01IntroToPython

January 23, 2022

1 Culture and Coding: Python - Unit 1 Coding Fundamentals

The following 'book' and 'slides' is actually a jupyter notebook, which allows for Markdown and Python side by side. You may be reading this in the exported PDF form.

The notebook file is available on the CS0: Culture and Coding Github.

1.0.1 Python: A Modern Programming Language

- Focuses on 'scripting'
- Interpreted
  - Takes another program to run the code
- Common Uses
  - Research
  - Data science
  - Machine learning and Artificial Intelligence
  - Web development (backend)
  - Full applications and games

Becoming more common, and one of the fast growing languages.

1.1 Programs are

- A recipe with certain steps
- They are followed in order

1.1.1 A computer only does what you tell it

- It can't guess
- It can't assume
- It follows the instructions
  - for better or worse
  - Line by line, unless we say otherwise

1.2 Setting Variables

A variable * a "named" item, that stores a value * Indentifier - another term we use for the variable name * We set values to the variables using the equal(=) sign.

[ ]:

```
var = 10
puppyCounter = 100
names_have_meaning = 12.0
```

The value 10 is stored in var. At any point, I can reference `var` and 10 will be returned.

The same is true for other variables referenced *before* the print.

[ ]:

```
print(var)
print(puppyCounter)
print(names_have_meaning)
```

```
10
100
12.0
```

I can also change the value of var at any point!

[ ]:

```
var = 12
print(var)
var = 13
print(var)
```

```
12
13
```

I can perform mathmatical operations, mainly basic elementary operations, such as:

- + addition
- - subtraction
- * multiplication
- / division

[ ]:

```
new_var = 10+4
print(new_var)
```

```
14
```

1.3 Getting Input from the Client

- We want programs to be interactive
- `input` is a python command that
  - Requests the client to enter information
  - Allows a 'prompt', so user knows what that input is
  - Returns the answer, which you have to store!
- Format:
  - variable = **input**("String prompt")

[ ]:

```
input("What is your name?") # wait, what do you do with the answer?
```

[ ]:

```
''
```

[ ]:
```
name = input("What is your name?")
print("Hello, " + name)
```

```
Hello, Aurther
```

1.4 Wanting Numbers

- Input always returns a "string"
  - A collection of characters in a set order
- However, sometimes we want numbers
  - We have to be explicit! Python needs to know!
- types conversion
  - `int(value)` converts value to a **whole** number
  - `float(value)` converts value to a **floating point** (has a decimal) number
  - `str(value)` converts a number to a string.

Pro Topic: While python is 'weakly typed', type still matters!

[ ]:
```
age = input("What is your age?")
half_age = age / 2  ## this will cause an error!
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_21212/1941047333.py in <module>
      1 age = input("What is your age?")
----> 2 half_age = age / 2  ## this will cause an error!

TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

[ ]:
```
age = int(input("What is your age?"))
half_age = age / 2
print("This is your half age!", half_age)
```

```
This is your half age! 5.0
```

1.5 Your Turn!

- Go to your canvas shell, you will find "In Class Activity: Intro to Python
- You will work on the assignment in **pairs**, with **one** person coding
  - the person coding should be the one with the **least** experience
  - everyone else guide them!