SKILLS NETWORK

Python Cheat Sheet: The Basics

Python Data Types

String

Series of characters or data stored as text

```
my_string = "Hello"
```

String Operations

```
# returns the string with all uppercase letters
my_string.upper()
```

```
# returns the length of a string
len(my_string)
```

```
# returns the index of the first instance of the string inside the
# subject string, otherwise -1
my_string.find('l')
```

```
# replaces any instance of the first string with the second in my_string
my_string.replace('H', 'C')
```

Integer

A whole number

```
my_integer = 12321
```

Float

A decimal number

```
my_decimal = 3.14
```

List

Changeable collection of objects

```
my_collection = [1, 1, 3.12, False, "Hi"]
```

List Operations

```
# returns the length of a list
len(my_collection)
```

```
# Add multiple items to a list
my_collection.extend(["More", "Items"])
```

```
# Add a single item to a list
my_collection.append("Single")
```

```
# Delete the object of a list at a specified index
del(my_collection[2])
```

```
# Clone a list
clone = my_collection[:]
```

```
# Concatenate two lists
my_collection_2 = ["a", "b", "c"]
my_collection_3 = my_collection + my_collection_2
```

```
# Calculate the sum of a list of ints or floats
number_collection = [1,2,3,4.5]
sum(number_collection)
```

```
# Check if an item is in a list, returns Boolean
item in my_collection
# Check if an item is not in a list, returns Boolean
```

SKILLS NETWORK

Python Cheat Sheet: The Basics

Loops

For Loops

```
for x in range(x):
    # Executes loop x number of times
```

```
for x in iterable:
    # Executes loop for each object in an iterable like a string, tuple,
list, or set
```

While Loops

```
while statement:
    # Executes the loop while statement is true
```

Conditional Statements

```
if statement_1:
  # Execute of statement_1 is true
elif statement_2:
  # Execute if statement_1 is false and statement_2 is true
else:
  # Execute if all previous statements are false
```

Try/Except

```
try:
   # Code to try to execute
except a:
   # Code to execute if there is an error of type a
except b:
```

Webscraping

```
# Import BeautifulSoup
from bs4 import BeautifulSoup
```

```
# Parse HTML stored as a string
soup = BeautifulSoup(html, 'html5lib')
```

```
# Returns formatted html
soup.prettify()
```

```
# Find the first instance of an HTML tag
soup.find(tag)
```

```
# Find all instances of an HTML tag
soup.find_all(tag)
```

Requests

```
# Import the requests library
import requests
```

```
# Send a get requests to the url with optional parameters
response = requests.get(url, parameters)
```

```
# Get the url of the response
response.url
# Get the status code of the response
response.status_code
# Get the headers of the request
response.request.headers
# Get the body of the requests
```