Python 1 Notes

What does a computer do?

- Performs calculations (a billion calculations per second)
- Remembers results (100s of gigabytes of storage)

What kind of calculations?

- built-in to the language
- the ones defined by you as a programmer

Simple calculations like...

- searching the World Wide Web
- playing chess

Complex calculations like...

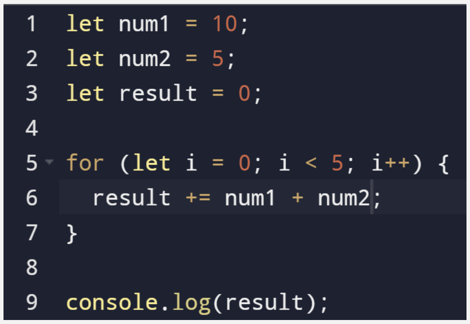
- Weather predictions
- cracking encryption schemes

What is Python?

- Code resembles the English language which
- computers understand python through a Python interpreter
- Strong emphasis on code readability and simplicity (this allows programmers to develop applications rapidly)

Why Python?

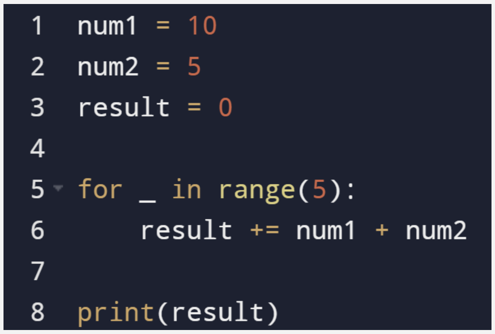
1. Easy to read
2. All high level languages (C, C++ and Java) are very similar to each other
3. Requires fewer lines of code to perform same task compared to languages like Javascript
4. Reduces development time



```
let num1 = 10;
let num2 = 5;
let result = 0;

for (let i = 0; i < 5; i++) {
  result += num1 + num2;
}

console.log(result);
```



```
num1 = 10
num2 = 5
result = 0

for _ in range(5):
    result += num1 + num2

print(result)
```

5. Leading to fewer programming errors
6. Potential to create a large variety of tasks like -> desktop apps, database apps, network programming, game programming, and even mobile development
7. Cross platform language (meaning: code written for Windows will work well on Mac OS or Linux without having to make changes to the Python code)
8. popular language thatruns
9. on almost every machine and is used at many big, important organizations like Google, Instagram, NASA, and Spotify
10. Convinced? Now let's get started!

Python Terms

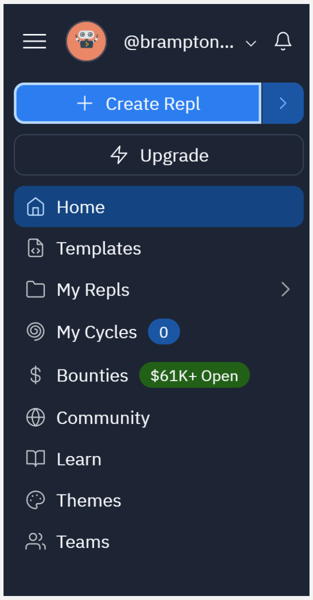
- A program is a sequence of definitions and commands
    - Definitions are evaluated
    - Commands are executed by Python interpreter in a shell
- Typed in a shell or stored in a file that the shell reads and evaluates
- Shell executes your Python programs, other pieces of Python code, or simple commands

Let's set up your account!

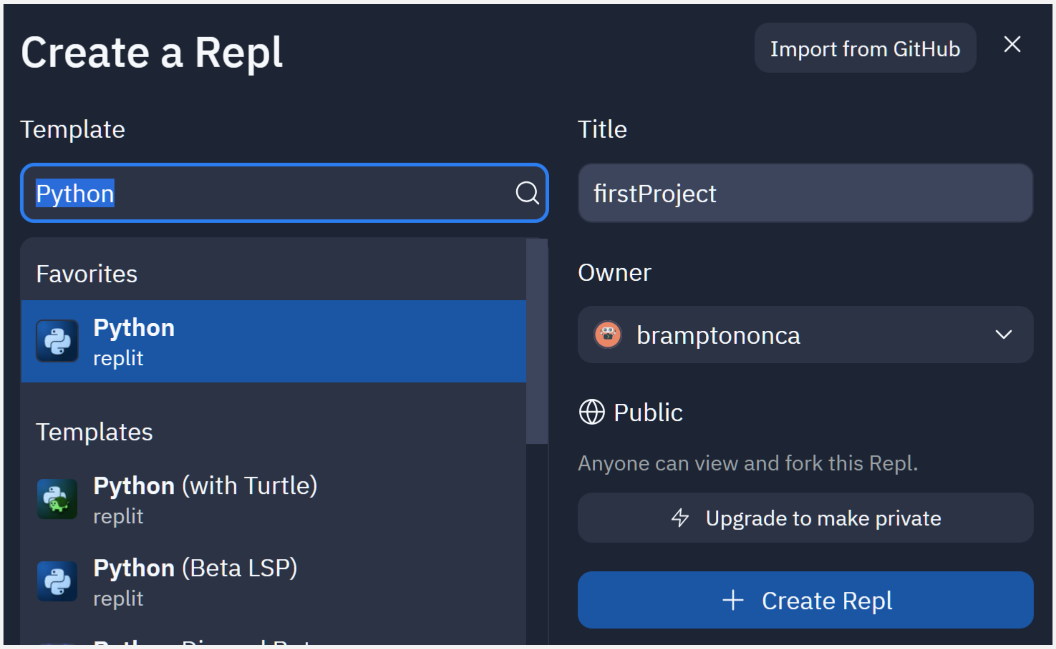
- *REPLIT STEP BY STEP INSTRUCTIONS*

Let's write our first python command!

1. Click on "Create Repl"



2. **Type "Python" for template, title your project and hit ENTER!**


Create a Repl

Template: Python

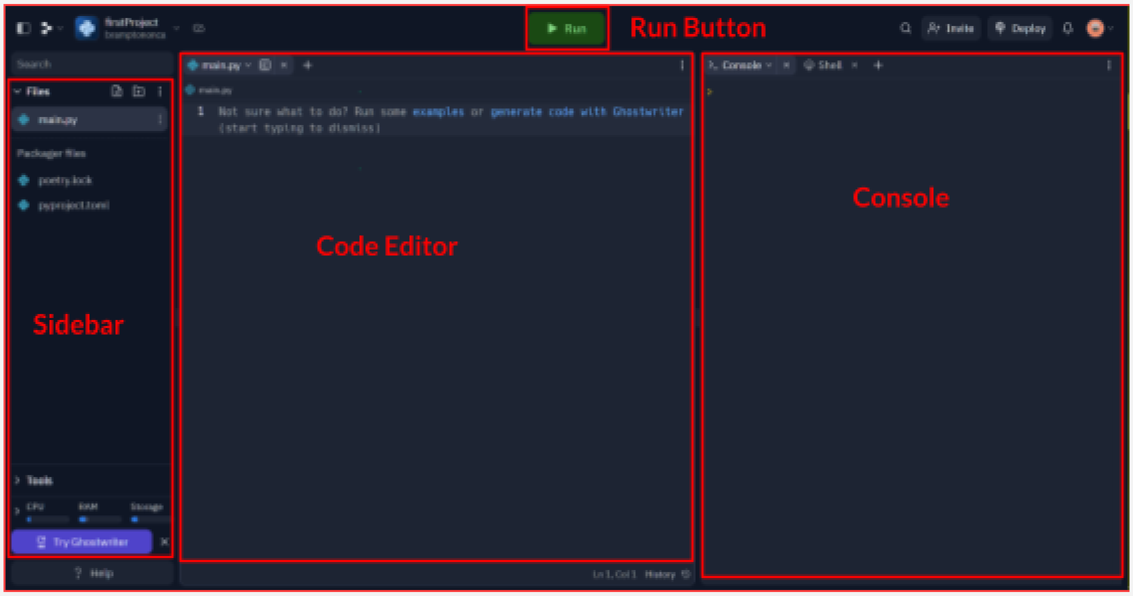
Title: firstProject

Owner: bramptononca

Public

Anyone can view and fork this Repl.

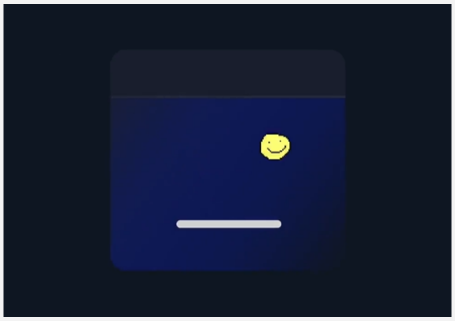
3. **You should see something like this:**



```
1  Not sure what to do? Run some examples or generate code with Ghostwriter
   (start typing to dismiss)
```

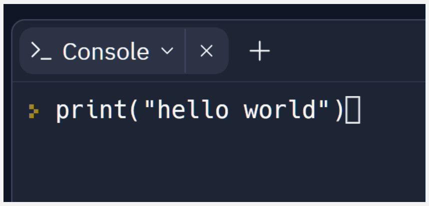
   a. Code Editor -> 'main.py' is your file name where you will be writing code. It highlights key python terms!
   b. Run Button -> executes your program and outputs the result
   c. Console -> this is where your output appears

```
hello world!
```



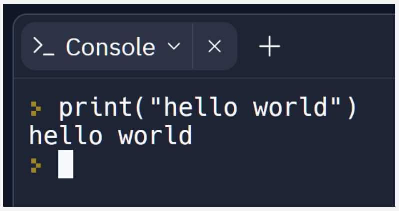
   d. Sidebar -> all your files and tools appear here

4. Replit autosaves your code so no need to worry about loosing your precious work!
5. We will familiarize ourselves with the console first!
   a. Type the following code in your console and hit ENTER



```
› print("hello world")
```

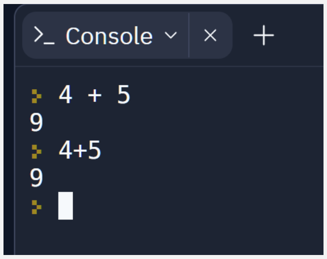
b. Do you see something like this? Well congratulation! You've just written your first line of Pythpn code!



```
› print("hello world")
hello world
›
```

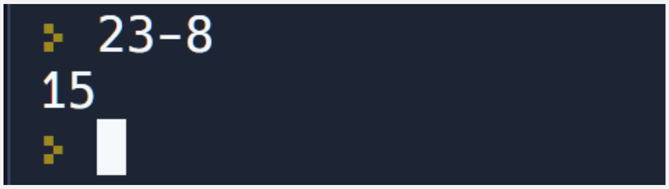
Let's try to do some simple math on our console!

- Let's add 4 + 5 in our console (spaces don't matter):



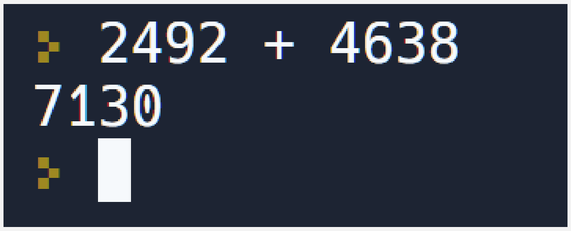
```
› 4 + 5
9
› 4+5
9
›
```

- Shout out the answer!!
- What's 23 - 8?



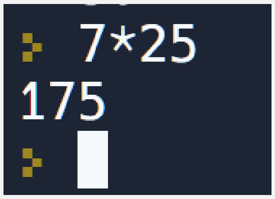
```
› 23-8
15
›
```

- What's 2492 + 4638?



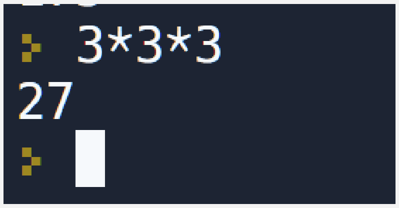
```
› 2492 + 4638
7130
›
```

- What's 7 times 25?



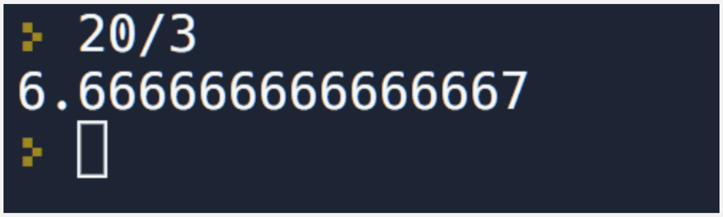
```
› 7*25
175
›
```

- What's 3*3*3?



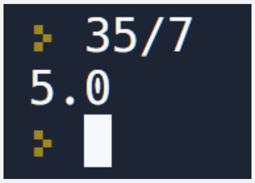
```
› 3*3*3
27
›
```

- What's 20/3?



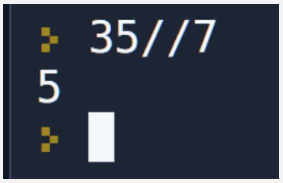
```
› 20/3
6.666666666666667
›
```

- What's 35/7?



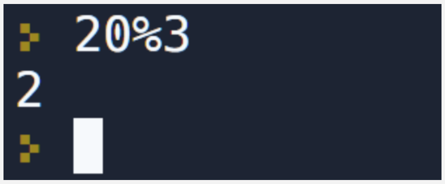
```
› 35/7
5.0
›
```

- What if we don't want the decimal 0? What do we do then?
    - We do 35//7
    - This gives us only the whole number value



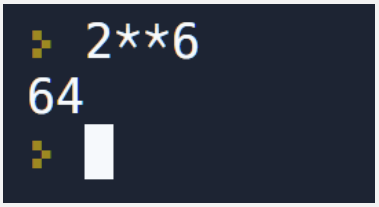
```
› 35//7
5
›
```

- How can you calculate the remainder of 20/3?
    - 20%3



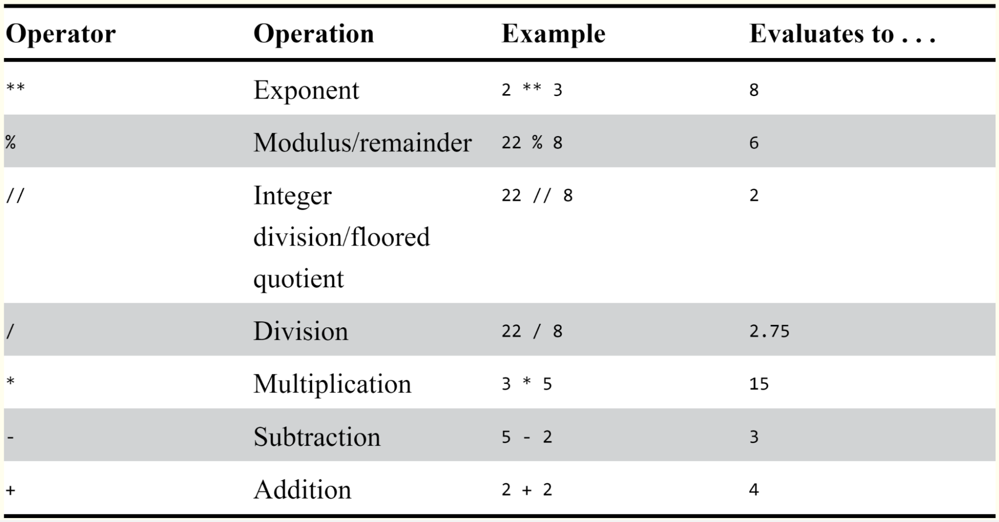
```
› 20%3
2
›
```

- Find 2 to the power of 6?
    - 2**6



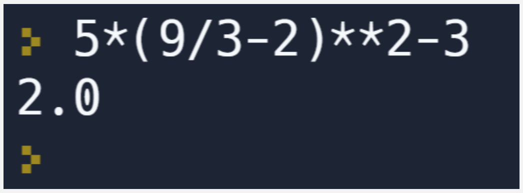
```
› 2**6
64
›
```

- Math Operators Summary table (retrieved from https://automatetheboringstuff.com/2e/chapter1/#:~:text=Table%201%2D1%3A%20Math%20Operators%20from%20Highest%20to%20Lowest%20Precedence)



Operator	Operation	Example	Evaluates to . . .
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4

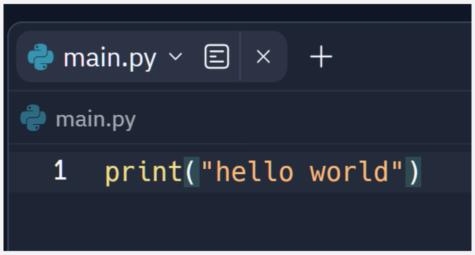
- Let's try something hard:
    - Who can tell me what this would equal to:
    - 5*(9/3-2)**2-3 -> 2.0



```
5*(9/3-2)**2-3
2.0
```

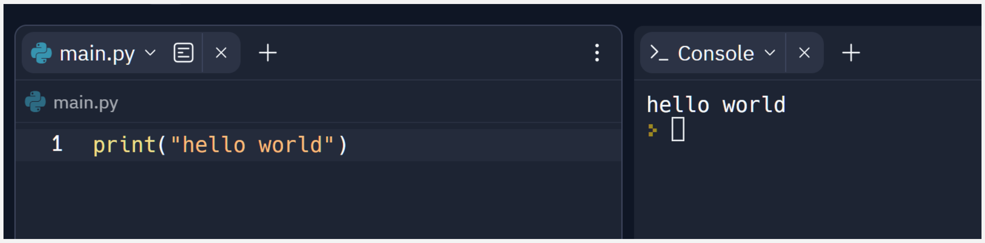
What if I want the code to run from a file?

1. Type the following code in your Code editor:



```python
print("hello world")
```

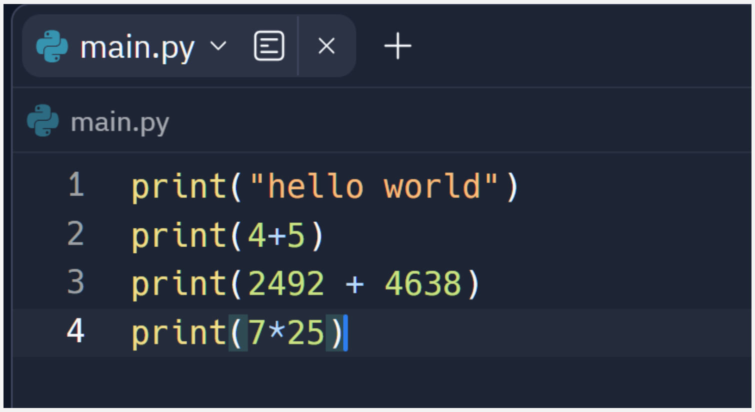
2. Hit the run button
3. You should see the output appear your console. How cool?!



```python
print("hello world")
```

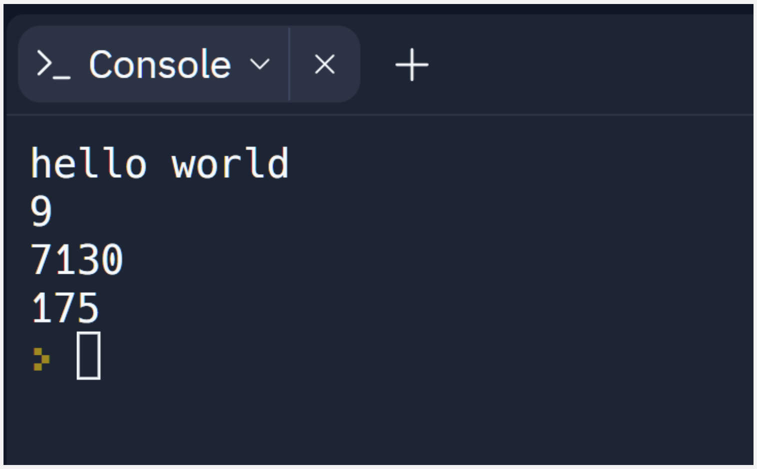
```
hello world
```

4. We can also write the math operations in our file!

a. 

```
print("hello world")
print(4+5)
print(2492 + 4638)
print(7*25)
```

5. When you RUN this script of code, it will output the results onto the console. Something like this…

a. 

```
hello world
9
7130
175
```

Let's analyze the few lines of code we just wrote

1. print("hello world")
    a. The print() function is used when we want to output a string. A string is a collection of letters, numbers and special characters. Strings are one of the data types that we will learn soon!
    b. print() takes a parameter which is a piece of information (aka input) you give a function to do something with.
    c. When we print "Hello world" (aka print a string), it must be in double quotations.
    d. When we print 4+5 or 7*25, we don't need quotation marks because the
    e. Can someone tell me what the print() function does?
    f. Can someone tell me why printing something to console can be useful?
        i. Answer: useful for debugging our code, which is the quest we undergo to find issues or mistakes in our code that cause it to not work as

expected. These issues or mistakes in our code are called bugs. When we are in the middle of debugging, we can print out parts of our code to double-check that it is doing what we expect it to do. (Debugging proper explanation: as it allows you to display the value of variables or other outputs at specific points in your code. This can help you identify where errors are occurring and fix them more easily.)

ii. the print() function can be used to display greetings to the user which are formatted messages, making it an essential tool for creating user-friendly programs

iii. We can also print results to some basic math

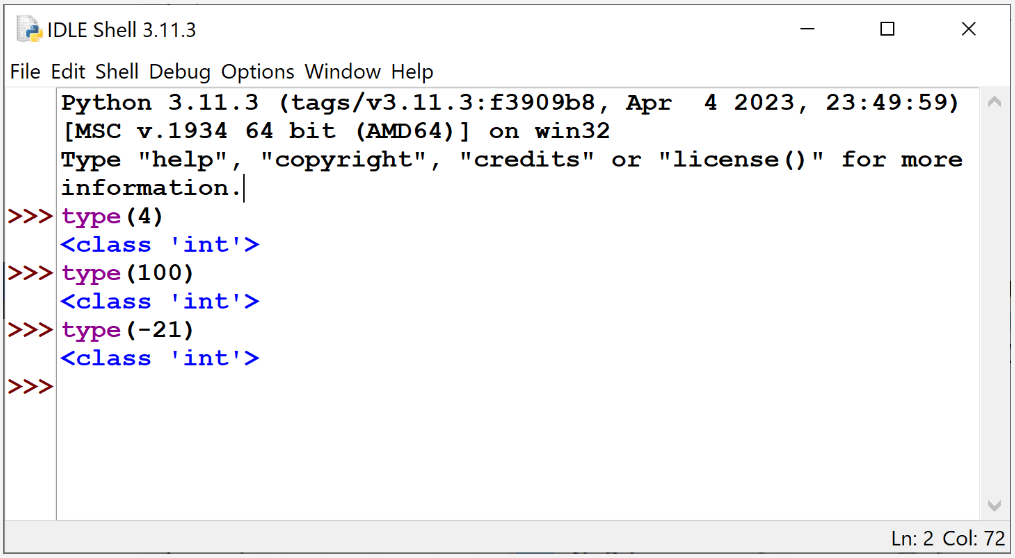
2. print(4+5)
    a.

Comments

- What do we do if we want to have a code in our program but we dont wat it to print, what do we do then?
    - Answer: We will comment it!
- Comments are pieces of code that do not gettranslated by the computer.
- They can be used as helpful messages you leave for yourself within the code as part of your code that you want the computer to ignore.
- Comments are created using the hashtag (#) before the line you'd like the computer to ignore.
-

Data Types

- Programs manipulate data objects
- Types of data objects:
    1. int -> represent integers   ex: 4, 21, 100



```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59)
[MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> type(4)
<class 'int'>
>>> type(100)
<class 'int'>
>>> type(-21)
<class 'int'>
>>>
```

2. float -> represent decimal numbers ex: 3.14, 0.19
3. bool -> represent True and False   ex: True and False
4. NoneType -> special type! Has only one value: None