WHILE

Цикл while имеет вид

while (условие) {

// код, тело цикла

}

Пока условие верно – выполняется код из тела цикла.

Например, цикл ниже выводит i пока i < 3:

var i = 0;

while (i < 3) {

alert( i );

i++;

}

Есле би i++ в коде више не било то цикл виполнялся би в теорій вечно

**Бесконечный цикл** можно сделать и проще:

while (true) {

// ...

}

**Условие в скобках интерпретируется как логическое значение, поэтому вместо while (i!=0) обычно пишут while (i)**:

var i = 3;

while (i) { // при i, равном 0, значение в скобках будет false и цикл остановится

alert( i );

i--;

}

## [Цикл do…while](http://learn.javascript.ru/while-for" \l "%D1%86%D0%B8%D0%BA%D0%BB-do-while)

## [Цикл do…while](http://learn.javascript.ru/while-for#%D1%86%D0%B8%D0%BA%D0%BB-do-while)

Проверку условия можно поставить под телом цикла, используя специальный синтаксис do..while:

do {

// тело цикла

} while (условие);

Цикл опсаний таким образом сначала виполняетца тело а затем проверяет условие

var i = 0;

do {

alert( i );

i++;

} while (i < 3);

## [Цикл for](http://learn.javascript.ru/while-for" \l "%D1%86%D0%B8%D0%BA%D0%BB-for)

Чаще всего применяется цикл for. Выглядит он так:

for (начало; условие; шаг) {

// ... тело цикла ...

}

Пример цикла коьорый виполняет lert(i) для i от 0 до 2 включительно (до 3):

var i;

for (i = 0; i < 3; i++) {

alert( i );

}

## [Пропуск частей for](http://learn.javascript.ru/while-for" \l "%D0%BF%D1%80%D0%BE%D0%BF%D1%83%D1%81%D0%BA-%D1%87%D0%B0%D1%81%D1%82%D0%B5%D0%B9-for)

Любая часть for может ьить пропущено

Например, можно убрать начало. Цикл в примере ниже полностью идентичен приведённому выше:

var i = 0;

for (; i < 3; i++) {

alert( i ); // 0, 1, 2

}

Модна убрать и шаг

var i = 0;

for (; i < 3;) {

alert( i );

// цикл превратился в аналог while (i<3)

}

А можно и вообще убрать всё, получив бесконечный цикл:

for (;;) {

// будет выполняться вечно

}

## [Прерывание цикла: break](http://learn.javascript.ru/while-for" \l "%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D1%86%D0%B8%D0%BA%D0%BB%D0%B0-break)

Выйти из цикла можно не только при проверке условия но и, вообще, в любой момент. Эту возможность обеспечивает директива break.

var sum = 0;

while (true) {

var value = +prompt("Введите число", '');

if (!value) break; // (\*)

sum += value;

}

alert( 'Сумма: ' + sum );

## [Следующая итерация: continue](http://learn.javascript.ru/while-for" \l "continue)

Директива continue прекращает выполнение текущей итерации цикла.

Она – в некотором роде «младшая сестра» директивы break

for (var i = 0; i < 10; i++) {

if (i % 2 == 0) continue;

alert(i);

}

## [Метки для break/continue](http://learn.javascript.ru/while-for" \l "%D0%BC%D0%B5%D1%82%D0%BA%D0%B8-%D0%B4%D0%BB%D1%8F-break-continue)

Бывает нужно выйти одновременно из нескольких уровней цикла.

Например, внутри цикла по i находится цикл по j, и при выполнении некоторого условия мы бы хотели выйти из обоих циклов сразу

outer: for (var i = 0; i < 3; i++) {

for (var j = 0; j < 3; j++) {

var input = prompt('Значение в координатах '+i+','+j, '');

// если отмена ввода или пустая строка -

// завершить оба цикла

if (!input) break outer; // (\*)

}

}

alert('Готово!');