File Input/Output and Funtions

Creating, reading, and writing to files is a core portion of an investigation.

Writing functions is an integral portion of programming as whole.

File I/O

,he built-in command `open()` accepts filenames as parameters, with a second parameter that says how the file should be opened. `w` is for writing, `r` is for reading and `a` is for appending to the file. (python creates an object related to the file itself)

Objects in the simpliest form are variables with additional definitions.

you can also use `r+` for reading and writing to the file, the stream will be positioned at the beginning of the file.

ther is also `a+` are opened for reading and appending.

files that do not contain plaintext can also manipulated with bytes using `rb` and written to with `wb` .

Once the file object is created and loaded into the program, built-in methods and functions for manipulating it in the system exist.

```
Text = File.read()
```

- the method reads the entire file and saves it as a string within a variable.
- The disadvantage of the method is the whole file is loaded at once. **crash potential**

```
File.readline()
```

- at the EOF this method will print `""` , an empty string.

you need to specify how you will open the file if you want to write to it. `file = open(locationInString, "a");`
`File.write("this is extra stuff \n")`

```
file.close()
```

- after reading a file, it is optimal to close it in order to reserver performance and memory

What if I want to continue to edit the file? use `file.flush()`

```
with
```

- tells the program to close the file automatically.

```
with open(locatino, "r") as file:
    for line in file:
        print line,
```

After the `with` command we'll write the `open()` with the usual parameters and add `as` and the variable name for the file object.

`with` runs error handling on the backend.

*Exercises

1. What is the type of object returned by the open function?
2. Create a replication machine, anything written to the prompt will be replicated in two different files.
3. Write a script that accepts the lyrics of a song file and prints the lyrics.
4. Write a script that accepts the location of an image and copies the image to the desktop
5. Practice both methods of `open()`
6. Write a script that takes in a file from the user and searches on the desktop if a file matches the content of it. Print true or false.
7. Explain the usage of the `r` parameter in `open()`

Functions

A function is a code fragment that has a name, can be called, and might accpet parameters.

functions begin with the keyword `def` and then the name of the function and parenthesis.

```
def hello(): print "hello"
```

Each time in the code, `hello()` appears, the console will print `hello` .

Functions can accept multiple parameters, and they are handled ordinally.

Functions can also `return` values to the main program.

```
def return_number():
    x = 1
    y = 2
    return x, y
```

Let's look at the following function:

```
def nothing():
    return None
```

What is the value of x in the following line: `x = nothing()`

“None” is a reserved word in the Python language that means - empty value. There are 3 situations in which a function returns a “None” value: * The function does not return. * The function has a return but without any value or variable, only return. * The function has a return none.

Scope

Scope is the space in which a variable can exist or influence execution of the code.

```
def func():
    txt = "hi"
    print txt
func()
print txt
```

- This program will crash; because it cannot "see" the variable `txt` .
- This variable is defined in, and only exist in, the function.

to solve this we make **global** variables

```
name = "mud"
def hello():
    text = "hi"
    print txt, name
hello()
```

Global variables, if they are redeclared, will also cause a program to crash.

```
text = "hello"
def nothing():
    text += "everyone"
    print text

nothing()
```

This program will crash because of the redeclaration of `text`

but by using the keyword `global` inside of the function definition, the program will be able to continue execution.

```
text = "hello"
def nothing():
    global text
    text += "everyone"
    print text

nothing()
```

Exercises

1. Write a function that accepts two values and returns their product
2. Write a function that accepts three values, two numbers and a symbol, and performs the mathmatical operation on the numbers. a. Write a function for all 4 of the math signs. `*, /, +, -` b. Remember to use error handling for the divide by zero errors!

3. Write a function called factorial that returns the result of `5!` (120)

4. Write a function that gets a message and prints "finished" at the end.