

Лекция
JavaScript Циклы

Циклы

Циклы могут выполнять блоки кода определенное количество раз. Они удобны в случаях, когда вы хотите запустить один и тот же код многократно, добавляя разное значение каждый раз.

В JavaScript есть три типа циклов: `for`, `while`, и `do while`.

Чаще всего используется цикл `for`.

ЦИКЛЫ

```
for (statement 1; statement 2;  
statement 3) {  
code block to be executed  
}
```

Statement 1(выражение 1) выполняется до запуска цикла (code block).

Statement 2(выражение 2) определяет условие для запуска цикла (code block).

Statement 3(выражение 3) выполняется каждый раз после выполнения цикла (code block).

Классический цикл for имеет три компонента, или выражения.

Цикл For

```
for (i=1; i<=5; i++) {  
  document.write(i + "<br />");  
}
```

В этом примере, Выражение 1 устанавливает переменную до запуска цикла (`var i = 1`).

Выражение 2 определяет условие для запуска цикла (`i` должно быть меньше либо равно 5).

Выражение 3 увеличивает значение (`i++`) после каждого выполнения блока кода в цикле.

Цикл While

Цикл `while` повторяет блок кода, пока определенное условие является `true`(ИСТИННЫМ).

```
while (condition) {  
code block  
}
```

Condition(Условием) может быть любое условное выражение, которое возвращает `true`(истина) или `false`(ложь).

Цикл Do...While

Цикл do...while является вариацией цикла while. Цикл выполнит блок кода один раз, до проверки правильности условия, а затем цикл будет повторяться, пока условие будет истинным.

```
do {  
code block  
}  
while (condition
```

Пример:

```
var i=20;  
do {  
document.write(i + "<br />");  
i++;  
}  
while (i<=25);
```

Цикл всегда будет выполняться как минимум один раз, даже если условие ложно, потому что блок кода выполняется до проверки условия.

Break

Выражение `break` "выпрыгивает" из цикла и продолжает выполнение с кода после цикла.

```
for (i = 0; i <= 10; i++) {  
  if (i == 5) {  
    break;  
  }  
  document.write(i + "<br />");  
}
```

Как только `i` будет равно 5, произойдет **ВЫХОД ИЗ ЦИКЛА**.

Вы можете использовать ключевое слово `return` для немедленного возврата некоторого значения из цикла внутри функции. Это также нарушит цикл.

Continue

Выражение `continue` останавливает только одну итерацию в цикле и продолжает со следующей итерации.

```
for (i = 0; i <= 10; i++)  
{  
  if (i == 5) {  
    continue;  
  }  
  document.write(i +  
    "<br />");  
}
```

Если использовать этот код, то значение 5 не будет напечатано, потому что выражение `continue` пропускает эту итерацию цикла.



1/3

 ОТВЕТ

Из какого количества
компонентов состоит
классический цикл "for"?

 3 2 4 1

Вызов завершен

ПРОВЕРИТЬ



1/3

 ОТВЕТ

Результатом условного выражения всегда является:

- Строковое значение
- Числовое значени
- Булево значение (true или false)

ПРОВЕРИТЬ



1/1

 ПОМОЩЬ ОТВЕТ

Примените ключевые слова "do" и "while" в их соответствующих позициях.

```
var count=1;
__ {
  document.write("hello <br />");
  count++;
}
_____ (count<=10);
```



1/2

 ОТВЕТ

Выражение "break":

- Заканчивает выполнение цикла
- Игнорирует текущую итерацию и переходит к следующей
- Останавливает целый скрипт

КОММЕНТОВ: 32

ПРОВЕРИТЬ



2/2

 ПОМОЩЬ ОТВЕТ

Что выведет следующий код?

```
var sum=0;
for(i=4; i<8; i++) {
  if (i == 6) {
    continue;
  }
  sum += i;
}
document.write(sum);
```

[ПРОВЕРИТЬ](#)