PYTHON COURSE

ENTRY LEVEL

Basics of programming in Python 3.10

This course will cover part of the arguments found in PCEP™ – Certified Entry-Level Python Programmer Certification

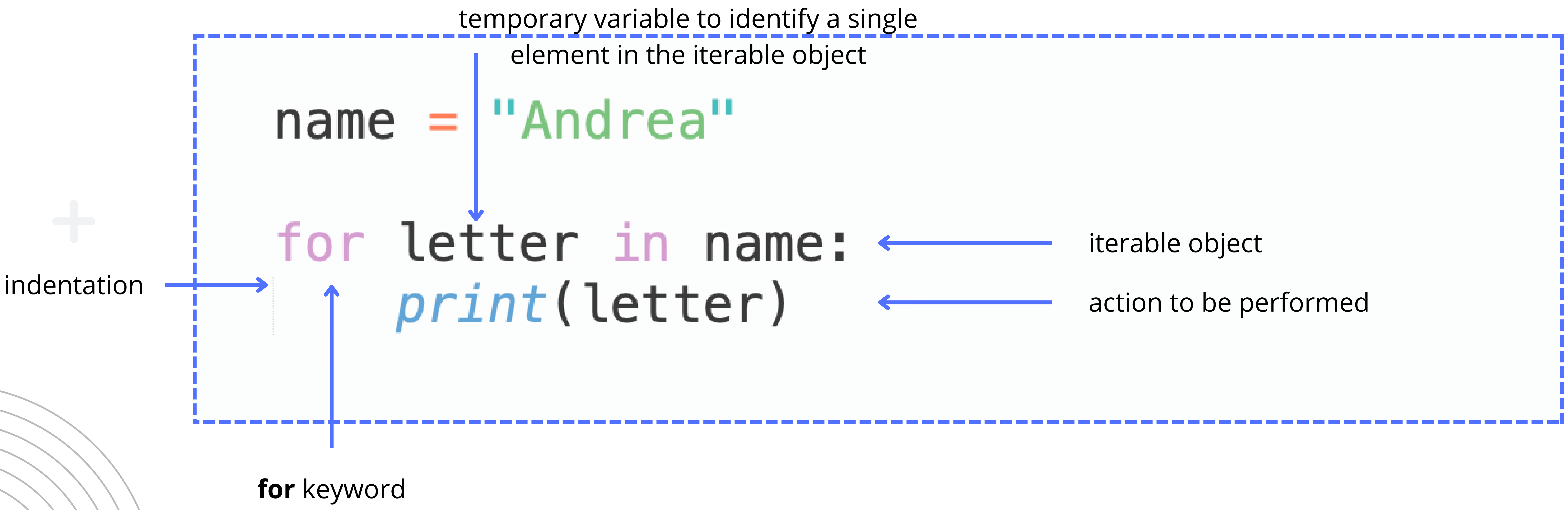
CONTROL FLOW: CONDITIONAL BLOCKS AND LOOPS

ITERATIONS

- iteration is the concept referred to repeat for a defined number of times certain actions
- In programming, iterative code blocks are called loops
- The program will so execute a defined number of times instructions contained in the iterative code block
- In Python, we can perform iterations with objects composed by a moltitude of elements, the so called Iterables
- In Python, String (considerable as multiple characters) are an example of iterable objects

LOOPS: FOR LOOP

- For loops consent to define a specific number of times we want the block to be executed and perform desired actions in loop for that many times:



```
name = "Andrea"

for letter in name:
    print(letter)
```

temporary variable to identify a single element in the iterable object
iterable object
action to be performed
indentation
for keyword

LOOPS: FOR LOOP

- How for loops works:

```python
name = "Andrea"

for letter in name:
    print(letter)
```

for every element in the iterable

perform this actions

the temporary variable letter stores a different value in each iteration

LOOPS: FOR LOOP

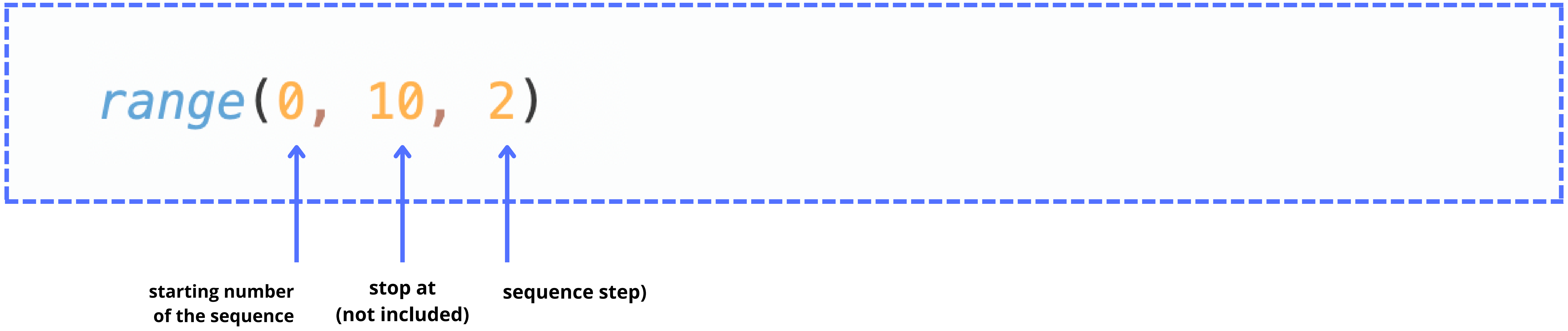
```
name = "Andrea"

for letter in name:
    print(letter)
```

- letter variable will store, on each iteration, the single element contained in the iterable oject
- so each character in the string object used in the for loop will be printed separately

LOOPS: FOR LOOP

You can use a range object to iterate and perform actions

- In order to create a range object, you'll need to use the range() function
- range() function returns an iterable object
- range object is an order sequence of numbers from start to stop passed as arguments
- the range() function is called a "Generator" function



```
range(0, 10, 2)
```

the only required argument is the stop one. If a single argument is passed to the range() function, the resulting sequence will starts from 0 to the value passed as argument(excluded) with step 1

```
range(10)
```

you can use the len() function to create sequences with same length of a selected element

```
a = "Test"

range(len(a))
```

LOOPS: FOR LOOP

use range objects (they are iterables) to perform for loops:

```python
a = "Test"

for i in range(len(a)):
    print(f"Character on position {i + 1}: {a[i]}")
```

in this case, the variable i holds a different number in every iteration for each number in the selected range:

```
Character on position 1: T
Character on position 2: e
Character on position 3: s
Character on position 4: t
```

LOOPS: FOR LOOP

You can nest multiple loops or conditional blocks inside loops:

```
a = "Test"

for i in range(len(a)):
    if a[i].lower() not in "aeiou":
        print(f"{a[i]} is not a vocal")
```

EXERCISES