

Циклы со счетчиком:

```
for i = 1, 10, 1 do
    -- Здесь ваш код
end
```

Используются когда мы заранее знаем количество повторений.

i - переменная-счетчик в которой хранится текущее значение

1 - начальное значение

10 - конечное значение

1 - шаг, значение на которое будет увеличиваться счетчик

Бесконечные циклы:

```
while true do
    -- Здесь ваш код
end
```

Бесконечные циклы будут выполняться всегда, поэтому следующие за ними инструкции никогда не выполнятся.

Поэтому бесконечные циклы лучше всего помещать в функции, а функции вызывать с помощью конструкции

```
coroutine.wrap(move)(p)
```

move - название функции

p - параметр функции

Циклы со счетчиком:

```
for i = 1, 10, 1 do
    -- Здесь ваш код
end
```

Используются когда мы заранее знаем количество повторений.

i - переменная-счетчик в которой хранится текущее значение

1 - начальное значение

10 - конечное значение

1 - шаг, значение на которое будет увеличиваться счетчик

Бесконечные циклы:

```
while true do
    -- Здесь ваш код
end
```

Бесконечные циклы будут выполняться всегда, поэтому следующие за ними инструкции никогда не выполнятся.

Поэтому бесконечные циклы лучше всего помещать в функции, а функции вызывать с помощью конструкции

```
coroutine.wrap(move)(p)
```

move - название функции

p - параметр функции

Циклы со счетчиком:

```
for i = 1, 10, 1 do
    -- Здесь ваш код
end
```

Используются когда мы заранее знаем количество повторений.

i - переменная-счетчик в которой хранится текущее значение

1 - начальное значение

10 - конечное значение

1 - шаг, значение на которое будет увеличиваться счетчик

Бесконечные циклы:

```
while true do
    -- Здесь ваш код
end
```

Бесконечные циклы будут выполняться всегда, поэтому следующие за ними инструкции никогда не выполнятся.

Поэтому бесконечные циклы лучше всего помещать в функции, а функции вызывать с помощью конструкции

```
coroutine.wrap(move)(p)
```

move - название функции

p - параметр функции

Циклы со счетчиком:

```
for i = 1, 10, 1 do
    -- Здесь ваш код
end
```

Используются когда мы заранее знаем количество повторений.

i - переменная-счетчик в которой хранится текущее значение

1 - начальное значение

10 - конечное значение

1 - шаг, значение на которое будет увеличиваться счетчик

Бесконечные циклы:

```
while true do
    -- Здесь ваш код
end
```

Бесконечные циклы будут выполняться всегда, поэтому следующие за ними инструкции никогда не выполнятся.

Поэтому бесконечные циклы лучше всего помещать в функции, а функции вызывать с помощью конструкции

```
coroutine.wrap(move)(p)
```

move - название функции

p - параметр функции