Lesson 0: Introduction To Functions, And Variables

Some basic information

1. Code runs line-by-line; what appears on line 1 will happen before what is on line 2. Line 2 will happen before what is on line 3. This continues until the end of the lines is reached.
2. Blank or empty lines are ignored by the computer.

What is a Function ?

A function is a tool used to tell a computer what to do. A computer will do EXACTLY what you tell it to. Nothing more, nothing less. A function is an instruction that performs a task of some sort.

A function has two parts:

1. The function name.
2. The function input. Some functions have no input.

The first function we will look at is the `print()` function .

- Its name is print
- Its input goes between the '(' and ')'

The `print()` function is a way we tell the computer to output a string. What is a string? A string is a sentence surrounded by double quotes: **"This is a string"**

In [1]:

```python
# Example 1:
print("Hello World!")
```

```
Hello World!
```

As you can see above, the `print("Hello World!")` function outputs the string given to it.

Note:
The line above the function starts with a `#` charachter. This tells the computer that we are writing a note on what is happening. The computer will ignore any lines starting with the `#` character.

The `print()` function requires a string as it's input. How do we give a function input? Whatever is between the opening '(' and closing ')' parenthesis is the input to the function . If the function takes no input, then simply put nothing between the parenthesis like so: `print()`

In [2]:

```
# Example 2:
# Good print() input
print("This is a string!")
```

```
This is a string!
```

In [4]:

```
# Example 3:
# Good print() input
print("This is a 2nd string that contains a number!")
```

```
This is a 2nd string that contains a number!
```

In [5]:

```
#Example 4:
# Good print() input
print("1, 2, 3, 4, 5")
```

```
1, 2, 3, 4, 5
```

In [6]:

```
# Example 5:
# Bad print() input
print(this is not a string and will cause an error)
```

```
  File "<ipython-input-6-3aa113146da0>", line 3
    print(this is not a string and will cause an error)
                             ^
SyntaxError: invalid syntax
```

What Is A Variable?

A Variable is a way to store data for use later. Variables are similar to a freezer. Pretend you decide to buy a tub of ice cream. You need somewhere to keep it safe until you want to eat it. So you put the ice cream in the freezer so you can have dessert after dinner. A Variable is similar in that you can store data in it for later use. Variables can store the string we learned about previously.

You can give a Variable almost any name you want, with some rules. A Variable cannot start with a number. **1wholeTubOfIceCream** is not an allowed name because it starts with the number '1'. **OneWholeTubOfIceCream** or **one_whole_tub_of_ice_cream** are allowed as either of these start with a letter. Variable names also cannot have spaces in them. **one whole tub of ice cream** is not allowed as a Variable name because it contains spaces. There are other rules, but for now just know you need to start a Variable name with a letter and you should name it something helpful.

Example: **TubOfIceCream** is much more helpful than a name such as **T**.

You put data into a Variable with the `=` character. The Variable is on the left side, and what you want to store inside of it goes on the right side of the `=` character. This may be best explained with an example.

In [8]:

```
# Example 6:
# Note that the variable name cannot have spaces,
# but the string data type can.

#  Variable Name    Data we want to store in the variable
#    |                 |
#   \/                \/
OneTubOfIceCream = "Chocolate Chip Cookie dough"

# Now we can use the print function we learned about
# and use the variable we created as its input.
print(OneTubOfIceCream)
```

```
Chocolate Chip Cookie dough
```

In [9]:

```
# Example 6.1:
# Notice that Example 6's output is equivalent to this example's output
print("Chocolate Chip Cookie dough")
```

```
Chocolate Chip Cookie dough
```

In [10]:

```
# Example 7:
# We can also call a variable one letter.
T = "Vanilla"
# Now call the print function and pass the 'T' variable as input.
print(T)
```

```
Vanilla
```

```
In [14]: # Example 8:
         # An example of an error
         One Tub Of Ice Cream = "Mint Chocolate Chip"
         # Now call the print command on this new variable we created.
         print(One Tub Of Ice Cream)
```

```
  File "<ipython-input-14-6307d2587172>", line 3
    One Tub Of Ice Cream = "Mint Chocolate Chip"
          ^
SyntaxError: invalid syntax
```

As you can see above, the Variable we tried to name **One Tub Of Ice Cream** gave an error at the output. We never even reached the `print()` function because of this error!

```
In [13]: # Example 9:
         # An example of an error
         1TubOfIceCream = "Mint Chocolate Chip"
         print(1TubOfIceCream)
```

```
  File "<ipython-input-13-0384dfabf3d2>", line 3
    1TubOfIceCream = "Mint Chocolate Chip"
                 ^
SyntaxError: invalid syntax
```

As you can see above, the Variable we tried to name **1TubOfIceCream** gave an error at the output. We never even reached the `print()` function because of this error!

Variables can be changed as you please. Below is an example of updating a variable before passing it to a function.
Notice that once you give data to a variable, it will hold that data until you change it.
Also notice that you can use the `+` charachter to connect multiple strings together. Since the variable **flavor** is holding a string, it can be added as well.

```
In [16]: # Example 10.1: (A little more complex)
         # Set a string variable to be a flavor
         flavor = "vanilla"
         # Call the print function.
         print("I want a bowl of " + flavor + " flavored ice cream")
```

```
I want a bowl of vanilla flavored ice cream
```

```
In [17]: # Example 10.2
         # Set a string variable to be a flavor
         flavor = "chocolate"
         # Call the print function
         print("I want a bowl of " + flavor + " flavored ice cream")
```

```
I want a bowl of chocolate flavored ice cream
```

In [18]:

```
# Example 10.3:
# Set a string variable to be a flavor
flavor = "mint chocolate chip"
# Call the print function.
print("I want a bowl of " + flavor + " flavored ice cream")
```

```
I want a bowl of mint chocolate chip flavored ice cream
```

Summary:

We have learned about functions, variables, and strings. Functions are instructions we provide to the computer to perform tasks. Variables are containers that hold data for us. Strings are a sentence that can be stored in variables.

Functions:

- `print()` Outputs strings or strings stored in variables.

Variables:

- Can have any name that does not start with a number and cannot have spaces between the letters of the name.
- Can hold the string data we learned about.

Assignment:

Using the `print()` function, variables, and anything else you learned, tell me each of your pets names. You may achieve this in any way possible.

Example output:

```
My pet's name is AppleJack.
My pet's name is Precious.
```

See the main.py file in this folder.
Code Skeleton to get you started (copy and paste this if you like):

In [ ]:

```
petName = ""
print("My pet's name is " + petName)

petName = ""
print("My pet's name is " + petName)
```