

# FOR LOOPS

# RANGE() AND FOR LOOPS

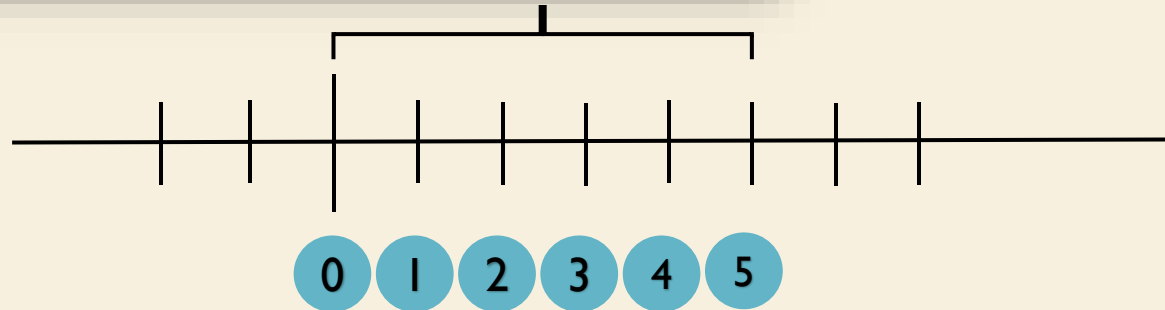
- Before we talk about the for loop we must talk about the range() function
  - The range() function represents a sequence of integers
  - the range function takes 3 arguments:
    - 1. The beginning of the range. 0 if no number is provided
    - 2. The end of the range, this number must be provided.
      - Not that the end value is not inclusive in the range (it will go up to but not including the number).
    - 3. The step of the range. Set to 1 if no number is provided

# RANGE() AND FOR LOOPS

- You can also think of the range() function as a range on a number line

```
1  for number in range(5):  
2  |     print(number)
```

The range goes from 0 up to five (but the five won't be included)



# RANGE() AND FOR LOOPS

- Consider this example

```
1  for number in range(5):  
2  |      print(number)
```

- Here only one number is provided to the range() function which means that the start value will be 0 and the step value will be 1
- The number provided, 5 in this case is the end value
- That means that in this example the range() function represents the sequence: 0, 1, 2, 3, 4
  - The variable **number** will take the value of each value in the sequence, one at a time and print each one

# RANGE() AND FOR LOOPS

- Consider these loops, Even though the range function is different in every example they all do the same thing

```
1  for number in range(5):  
2  |    print(number)
```

```
1  ☐ for number in range(0, 5):  
2  |    print(number)
```

```
1  ☐ for number in range(0, 5, 1):  
2  |    print(number)
```

# RANGE() AND FOR LOOPS

- Consider this example

```
1  for number in range(2, 5):  
2      print(number)
```

- In this example two numbers are provided to the range() function which means that the start value will be 2, the end value 5 and the step value will be 1
- Here the range() function represents the sequence: 2, 3, 4

# RANGE() AND FOR LOOPS

- Do note that the iteration variable can have any name you choose
- These two loops do the same thing
- But try to choose a name that describes the value

```
1  for number in range(2, 5):  
2      print(number)
```

```
1  for x in range(2, 5):  
2      print(x)
```

# RANGE() AND FOR LOOPS

- It is not necessary to use the number of the range
- As an example, let's say you just want to print something 3 times
  - In that case you can use the range function like this

As you can see the x variable is not used within the loop

```
1  for x in range(3):  
2  | ... print("hello-world!")
```



# FOR LOOPS VS. WHILE LOOPS

- These two examples do the same thing, one using a for loop and the other one using a while loop
- It is pretty clear that the for loop example is much shorter and neater

```
1  for number in range(1, 4):  
2  |      print("Mississippi", number)
```

```
1  counter = 1  
2  
3  while counter <= 3 :  
4  |      print("Mississippi", counter)  
5  |      counter += 1
```

# RANGE() AND FOR LOOPS

- The for statement does not have to use the range() function
  - You can iterate over each character of a string with the for loop
  - You can iterate over each value in a list with a for loop (more on lists later)

```
1 word = "pineapple"
2
3 for letter in word:
4     print(letter)
```

# FOR-LOOPS

- Consider this example
  - Here we see what happens in each iteration of the loop

1

The value of x is 1. The body of the loop executes and the value of  $x * 2$  is printed.

2

The value of x is 2. The body of the loop executes and the value of  $x * 2$  is printed.

3

The value of x is 3. The body of the loop executes and the value of  $x * 2$  is printed.

4

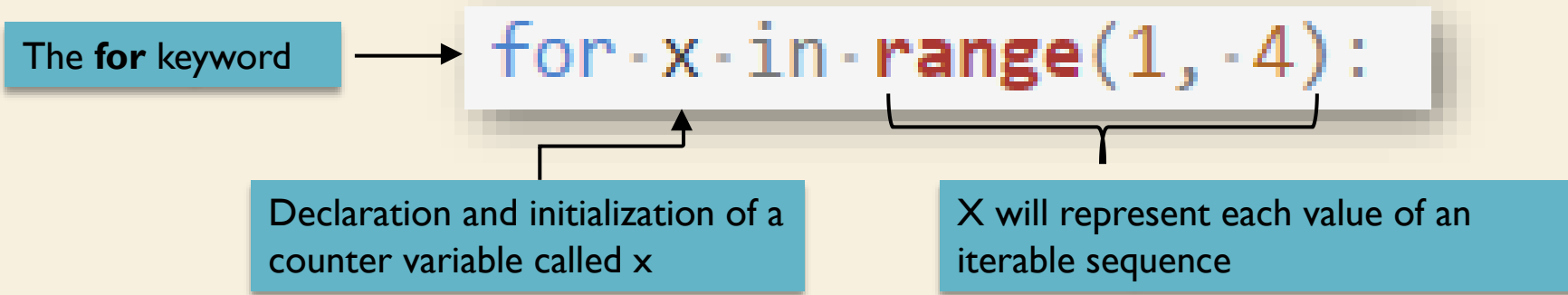
The value of x is 4 and the loop stops its execution.

```
1  for x in range(1, 4):  
2  |    print(x * 2)
```

# FOR-LOOPS

## THE SYNTAX

- Lets analyze the syntax of the **for**-loop



- The flow of this **for-loop** is as follows
  - A variable representing a counter is declared and initialized to the start value of the range
  - The value 1 is added to the i variable at the end of every iteration

# FOR-LOOPS

## MORE EXAMPLES

```
for x in range(4, -0, -1):  
    print(x)
```

Here is a loop that goes from 4 down to 0 and prints each number of that sequence

```
for x in range(-2, -6, -2):  
    print(x)
```

Here is a loop that goes from -2 up to 6 prints every other number of the sequence because the step parameter is 2

```
for x in range(0, -110, -10):  
    print(x)
```

Here is a loop that goes from 0 upto 110 and prints every tenth number of the sequence sequence because the step parameter is 10

# FOR-LOOPS

## FOR-LOOPS VS.WHILE-LOOPS

- Remember!
  - Everything you can do with a **for-loop**, you can also do with a **while-loop**
- These two code examples do the exact same thing with different kinds of **loops**

```
1  for x in range(5):  
2      print("hello!")
```

```
1  x = 0  
2  while x < 5:  
3      print("hello!")  
4      x += 1
```