Intro to Python

Variables, Data Types, Operators, and Input/Output

Getting started

Today we will be using **Google Colab!**

Click the link in Chat

Link

Then click file -> "Open in playground mode"

Google Colab

Colab runs Jupyter Notebooks, a type of file that combines text and Python code in blocks called cells. Some blocks are text just for reading, and some have code, which can be run.

- Tips for Windows:
  -  +  or  to put your windows side-by-side
  - `Alt+Tab` to switch between windows
  - `Ctrl+Tab` and `Ctrl+Shift+Tab` to go back and forth between tabs

Hello World

Keeping with tradition we'll start by printing "Hello, World!" to the console.

```
print("Hello, World!")
```

- It will take a bit longer to run this first code block
- Run it by clicking the little play button: 
- or typing `Shift + Enter`
- Try changing the words between the double-quotes and run it again!

Variables

We can tell the computer to "remember" something using variables.

Run the cell below:

```
name = "Alice"
age = 30
print("Hello,", name)
print("You are", age, "years old.")
```

Naming Variables

In lines 1 and 2 we are "assigning" a value to variables. You can name a variable whatever you want, but it should describe what it stores, just like labeling moving boxes!

Can you pick out the least helpful name?

a. Clothes
b. Books
c. Stuff
d. Pillows

Another way to print variables

If you put a variable on the last line of a code cell, it will print the value of that variable.

For example

```
name
```

More on Variables

```
age = 30
```

- The `=` in the above statement is called the assignment operator.
- Unlike in Math, the order matters and the variable always goes on the left; for example, this will give you a syntax error:

```
30 = name
```

Variables: Try it some more

In the empty cell below, create a *new* variable and assign a value to it, then print the value of your variable. For example:

```
today = "Monday"
print(today) # this would print 'Monday'
```

```
# Put your code below

```

More on Variable Names in Python

- Variable names must begin with a letter or an underscore (`_`).
- There are some *conventions* you should follow:
    - If your name has a space in it replace the space with an underscore. This is called "snake case"!
        - Example: `my_name`
    - Only use lowercase letters for variable names.
        - Uppercase names are reserved for classes, which we'll get to later.
        - All capital letters are for constants, variables that should not change value.

Data Types

Python has strings, integers, floats, and booleans.

To find the type of something, you can use the `type()` function

```
x = 10   # integer
y = 3.14  # float
z = "Python" # string
a = True  # boolean

type(a)
```

Data Types: Try It

Check the type of the variable you created previously, using `type()`. Alternatively, you can make a new variable if you like.

Example

```
dog = "Fido"
type(dog)
```

(This example would print <class 'str'> and 'str' is short for string)

```
# Put your code below

```

Operators

We can do arithmetic with these operators:

- Addition: `+`
- Subtraction: `-`
- Multiplication: `*`
- Division: `/`

Run this code block:

```
result = 10 + 5
result
```

Operators: Try it

- In the cell below, do the following:
  - store the result of 5 + 3 in a variable called `sum`
  - store the result of 5 - 3 in a variable called `difference`
  - store the result of 7 x 2 in a variable called `product`
  - store the result of 6 / 2 in a variable called `quotient`

```
# Put your code below

```

Input and Output

We can interact with the user by taking input and displaying output.

```
name = input("What is your name? ")
print("Hello,", name)
```

Input and Output: Try it

Ask the user for their name and then print a greeting using their name.

```
# Put your code below
```

That's it for now

Remember: the best way to learn Python is by playing with it!

Practice and experiment whenever you can!