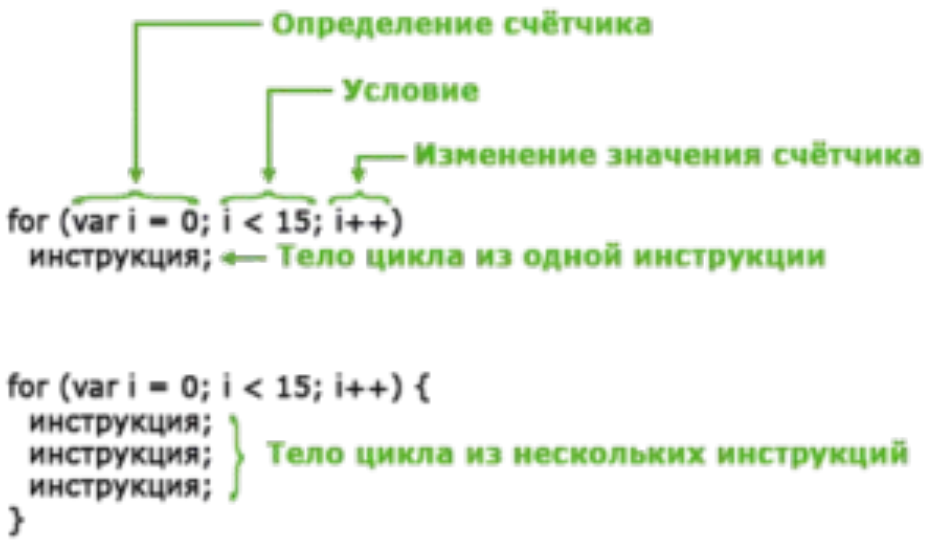
Циклы

Цикл это рамки, код внутри которых выполняется сверху вниз и повторяется с начала, когда достигает конца. Продолжается это дело до тех пор, пока выполняется какое то условие.

Конструкция **for** используется для повторения блока операторов, заключенных в фигурные скобки. Счетчик приращений обычно используется для приращения и завершения цикла. Оператор **for** подходит для любых повторяющихся действий и часто используется в сочетании с массивами коллекций данных/выводов.

Цикл for обычно содержит переменную, которая изменяется на протяжении работы цикла, мы можем пользоваться её меняющимся значением в своих целях. Переменная является **локальной для цикла**, если она создаётся при инициализации.



- **Инициализация** – здесь обычно присваивают начальное значение переменной цикла. Например: int i = 0;
- **Условие** – здесь задаётся условие, при котором выполняется цикл. Как только условие нарушается, цикл завершает работу. Например: i < 100;
- **Изменение** – здесь указывается изменение переменной цикла на каждой итерации. Например: i++;

```
for (int i = 0; i < 100; i++)
{
// тело цикла
}
```

В цикле for можно сделать несколько счётчиков, несколько условий и несколько инкрементов, разделяя их при помощи оператора запятая, смотрите пример:

```
// объявить i и j
// прибавлять i+1 и j+2
for (byte i = 0, j = 0; i < 10; i++, j += 2)
{
// тут i меняется от 0 до 9
// и j меняется от 0 до 18
}
```

Также в цикле **может вообще не быть настроек**, и такой цикл можно считать вечным, замкнутым:

```
for (;;) {
// выполняется вечно...
}
```

Использование замкнутых циклов не очень приветствуется, но иногда является очень удобным способом поймать какое-то значение, или дать программе “зависнуть” при наступлении ошибки. Но, как мы знаем, нет ничего вечного, поэтому из цикла таки можно выйти при помощи оператора break

.

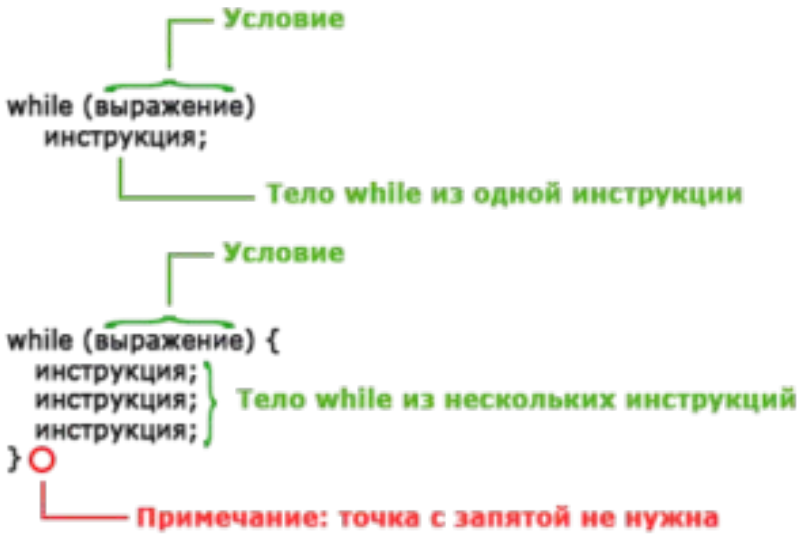
Оператор break
(англ. “ломать”) позволяет досрочно покинуть цикл, использовать его можно как по условию, так и как-угодно-удобно. Например давайте досрочно выйдем из цикла при достижении какого-то значения:

```
for (;;) {
if (кнопка нажата) break;
}
```

Оператор WHILE используется в C++ и Ардуино для организации повтора одних и тех команд произвольное количества раз.

Цикл while

(англ. “пока”), он же называется цикл с предусловием, выполняется до тех пор, пока верно указанное условие. Если условие сразу неверно, цикл даже не начнёт свою работу и будет полностью пропущен. Объявляется очень просто: ключевое слово while, далее **условие** в скобках, и тело цикла



```
int i = 0;
while (i < 10) {
i++;
}
```

Цикл while
тоже удобно использовать как вечный цикл, например, ожидая наступление какого-либо события (нажатие кнопки):

```
// выполняется, пока не нажата кнопка
while (кнопка не нажата);
```

Цикл do while

- do while
- – “делать пока”, работа этого цикла полностью аналогична циклу while
- за тем исключением, что здесь условие задаётся после цикла, т.е. **цикл выполнится один раз**, затем проверит условие, а не наоборот.

```
do {
// тело цикла
}
while (условие);
```