Learning Objectives

- **Run a python file**
- **Create a variable**
- **Store numerical data in a variable**
- **Commenting code**
- **Print the data stored in variables**

definition

Limitations

- This section will primarily cover variables. Data frames and functions will be covered in a later section.

Printing and Comments

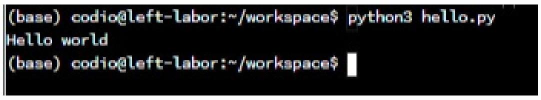
Printing to the console

As you work with Python, you will write code in the code editor to the left. Use a `print` command to see the output of your program. Enter the code below (or copy/paste the code) and click the `TRY IT` button.

```
print("Hello world")
```

The terminal at the bottom left can also be used to run your code. To run your code, you need to know the file name and then use the command `python3 filename.py`.

Try running the hello.py file from the terminal.



```
(base) codio@left-labor:~/workspace$ python3 hello.py
Hello world
(base) codio@left-labor:~/workspace$
```

terminal

The reason you were able to see the words appear is because of the **print** command. Change your code to look like this

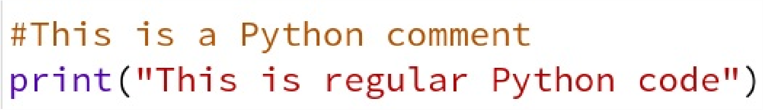
```
"Hello world"
```

and run it again. Make sure you run it after removing the print statement.

Without the print command, nothing will print out in our console. We use the `print` command to help better visualize information that we might need to retrieve from our data.

Comments

You may have wondered why a couple of lines of code appear in a different color (in the below example, light brown, but it depends on the Theme you have picked):



```
#This is a Python comment
print("This is regular Python code")
```

images/comments

In Python, to write notes in code without impacting its function we can use # to make a **comment**.

```
# This is a python comment
```

Comments can also be used to help you fix your code. You can “comment out” lines of code that are not working or you suspect are causing problems.

challenge

What happens if you:

- First print `print("Red")`
- Change `print("Red")` to `prnt("Red")`?
- Comment out the line of code with the typo?

Comment Blocks

To make a multi-line comment you can either combine the single line characters # or wrap the set of lines in triple quotes (`'''`).

```
'''
This is a multi-line comment
You can then easily end the comment with a triple quote (see below)
'''


#You can also do a multi-line comment
#like this!

print("Notice code that runs is not the same color as single-line comments");
print("This feature is called syntax highlighting");
print("It is a common feature of IDEs");
```

The syntax highlighting is different for comments with #and comments with `'''`. That is because the triple quotation marks are also used for multi-line strings (see the Strings lesson). When a multi-line string is by itself (no `print` statement), then Python treats it as multi-line comment.

Variables

Variable Names

Variables are used to store a value, and these values have a data type. Data types describe the kind of information that is being stored. Numbers are different than text, and integers differ from numbers with decimals. A variable declaration is when you create a variable and assign it a value. Enter the name of the variable you want to create, an `=` (called the assignment operator), and the value you want to store in the variable. In Python you do not have to indicate the data type when declaring a variable. Use the `print` statement to see the value of the variable.

```
my_variable = "Hello world"
print(my_variable)
```

Do not use quotation marks when printing a variable. Using quotation marks will print the variable name, not its value.

```
my_variable = "Hello world"
print(my_variable)
print("my_variable")
```

Variable Naming Rules

Here are the rules for declaring a variable.

Rule	Correct	Incorrect
Start with a letter or an underscore	`variable`, `_variable`	`1variable`
Remainder of variable name is letters, numbers, or underscores	`var_i_able`, `var1able`	`var-i-able`, `var!able`
Cannot use a Python	`my_class`	`class`

keyword	
Variables are case sensitive	`variable`, `Variable`, and `VARIABLE` are all different variables

▼ What are some of the Python keywords?

and	as	assert	break
class	continue	def	del
elif	else	except	FALSE
finally	for	from	global
if	import	in	is
lamda	None	nonlocal	nont
or	pass	raise	return
TRUE	try	while	with
yield			