Unit 3: Text-Based Programming

Lesson 4

Overview

So far you have only been using print to output text/strings that are hard coded into your program; your program currently lacks the ability to interact with the user.

You will be using **input** to prompt the user to enter some data.

In this activity you will:

- learn how to use input to get strings from the user

Getting Input from the User

`input` will put a message on the screen, wait for the user to give a response and press Enter, and then store the response as a string in a variable so you can use it in your program.

`print` and `input` are examples of **built-in-functions**. Built-in-functions are pre-made programs which are typically created to perform a single task. Python has many inbuilt functions allowing you to perform a large range of tasks.

Getting Input from the User

- Open your hello_world.py program.

- Replace line 3 with the following line:

```
my_name = input("What is your name? ")
```

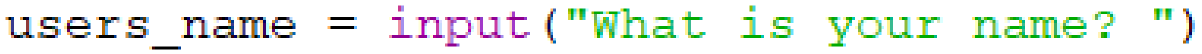
- You should put a space after your question mark.

- This will save whatever you type in as the variable my_name as a string.

Getting Input from the User

- Open your hello_world.py program.

- Replace line 3 with the following line:



```
users_name = input("What is your name? ")
```

- You should put a space after your question mark.

Understanding the Code

Breaking down this line of code you can see what each element does:

- `users_name =` creates a variable called users_name and assigns it a value
- `input()` uses the function input to display a prompt and capture the user's response
- `"What is your name? "` – the value between the () – is the prompt which will be output to the screen

Now the text the user entered is stored in a variable (users_name) you can use the data in your program.

- Lets use an f-string to print to output the user’s name.

```
print(f"Hello {users_name}, welcome.")
```

Note: you should enter your name in the REPL(in Mu) or in the Shell(in IDLE) after the “What is your name? ” prompt and press Enter when you have done so.

Code

```
print("Hi")
print("I am a computer program")
users_name = input("What is your name? ")
print(f"Hello {users_name}, welcome.")
```

Output

```
Hi
I am a computer program
What is your name? Robert
Hello Robert, welcome.
>>>
```

The black text is what I typed

Syntax

Now you are starting to create more complex instructions, it’s time to talk about syntax, the inevitable **“syntax errors”,** and tips for resolving them.

Syntax is a set of rules which describe how the code must be laid out. Computers are sticklers for precision, and if you break the rules they will tell you in the only way they know how: with errors. They just don’t have the imagination to cope with anything unusual!

Let's talk through some of the syntax rules for this line of code:

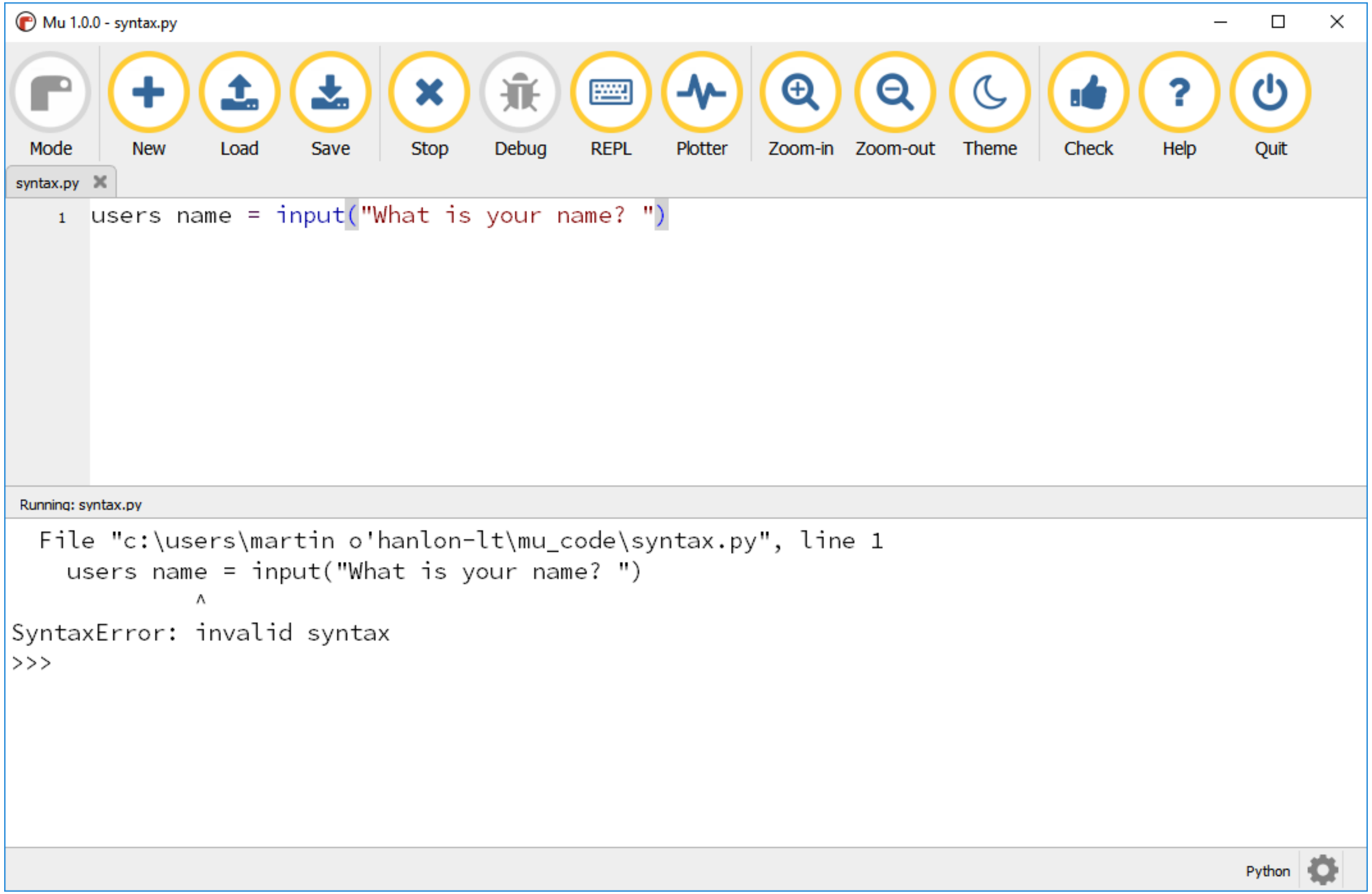
```
users_name = input("What is your name? ")
```

`users_name` – this is the variable name and there are syntax rules about how variables can be named, variable names cannot contain a space, they cannot start with a number and they are case sensitive.

If you were to run the following code, you would be presented with an error (also below):

```
users name = input("What is your name? ")
               ^

SyntaxError: invalid syntax
```



```
users name = input("What is your name? ")
```

```
  File "c:\users\martin o'hanlon-lt\mu_code\syntax.py", line 1
    users name = input("What is your name? ")
             ^
SyntaxError: invalid syntax
>>>
```

Python is telling you that you have broken the rules and you need to fix the problem otherwise your program cannot be run.

In this snippet of code, the function input is being used. We must give it a prompt for the user, and the syntax rules say we must enclose this prompt in round brackets (). Missing out either or both of the brackets will result in an error.

```
users_name = input "What is your name? "
                                       ^
SyntaxError: invalid syntax
```

Tip: when presented with a SyntaxError, use the error message to find the **approximate** position of the error. **It could be the line before or after the one shown**. Be critical of your own code or get someone else to look over it for you.