Programming with Python

Lesson 3: Writing, Reading and Getting Funky with Functions!

November 15th, 2016

Last week’s goals

Last week's goals

- We have learnt about conditionals

Last week's goals

- We have learnt about conditionals
- We have used conditionals to make our calculator be able to add, subtract, divide or multiply

Last week's goals

- We have learnt about conditionals
- We have used conditionals to make our calculator be able to add, subtract, divide or multiply
- We have used conditionals to handle errors

Last week's goals

- We have learnt about conditionals
- We have used conditionals to make our calculator be able to add, subtract, divide or multiply
- We have used conditionals to handle errors
- We have learnt about loops and used them to make our program much more usable

Last week's goals

- We have learnt about conditionals
- We have used conditionals to make our calculator be able to add, subtract, divide or multiply
- We have used conditionals to handle errors
- We have learnt about loops and used them to make our program much more usable

Messing with files

Where should we go now with the calculator?

Messing with files

Where should we go now with the calculator?
We are going to work through a log for our calculator!

File input and output

File input and output

In python, the *myFileObj = open()* function will open a file.

File input and output

In python, the *myFileObj* $=$ *open()* function will open a file. When you run *open()* you get an object which represents the file.

File input and output

In python, the *myFileObj* = *open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

File input and output

In python, the *myFileObj = open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

- *r*

File input and output

In python, the $myFileObj = open()$ function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

- *r* read

File input and output

In python, the *myFileObj = open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

- *r* read
- *w*

File input and output

In python, the *myFileObj* = *open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

- *r* read
- *w* write

File input and output

In python, the *myFileObj = open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

- *r* read
- *w* write
- *a*

File input and output

In python, the *myFileObj* = *open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

r	read
w	write
a	append

File input and output

In python, the *myFileObj* = *open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

- *r* read
- *w* write
- *a* append
- *w*+

File input and output

In python, the *myFileObj* = *open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

- *r* read
- *w* write
- *a* append
- *w+* read & write

File input and output

In python, the *myFileObj = open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

r	read
w	write
a	append
w+	read & write
a+	

File input and output

In python, the *myFileObj* = *open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

r	read
w	write
a	append
w+	read & write
a+	read & append

File input and output

In python, the *myFileObj* = *open()* function will open a file. When you run *open()* you get an object which represents the file.
Five modes for opening a file:

r	read
w	write
a	append
w+	read & write
a+	read & append

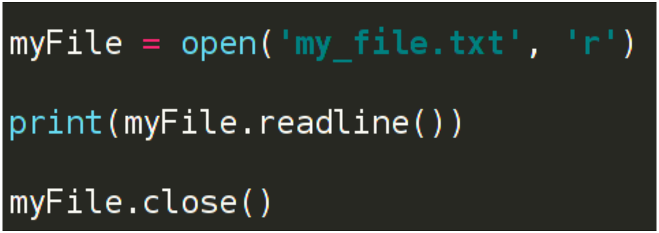
Which mode should we use for a log for our calculator?

The *myFileObj.close()* function will close a file.

The *myFileObj.close()* function will close a file.
The *myFileObj.write()* function writes to a file.

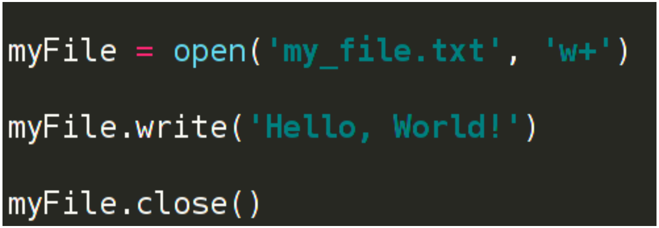
The *myFileObj.close()* function will close a file.
The *myFileObj.write()* function writes to a file.
The *myFileObj.read()* function reads from a file.

Reading from a file



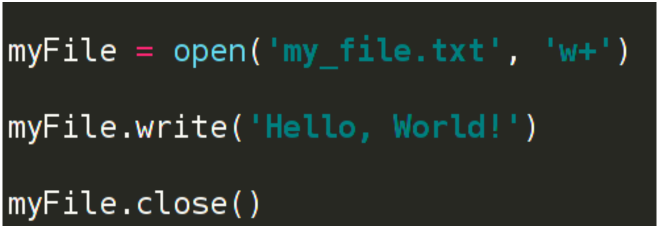
```
myFile = open('my_file.txt', 'r')

print(myFile.readline())

myFile.close()
```

Writing to a file



```
myFile = open('my_file.txt', 'w+')

myFile.write('Hello, World!')

myFile.close()
```

Writing to a file



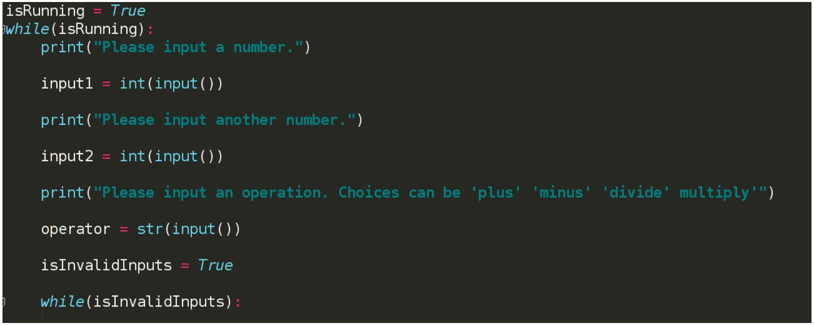
```
myFile = open('my_file.txt', 'w+')

myFile.write('Hello, World!')

myFile.close()
```

Note, if a file does not exist, it will be created.

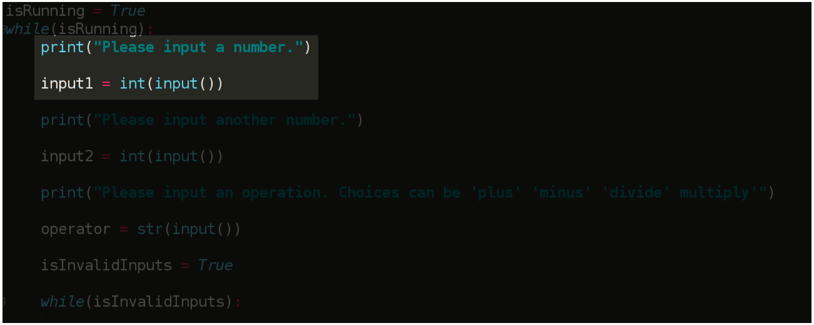
Time for a little live coding example!

Time for some code cleaning!

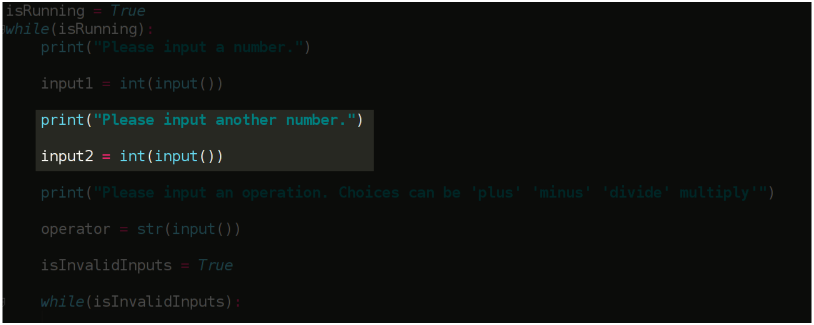
Before we add file IO to our program, we should have a quick look at it.



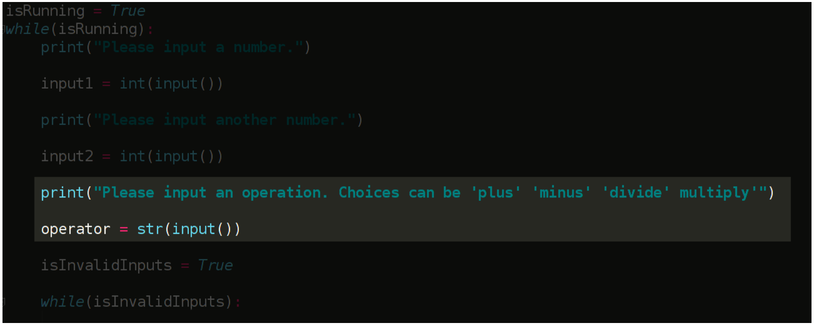
```
isRunning = True
while(isRunning):
    print("Please input a number.")

    input1 = int(input())

    print("Please input another number.")

    input2 = int(input())

    print("Please input an operation. Choices can be 'plus' 'minus' 'divide' multiply'")

    operator = str(input())

    isInvalidInputs = True

    while(isInvalidInputs):
```



```
isRunning = True
while(isRunning):
    print("Please input a number.")

    input1 = int(input())

    print("Please input another number.")

    input2 = int(input())

    print("Please input an operation. Choices can be 'plus' 'minus' 'divide' multiply'")

    operator = str(input())

    isInvalidInputs = True

    while(isInvalidInputs):
```



```
isRunning = True
while(isRunning):
    print("Please input a number.")

    input1 = int(input())

    print("Please input another number.")

    input2 = int(input())

    print("Please input an operation. Choices can be 'plus' 'minus' 'divide' multiply'")

    operator = str(input())

    isInvalidInputs = True

    while(isInvalidInputs):
```



```
isRunning = True
while(isRunning):
    print("Please input a number.")

    input1 = int(input())

    print("Please input another number.")

    input2 = int(input())

    print("Please input an operation. Choices can be 'plus' 'minus' 'divide' multiply'")

    operator = str(input())

    isInvalidInputs = True

    while(isInvalidInputs):
```

We don't want to have to repeat ourselves!

One of the most important rules of good programming is:

We don't want to have to repeat ourselves!

One of the most important rules of good programming is:

D

We don’t want to have to repeat ourselves!

One of the most important rules of good programming is:

D

R

We don’t want to have to repeat ourselves!

One of the most important rules of good programming is:

D

R

Y

We don't want to have to repeat ourselves!

One of the most important rules of good programming is:

D on't

R

Y

We don't want to have to repeat ourselves!

One of the most important rules of good programming is:

D on't

R epeat

Y

We don’t want to have to repeat ourselves!

One of the most important rules of good programming is:

D on’t

R epeat

Y ourself

We don't want to have to repeat ourselves!

One of the most important rules of good programming is:

D on't

R epeat

Y ourself

Repeating yourself makes updating/maintaining code horrible and it makes finding bugs in your code impossible!

Functions

Functions - the cause of, and solution to all of life's problems!

Getting funky with functions

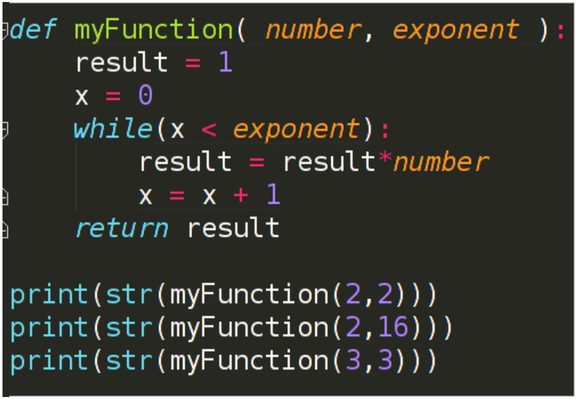
Getting funky with functions

All functions are, are bits of code put together, so we don't have to copy and paste loads of lines everywhere!

Getting funky with functions

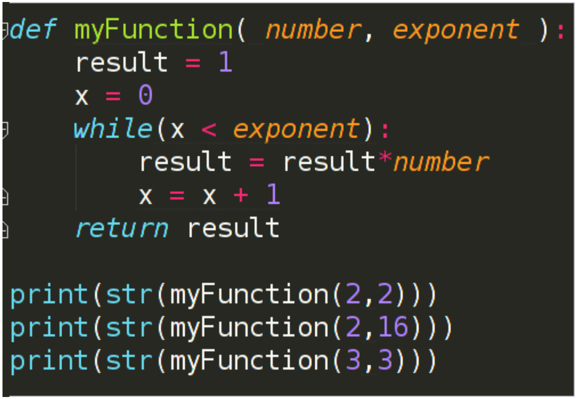
All functions are, are bits of code put together, so we don't have to copy and paste loads of lines everywhere! Functions usually take in values (called arguments or parameters) and usually return a value (though this is not always the case).

An example function



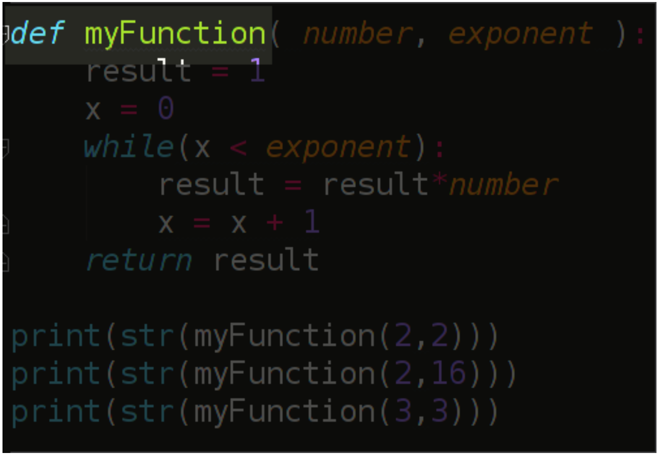
```
def myFunction( number, exponent ):
    result = 1
    x = 0
    while(x < exponent):
        result = result*number
        x = x + 1
    return result

print(str(myFunction(2,2)))
print(str(myFunction(2,16)))
print(str(myFunction(3,3)))
```

An example function

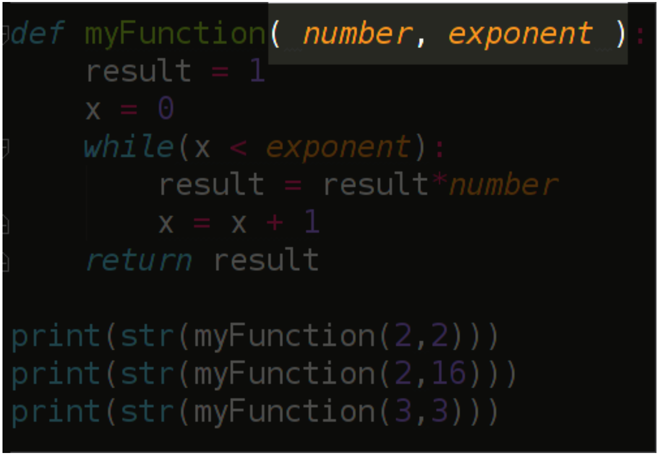


```python
def myFunction( number, exponent ):
    result = 1
    x = 0
    while(x < exponent):
        result = result*number
        x = x + 1
    return result

print(str(myFunction(2,2)))
print(str(myFunction(2,16)))
print(str(myFunction(3,3)))
```

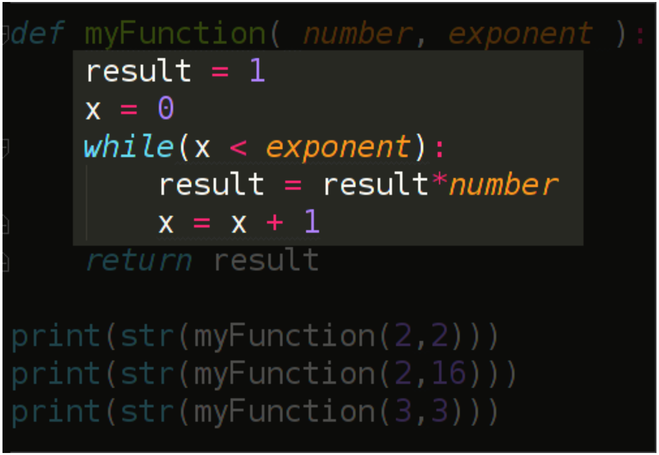
Anyone know what myFunc does?



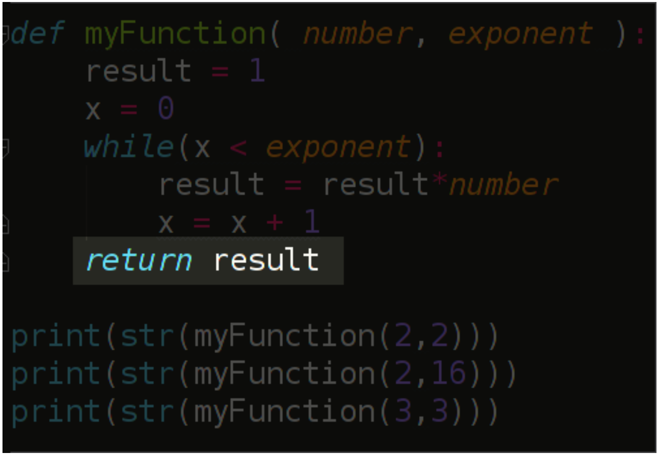
```
def myFunction( number, exponent ):
    result = 1
    x = 0
    while(x < exponent):
        result = result*number
        x = x + 1
    return result

print(str(myFunction(2,2)))
print(str(myFunction(2,16)))
print(str(myFunction(3,3)))
```



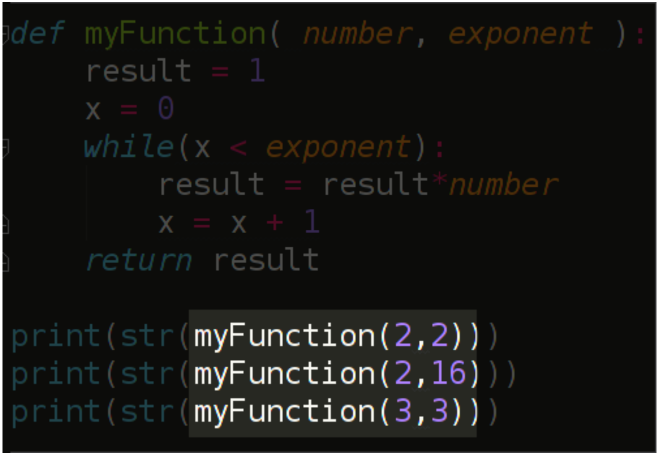
```
def myFunction( number, exponent ):
    result = 1
    x = 0
    while(x < exponent):
        result = result*number
        x = x + 1
    return result

print(str(myFunction(2,2)))
print(str(myFunction(2,16)))
print(str(myFunction(3,3)))
```



```
def myFunction( number, exponent ):
    result = 1
    x = 0
    while(x < exponent):
        result = result*number
        x = x + 1
    return result

print(str(myFunction(2,2)))
print(str(myFunction(2,16)))
print(str(myFunction(3,3)))
```



```
def myFunction( number, exponent ):
    result = 1
    x = 0
    while(x < exponent):
        result = result*number
        x = x + 1
    return result

print(str(myFunction(2,2)))
print(str(myFunction(2,16)))
print(str(myFunction(3,3)))
```



```
def myFunction( number, exponent ):
    result = 1
    x = 0
    while(x < exponent):
        result = result*number
        x = x + 1
    return result

print(str(myFunction(2,2)))
print(str(myFunction(2,16)))
print(str(myFunction(3,3)))
```

Bringing it all together

Mega live coding sesh

Bringing it all together

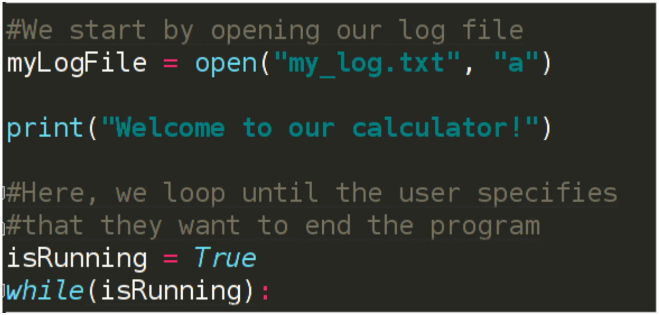
Mega live coding sesh
Challenge: Can anyone make a program which reads out the log?

Comments

Comments are lines in your code which don't do anything.

Comments

Comments are lines in your code which don't do anything. They are normally used to help explain the code.



```
#We start by opening our log file
myLogFile = open("my_log.txt", "a")

print("Welcome to our calculator!")

#Here, we loop until the user specifies
#that they want to end the program
isRunning = True
while(isRunning):
```

That’s all for tonight

To summarise:

That's all for tonight

To summarise:

- We have learnt about opening and closing files

That's all for tonight

To summarise:

- We have learnt about opening and closing files
- We have learnt to write to, and read from files using the file object

That's all for tonight

To summarise:

- We have learnt about opening and closing files
- We have learnt to write to, and read from files using the file object
- We have learnt about the importance of functions and keeping code clean

That's all for tonight

To summarise:

- We have learnt about opening and closing files
- We have learnt to write to, and read from files using the file object
- We have learnt about the importance of functions and keeping code clean
- We have used functions to make our code more readable, as well as to write to a log

For next week

Source code plus lecture slides will be available online soon after the lesson.
If you are new to HackSocNotts, please join us on *http://hacksocnotts.slack.com*.
If you have any questions, feel free to ask now or over slack.