Loops with a counter:

```
for i = 1, 10, 1 do
    -- Here is your code
end
```

Used when we know in advance the number of iterations.
i - the counter variable in which the current value is stored

1 - starting value
10 - ending value
1 - step, the value by which the counter will be incremented

Infinite loops:

```
while true do
    -- Here is your code
end
```

Infinite loops will always execute, so any instructions following them will never get executed.

Therefore, it's best to place infinite loops inside functions, and call those functions using a construction.

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

move - name of the function
p - parameter of the function

Loops with a counter:

for i = 1, 10, 1 do
-- Here is your code
end

Used when we know in advance the number of iterations.
i - the counter variable in which the current value is stored

1 - starting value
10 - ending value
1 - step, the value by which the counter will be incremented

Infinite loops:

while true do
-- Here is your code
end

Infinite loops will always execute, so any instructions following them will never get executed.

Therefore, it's best to place infinite loops inside functions, and call those functions using a construction.

coroutine.wrap(move)(p)

move - name of the function
p - parameter of the function

Loops with a counter:

for i = 1, 10, 1 do
-- Here is your code
end

Used when we know in advance the number of iterations.
i - the counter variable in which the current value is stored

1 - starting value
10 - ending value
1 - step, the value by which the counter will be incremented

Infinite loops:

while true do
-- Here is your code
end

Infinite loops will always execute, so any instructions following them will never get executed.

Therefore, it's best to place infinite loops inside functions, and call those functions using a construction.

coroutine.wrap(move)(p)

move - name of the function
p - parameter of the function

Loops with a counter:

for i = 1, 10, 1 do
-- Here is your code
end

Used when we know in advance the number of iterations.
i - the counter variable in which the current value is stored

1 - starting value
10 - ending value
1 - step, the value by which the counter will be incremented

Infinite loops:

while true do
-- Here is your code
end

Infinite loops will always execute, so any instructions following them will never get executed.

Therefore, it's best to place infinite loops inside functions, and call those functions using a construction.

coroutine.wrap(move)(p)

move - name of the function
p - parameter of the function