;
var p, q;
var greeting = "hello" + name;
var x = 2.34, y = Math.cos(0.75), r, theta;
```

```
for(let i = 0, j=10; i < 10; i++,j--){
    document.write(i*j, "<br>");
}
for(let i in o){
    document.write(i, "<br>");
}
```

## Пустая инструкция



```
;
```

### Инициализация массива

```
for(i=0; i < a.length; a[i++] = 0) ;
```

```
if ( a == 0 || b == 0 );  
    o = null;                // выполнится всегда
```