

# Reference guide: Python concepts from module 4

## File operations

The following functions, methods, and keywords are used with operations involving files.

### **with**

Handles errors and manages external resources

```
with open("logs.txt", "r") as file:
```

Used to handle errors and manage external resources while opening a file; the variable `file` stores the file information while inside of the `with` statement; manages resources by closing the file after exiting the `with` statement

### **open( )**

Opens a file in Python

```
with open("login_attempts.txt", "r") as file:
```

Opens the file "login\_attempts.txt" in order to read it ( "r" )

```
with open("update_log.txt", "w") as file:
```

Opens the file "update\_log.txt" into the variable `file` in order to write over its contents ( "w" )

```
with open(import_file, "a") as file:
```

Opens the file assigned to the `import_file` variable into the variable `file` in order to append information to the end of it ( "a" )

### **as**

Assigns a variable that references another object

```
with open("logs.txt", "r") as file:
```

Assigns the `file` variable to reference the output of the `open( )` function

## **.read()**

Converts files into strings; returns the content of an open file as a string by default

```
with open("login_attempts.txt", "r") as file:  
    file_text = file.read()
```

Converts the file object referenced in the `file` variable into a string and then stores this string in the `file_text` variable

## **.write()**

Writes string data to a specified file

```
with open("access_log.txt", "a") as file:  
    file.write("jrafael")
```

Writes the string "jrafael" to the "access\_log.txt" file; because the second argument in the call to the `open()` function is "a", this string is appended to the end of the file

## Parsing

The following methods are useful when parsing data.

### **.split()**

Converts a string into a list; separates the string based on the character that is passed in as an argument; if an argument is not passed in, it will separate the string each time it encounters whitespace characters such as a space or return

```
approved_users = "elarson,bmoreno,tshah".split(",")
```

Converts the string "elarson,bmoreno,tshah" into the list

`["elarson", "bmoreno", "tshah"]` by splitting the string into a separate list element at each occurrence of the `" , "` character

```
removed_users = "wjaffrey jsoto abernard".split()
```

Converts the string "wjaffrey jsoto abernard" into the list

`["wjaffrey", "jsoto", "abernard"]` by splitting the string into a separate list element at each space

### **.join()**

Concatenates the elements of an iterable into a string; takes the iterable to be concatenated as an argument; is appended to a character that will separate each element once they are joined into a string

```
approved_users = ",".join(["elarson", "bmoreno", "tshah"])
```

Concatenates the elements of the list [ "elarson", "bmoreno", "tshah" ] into the string "elarson,bmoreno,tshah" , separating each element with the " , " character within the string

---

Revision #2

Created 3 January 2024 00:32:55 by naruzkurai

Updated 3 January 2024 00:33:30 by naruzkurai