

### Pętla z licznikiem:

```
for i = 1, 10, 1 do
    -- Tutaj jest twój kod
end
```

Używane, gdy z góry wiemy, ile razy pętla ma się wykonać.

i - zmienna-licznik, w której przechowywana jest aktualna wartość

1 - wartość początkowa

10 - wartość końcowa

1 - krok, o który zwiększa się licznik

### Nieskończone pętla:

```
while true do
    -- Tutaj jest twój kod
end
```

Nieskończone pętla będą się wykonywać zawsze, więc instrukcje za nimi nigdy się nie wykonają.

Dlatego nieskończone pętla najlepiej umieścić w funkcjach, a funkcje wywołać za pomocą konstrukcji.

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

move - nazwa funkcji

p - parametr funkcji

### Pętla z licznikiem:

```
for i = 1, 10, 1 do
    -- Tutaj jest twój kod
end
```

Używane, gdy z góry wiemy, ile razy pętla ma się wykonać.

i - zmienna-licznik, w której przechowywana jest aktualna wartość

1 - wartość początkowa

10 - wartość końcowa

1 - krok, o który zwiększa się licznik

### Nieskończone pętla:

```
while true do
    -- Tutaj jest twój kod
end
```

Nieskończone pętla będą się wykonywać zawsze, więc instrukcje za nimi nigdy się nie wykonają.

Dlatego nieskończone pętla najlepiej umieścić w funkcjach, a funkcje wywołać za pomocą konstrukcji.

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

move - nazwa funkcji

p - parametr funkcji

### Pętla z licznikiem:

```
for i = 1, 10, 1 do
    -- Tutaj jest twój kod
end
```

Używane, gdy z góry wiemy, ile razy pętla ma się wykonać.

i - zmienna-licznik, w której przechowywana jest aktualna wartość

1 - wartość początkowa

10 - wartość końcowa

1 - krok, o który zwiększa się licznik

### Nieskończone pętla:

```
while true do
    -- Tutaj jest twój kod
end
```

Nieskończone pętla będą się wykonywać zawsze, więc instrukcje za nimi nigdy się nie wykonają.

Dlatego nieskończone pętla najlepiej umieścić w funkcjach, a funkcje wywołać za pomocą konstrukcji.

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

move - nazwa funkcji

p - parametr funkcji

### Pętla z licznikiem:

```
for i = 1, 10, 1 do
    -- Tutaj jest twój kod
end
```

Używane, gdy z góry wiemy, ile razy pętla ma się wykonać.

i - zmienna-licznik, w której przechowywana jest aktualna wartość

1 - wartość początkowa

10 - wartość końcowa

1 - krok, o który zwiększa się licznik

### Nieskończone pętla:

```
while true do
    -- Tutaj jest twój kod
end
```

Nieskończone pętla będą się wykonywać zawsze, więc instrukcje za nimi nigdy się nie wykonają.

Dlatego nieskończone pętla najlepiej umieścić w funkcjach, a funkcje wywołać za pomocą konstrukcji.

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

move - nazwa funkcji

p - parametr funkcji