

Python Interpreter

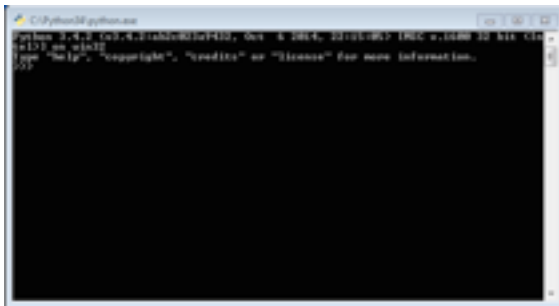
Purpose:

We don't speak *machine language*. But **Python** can! We need a way to talk to the computer in a way that we can easily understand. Python takes the commands we want to run and tells the computer what we want to do. This process is called *interpreting* and the **Python Interpreter** is what does this for us.

Goals:

- ★ Discover the purpose of an interpreter.
- ★ Learn the different modes of the Python interpreter
- ★ Open the Python Interpreter
- ★ Discover how to interact with the interpreter to get results!

Two ways to access it.



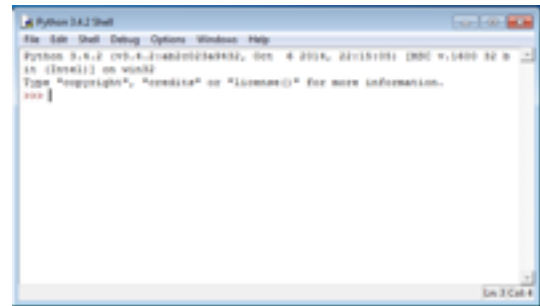
Command Line

PROS:

Full access to Python.
Quickly test out code.

CONS:

No extra features.
For more advanced users.



IDLE GUI

PROS:

All the good from the command line.
Syntax highlighting.
Script editing.
And More!

Let's use the IDLE GUI

IDLE is just the name Python gave to its built in editor. In fact it is much more than just an editor but we'll cover that later.

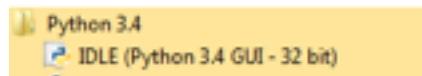
GUI stands for **G**raphical **U**ser **I**nterface. Basically it means that it interacts with you in a visual way as opposed to the plain boring text of the command line.

Flip Over to get started!

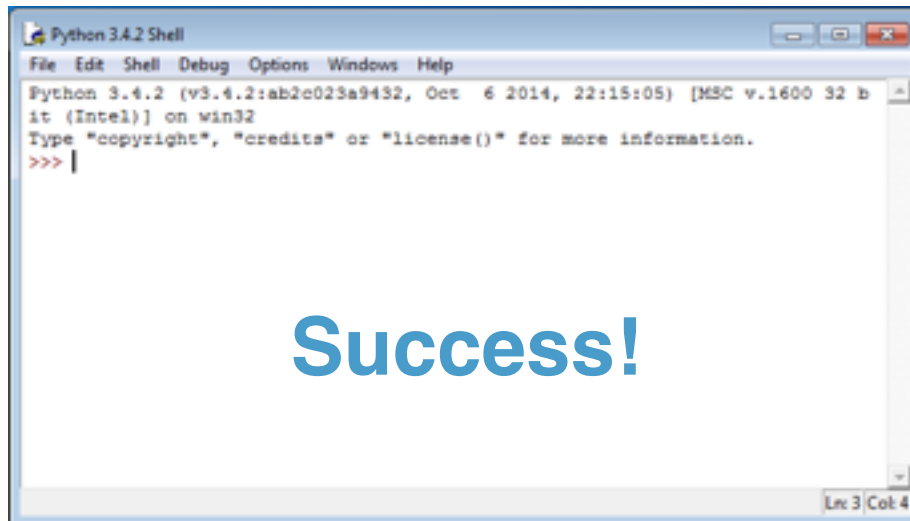
Start the IDLE GUI



All Programs



To start the IDLE GUI click on the Start button then the All Programs menu item. Look for the Python 3 menu and click that. Don't worry if yours has 3.4 or 3.5 it is all still **Python 3**. Then click the menu item that say **IDLE** (Python 3 **GUI**).



Now what?

You interact with the Python Interpreter by typing your commands in the interpreter window at the prompt which looks like `>>>`. After you type in your command and press the Enter or Return key, Python will interpret your command and display the results. If everything goes OK then you will see the result in **blue**. If there is a problem you will see the error messages in **red**.

Try typing these:

```
1 + 1
3 * 2
6 / 2
5 / 2
5 % 2
"hi"
print("hi")
```

Observations:

What do you think the `/` character does?
Does `5 / 2` return the expected answer? What is missing?
What do you think the `%` character does?
Was there a difference in the output between `"hi"` and `print("hi")`?
What do you think that difference might indicate?

Did you make any **errors**? **GREAT!** Were you able to fix them based on what the interpreter told you was wrong? **EVEN BETTER!!**

Don't be afraid of **errors and mistakes**. You will learn more from making mistakes in your programs than from anything else.