HackerFrogs Afterschool
Python Programming Basics: Part 4

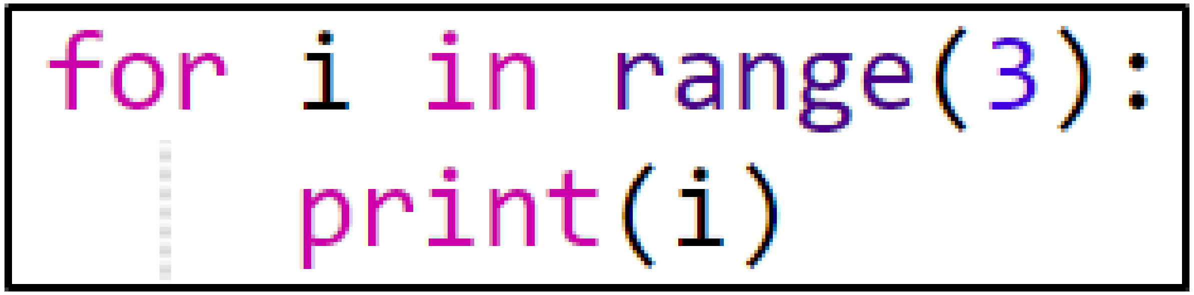
```
Class:
Programming

Workshop Number:
AS-PRO-PY-04

Document Version:
1.0

Special Requirements:
Completion of AS-PRO-PY-03
```

What We Learned Before

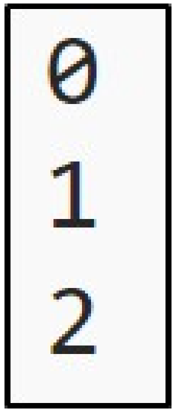
This workshop is the fourth and final class for intro Python programming.

During our last workshop, we learned about a few programming concepts through Python, including the following:

Programming Loops



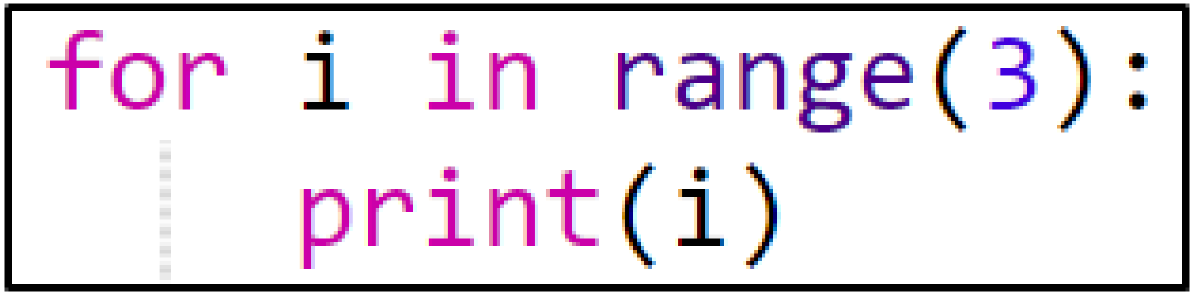
```
for i in range(3):
    print(i)
```



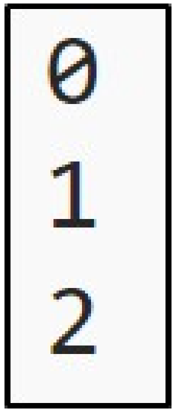
```
0
1
2
```

Loops are tools that programmers use to run the same instructions multiple times.

Programming Loops



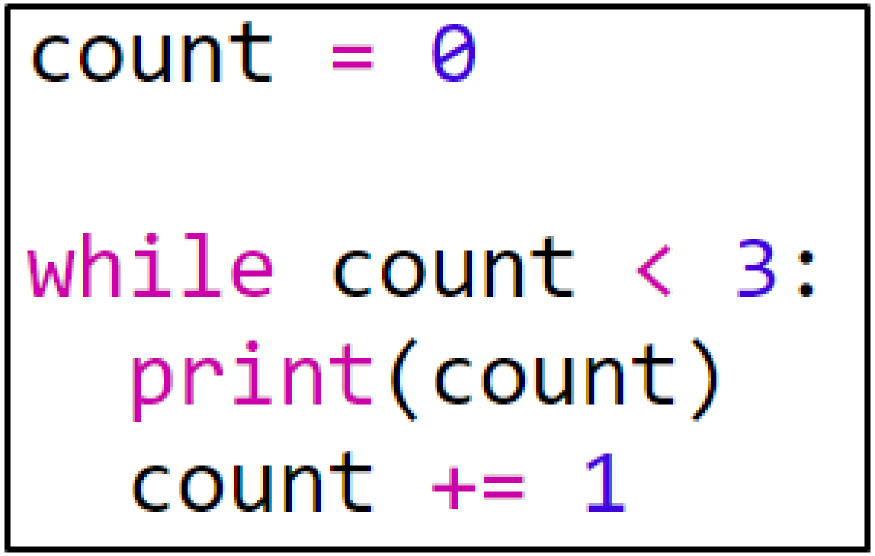
```python
for i in range(3):
    print(i)
```



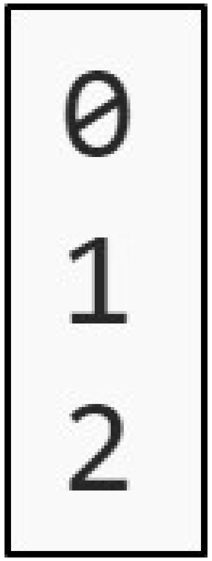
```
0
1
2
```

Loops can either run a preset number of times before terminating, or iterate through all items in a list or string, in the case of **for loops**...

Programming Loops



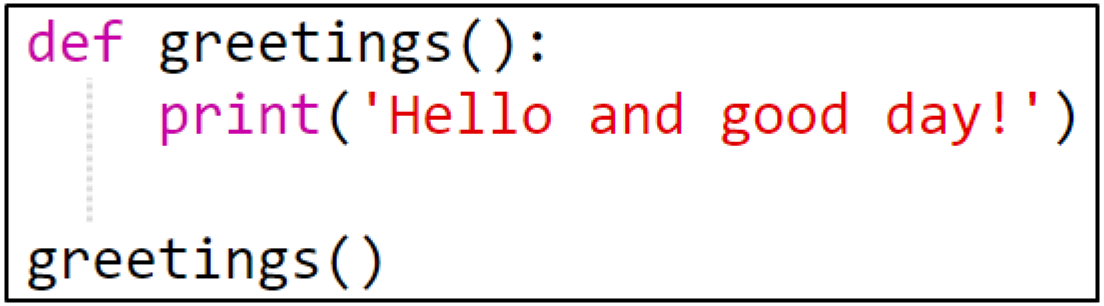
```python
count = 0

while count < 3:
  print(count)
  count += 1
```



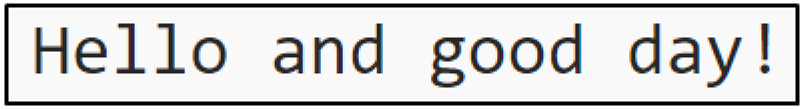
```
0
1
2
```

Or run continuously until a predetermined trigger state is achieved, in the case of **while loops**.

Functions



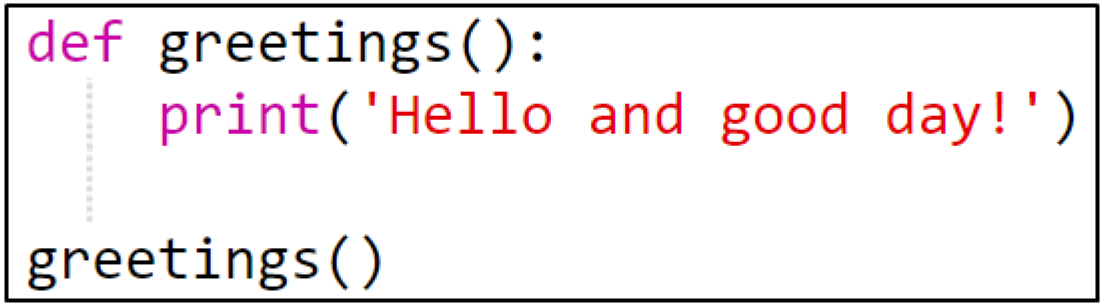
```python
def greetings():
    print('Hello and good day!')

greetings()
```



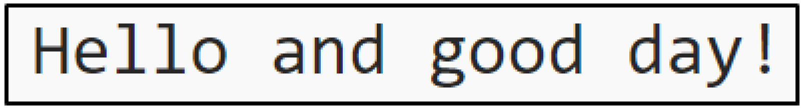
```
Hello and good day!
```

Functions are defined blocks of code that can be executed multiple times in the same program.

Functions



```python
def greetings():
    print('Hello and good day!')

greetings()
```



```
Hello and good day!
```

Executing a function is usually referred to as “calling a function”, or “a function call”.

Classes & Objects

```
print(type(1), type('word'))
```

```
<class 'int'> <class 'str'>
```

Python is an object-oriented programming language, and as such all data in Python code are considered objects.

Classes & Objects

```
print(type(1), type('word'))
```

```
<class 'int'> <class 'str'>
```

All objects belong to a class, and an object's class indicates which built-in functions, methods, and variables it has.

This Workshop's Topics

New topics for this session:

Dictionaries

This Workshop's Topics

New topics for this session:

Dictionaries

Dictionaries

```
country_capitals = {"England":"London", "Canada":"Ottawa"}
```

Dictionaries in Python are similar to other container objects, such as lists, except that data stored in dictionaries are pairs of keys / values, as opposed to individual items.

Dictionaries

```
country_capitals = {"England":
"London", "Canada": "Ottawa"}
```

Dictionaries can be identified by a pair of curly braces, within which are pairs of keys / values, each pair separated by commas. The key and value are themselves separated by a colon.

Dictionaries

```
country_capitals = {
  "England": "London",
  "Canada": "Ottawa"
}
```

Dictionaries can also be initialized in the format above, which is easier-to-read.

Iterating Over Dictionaries

```python
country_capitals = {"England": "London", "Canada": "Ottawa"}

for i, j in country_capitals.items():
  print(i, j)
```

```
England London
Canada Ottawa
```

Items in a dictionary can be iterated over using a for loop, but we must use the **items** method, which renders the dictionary as a list with tuples as the content, which hold the key / value pairs.

Adding Keys / Values

```python
friend_city = {'Carl': 'Toronto', 'Rachel': 'New York'}
friend_city['Bobby'] = 'Texas'
print(friend_city)
```

```
{'Carl': 'Toronto', 'Rachel':
'New York', 'Bobby': 'Texas'}
```

To add entries to an existing dictionary, simply define the dictionary's index with the name of the key, then supply the value on the other side of the equals sign, as shown above.

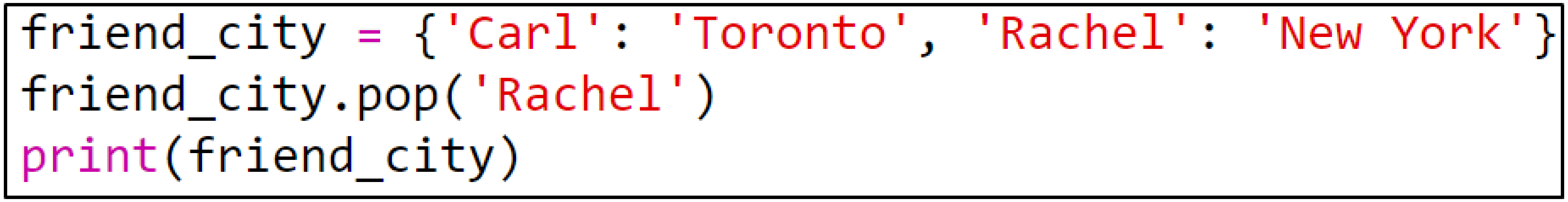
Removing Keys / Values

```
friend_city = {'Carl': 'Toronto', 'Rachel': 'New York'}
del friend_city['Carl']
print(friend_city)
```

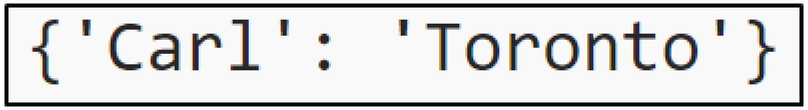
```
{'Rachel': 'New York'}
```

To remove entries from a dictionary, there are two ways. The first way is to use the **del** keyword along with the dictionary key as the index name.

Removing Keys / Values



```python
friend_city = {'Carl': 'Toronto', 'Rachel': 'New York'}
friend_city.pop('Rachel')
print(friend_city)
```



```
{'Carl': 'Toronto'}
```

The other way of removing entries from dictionaries is to the **pop** dictionary method, as illustrated above.

Dictionaries Exercise

Let's practice using Python dictionaries at the following URL:

https://learnpython.org/en/Dictionaries

Dictionaries Quiz Q1

What is the output of the following code?

```
fruit_colors = {'apple':'red', 'banana':'yellow'}
for i in fruit_colors:
  print(i)
```

A) apple:red, banana:yellow
B) apple banana
C) apple red banana yellow

Dictionaries Quiz Q1

What is the output of the following code?

```
fruit_colors = {'apple':'red', 'banana':'yellow'}
for i in fruit_colors:
  print(i)
```

A) apple:red, banana:yellow
B) apple banana
C) apple red banana yellow

Dictionaries Quiz Q2

```
fruit_colors = {'apple':'red', 'banana':'yellow'}
```

How do we add the entry `'cherry':'red'`?

A) `fruit_colors['cherry'] = 'red'`
B) `fruit_colors.update({'cherry': 'red'})`
C) `fruit_colors + {'cherry': 'red'}`
D) Either A or B

Dictionaries Quiz Q2

```
fruit_colors = {'apple':'red', 'banana':'yellow'}
```

How do we add the entry `'cherry':'red'`?

A) `fruit_colors['cherry'] = 'red'`
B) `fruit_colors.update({'cherry': 'red'})`
C) `fruit_colors + {'cherry': 'red'}`
D) Either A or B

Dictionaries Quiz Q3

```
fruit_colors = {'apple':'red', 'banana':'yellow'}
```

How do we remove the `'banana'` entry?

A) `fruit_colors['banana'] = False`
B) `fruit_colors.pop('banana')`
C) `del fruit_colors['banana']`
D) Either B or C

Dictionaries Quiz Q3

```
fruit_colors = {'apple':'red', 'banana':'yellow'}
```

How do we remove the `'banana'` entry?

A) `fruit_colors['banana'] = False`
B) `fruit_colors.pop('banana')`
C) `del fruit_colors['banana']`
D) Either B or C

Dictionaries Review Exercise

Before we finish, let's a write a program which features many of the concepts we learned in this workshop.

A game developer wants to write a program that keeps track of a game character's statistics, as well as function that adjusts one of those statistics.

Workshop Review Exercise

We can write the program on one of the learnpython.org website interactive windows.

The program should have a dictionary named `stats` with the following key / value pairs:

'name': 'player'
'money': 0
'health': 100

Workshop Review Exercise

The program should have a function named “Damage” which updates the dictionary's Health key, decreasing it by 20 when the function is called.

The Damage function should be called, then the dictionary should be printed to the console, showing the newly updated Health key.

Workshop Review Exercise

- a dictionary with `'name': 'player', 'money': 0, 'health': 100`

- a function called **damage**, which decreases the value of the dictionary's **health** value by 20

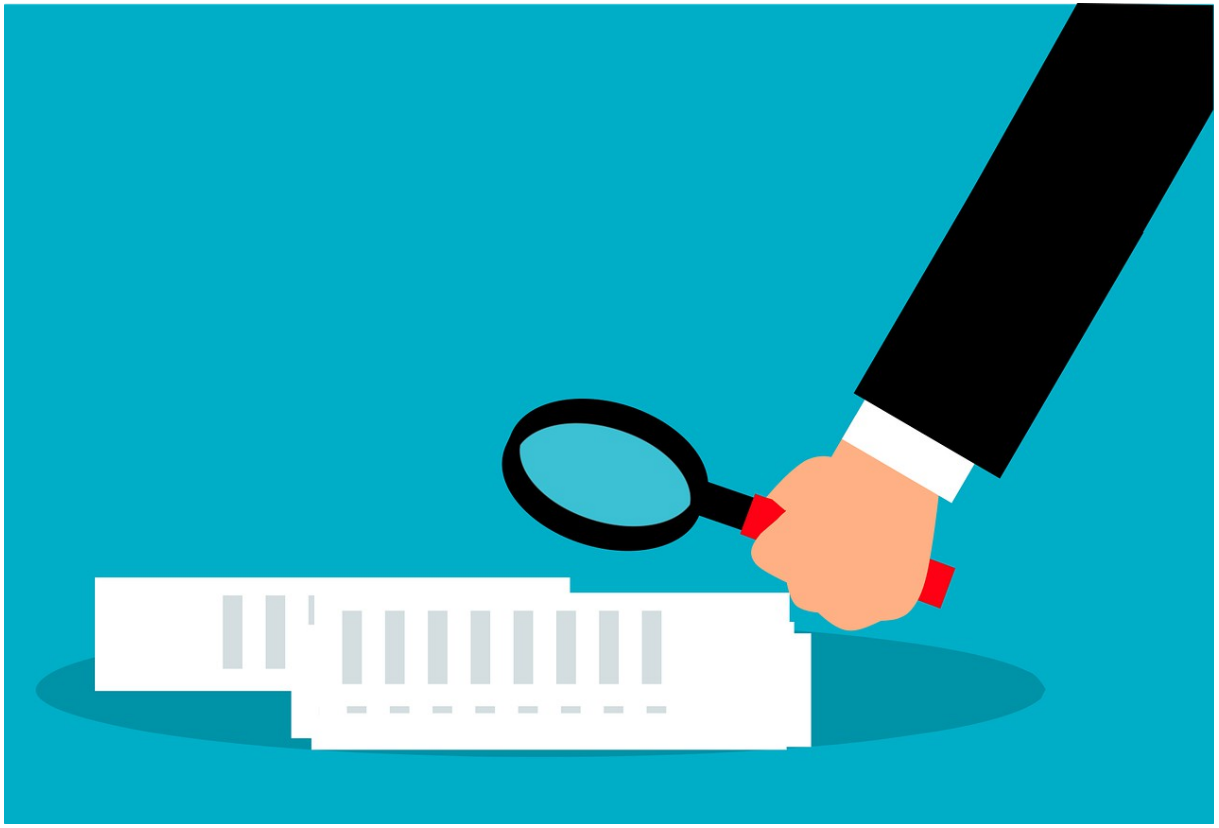
- the end of the program should be a function call to the **damage** function, then print out the contents of the dictionary.

Workshop Review Exercise

The following code fulfills the program requirements:

```
statistics = {'name': 'player', 'money': 0, 'health': 100}

def damage():
  statistics['health'] -= 20

damage()
print(statistics)
```

Summary



Let's review the programming concepts we learned in this workshop:

Dictionaries

```
country_capitals = {"England":"London", "Canada":"Ottawa"}
```

Dictionaries in Python are similar to other container objects, such as lists, except that data stored in dictionaries are pairs of keys / values, as opposed to individual items.

Dictionaries

```
john_vital_stats = {
  "height": 170,
  "weight": 90.2,
  "age": 25,
  "gender": "male"
}
```

Dictionaries are useful for recording different key / value pairs that all pertain to one subject, or one common value across a number of subjects.

What's Next?

In the next HackerFrogs AfterSchool Python programming workshop, we'll tackle the last topic in our series, modules and packages!

Extra Credit

Looking for more study material on this workshop's topics?

See this video's description for links to supplemental documents and exercises!

Until Next Time, HackerFrogs!