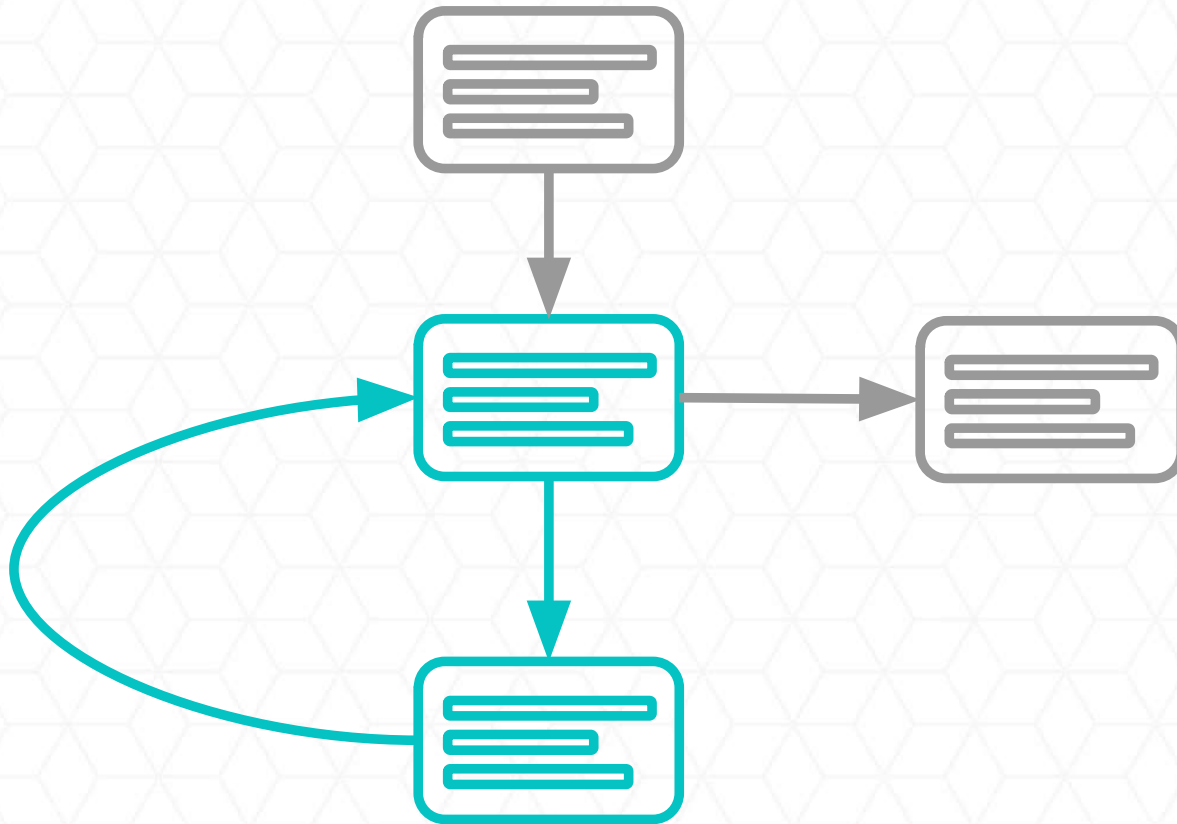


Pętle



Pętla for

Pętla **for** to podstawowe narzędzie do iterowania po kolekcjach. Pobieranie kolejnych elementów i praca z nimi.

```
for element in collection:  
    # code that might use element
```

W przypadku stringa:

```
for character in sequence:  
    # code that might use character
```

Range

Inny przykład obiektu iterowalnego to **range** - narzędzie do generowania liczb z danego zakresu. Tak jak przy slicingu możemy podać start, stop and krok.

range(stop)

`range(3)`

- <0, 1, 2>

range(start, stop)

`range(4, 8)`

- <4, 5, 6, 7>

range(start, stop, krok)

`range(0, 10, 3)`

- <0, 3, 6, 9>

Pętla while

Pętla **while** to podstawowa pętla ogólnego zastosowania.

```
while condition:
```

```
    # code
```

```
np.
```

```
while command != "stop":
```

```
    execute_code()
```

Loop control statements

break

Natychmiast przerywa (i opuszcza) pętlę.
Program przechodzi do pierwszej instrukcji poza pętlą.

continue

Natychmiast przerywa aktualną iterację pętli.
Program wraca na górę bloku pętli - **while** ponownie sprawdza warunek, **for** pobiera kolejny element.

Break

```
5 print("Repeater program v2")
6
7 while True:
8     command = input("Please provide command: ")
9     if command == "":
10         break
11     print(command)
12
13 print("Program finished")
```


Continue

```
37     print("Repeater program v3")
38
39     while True:
40         command = input("Please provide command: ")
41         if command == "skip":
42             continue
43         print(command)
44
45     print("Program finished")
46
```