CyberCamp at UNK

Data types, variables definition and variables formatted output

What you will learn with this tutorial...

In this tutorial you will be introduced to the concept of ***data types****, and what are the most common data types used in Python. After that, you will understand what* ***variables*** *are, and then you will see how the output of variables values happen in the console. To finish this tutorial you will be given an assignment where you will be able to apply all these new learnt concepts. Let's start!*

1 Data Types

Data type is the way we can classify one of various types of data that our programs can have. Think a little about it, our programs can be composed of integer numbers, non integer numbers, letters (alone or in a set, like in a sentence), and many other types, as we will see.

Python let us work with a lot of these data types. Down below you can see what are the most common data types used in this language:

- **Numbers:** We use a lot of numbers when coding programs. In Python we can represent integer, float and complex numbers, this last one we will not use right now though, but it is good to keep in mind that it exists.
- **Strings:** Strings are sequence of characters. It can be a simple letter, like *y* or *n* or full words and sentences, like *Hello World!* from last tutorial.
- **Lists:** Lists are collections of items. In Python we can have lists of numbers, strings, or even mix up them together. They are really useful and easy to use.

Now that you know the concept and what data types are, we will study variables!

2 Variables

If data types are how we classify data, *variables* are where we can store this data. It's as simple as that!

Variables are composed by an ***identifier***, that is the symbolic name we associated to it, and by a ***value***, that is the data we want to be stored. This is the declaration of an integer variable in Python:

```
age = 18
```

In this case, *age* is the name of our variable. This is how we will call it every time we need to access its value during our program. The equal is an *attribution* sign, and 18 is the value we want the variable to have. Hence, what the attribution sign does is bind the desired value to the desired identifier.

As you can notice, we did not need to specify the type (integer) of the data we are storing for the variable `age`. Python automatically knows which is the type we want to store, so you will never need to worry about it when coding with this language. In many languages it is mandatory to specify the data type when declaring new variables.

Below we have more examples of variables declaration for different data types:

- `price = 4.99`
- `letter = "a"`
- `sentence = "Three word sentence"`
- `list = [18, 4.99, "a", "Three word sentence"]`

The last variable, `list`, as discussed in Section 1, is a collection of items. In this case it is composed by four elements, an integer, a float, and two strings. Note that when declaring a list, all its elements must be inside *[brackets]* and the values separated by comma.

What we are going to do now is to code these variables and `print` them on the screen. As we already know, we will define a `main()` function, and inside it, respecting our blocks indentation, we will declare the variables and print their values.

Open your code editor and type:

```
1   def main():
2     age = 18
3     price = 4.99
4     letter = "a"
5     sentence = "Three word sentence"
6     items = [18, 4.99, "a", "Three word sentence"]
7
8     print age
9     print price
10    print letter
11    print sentence
12    print items
13
```

```
14    main()
```

After executing your code you should get this output:

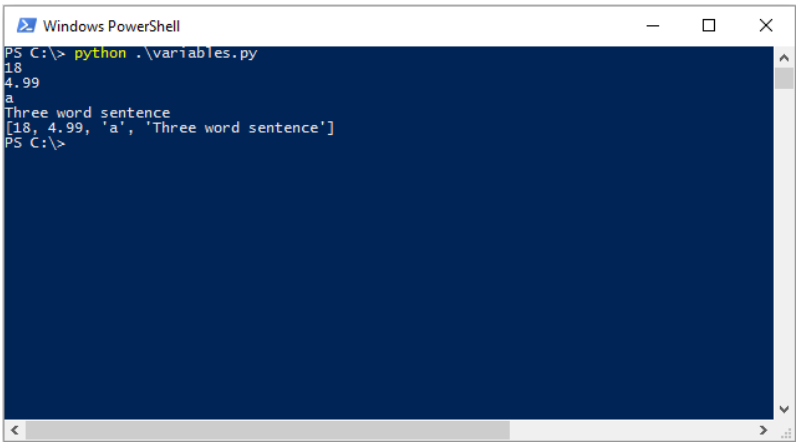

Figure 1: Variables declaration and print.

Now you know how to declare and print a variable! However, we can print the values of our variables in a different way. As you can see, we just printed out the values we had, but what about if we wanted to print a sentence together with them, like: "I am 18 years old", instead of just showing "18"? That's what we will learn now.

3 Variables formatted output

Let's build another example together. Now we will have two variables, `name` and `age`, and what we want to do is display these two values in the console output with just one sentence. Our result should be: "My name is John and I'm 20 years old". Create a new file in your code editor and type:

```
1    def main():
2        name = "John"
3        age = 20
4
5        print "My name is %s and I'm %i yo." % (name, age)
6
7    main()
```

Your output should be:

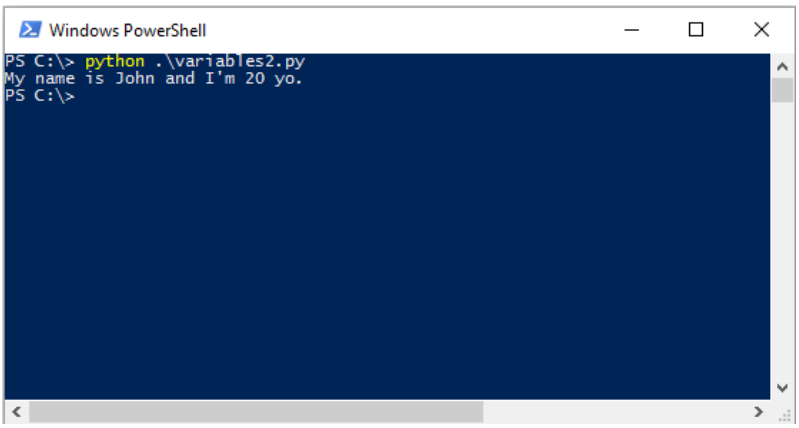

Figure 2: Variables console output.

Now let's understand what is going on with our code. We declared and applied values to two variables, `name` and `age`. We want to display these values inside a sentence within the same `print` command. Note that we have one ***%s*** and one ***%i*** inside our `print`. We do this when we want to tell Python that in those exactly spots in our sentence we want to output the values of variables. More specifically, in this case, that we want to output one string value and one integer value.

After the end of the sentence, when we close our double quotes, we have another ***%***. This indicates to Python that now we are going to tell which variables we want to be displayed. So we put inside parenthesis and separated by comma the names of the variables we want to fill the ***%s*** and ***%i*** spots. The order followed will be left to right, so `name` will fill the ***%s*** and `age` will fill the ***%i***.

If we want to display an integer we use ***%i***, and if we want to display a non integer (a *float*) we use ***%f***. If we are dealing with lists, for now, it's okay to use ***%s***. We will study some more interesting properties about lists later.

This way we can mix sentences with variables values in the console output in Python.

Now try for yourself!

Now it's your turn! Create a new file and name it ***exTutorial2.py****.*
You will declare some variables and print some messages in the console. Follow these steps:

1. Define your `main()` function, just like we have done so far in our previous examples.
2. Create seven new variables and set values to them. Your variables should be: `name`, `surname`, `age`, `day`, `month`, `year` and `hobbies`. **Attention**: `hobbies` should be a list and should contain three different values.
3. Create now four `print` commands, with the following sentences:
   - My name is `name surname`.
   - I'm `age` yo.
   - I was born on `month day`, `year`.
   - My favorite hobbies include: `hobbies`

   The names of the variables in the sentences should be replaced by the values you declared to them.
4. Call your `main()` function.

Your output should be something like this:

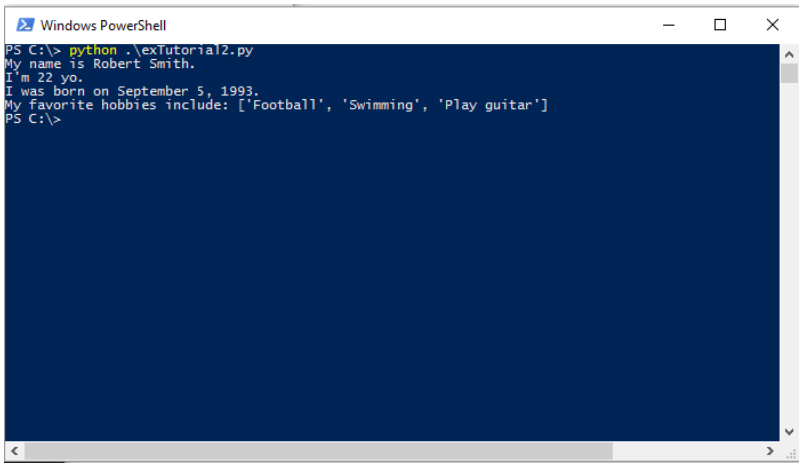

Figure 3: Exercise output example.

Good luck!