

Introduction

- Installation
 - IDLE
- Python Shell
 - REPL
- Code Editors
 - Interpreters vs compilers
- Turtle Graphics

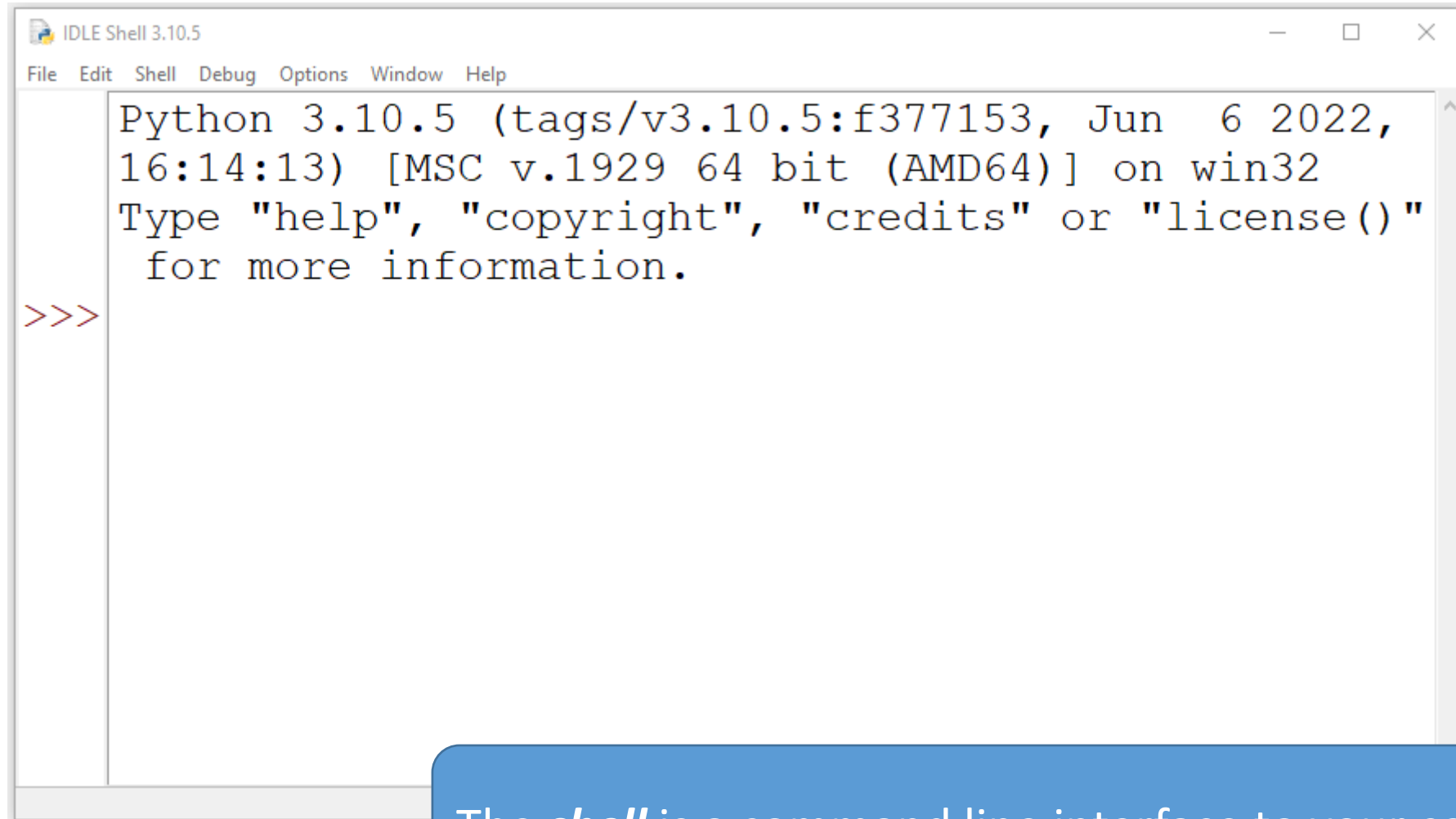
Installations

Installations

- Go to
 - <https://www.python.org/>
- Click Download
 - Execute the downloaded setup file



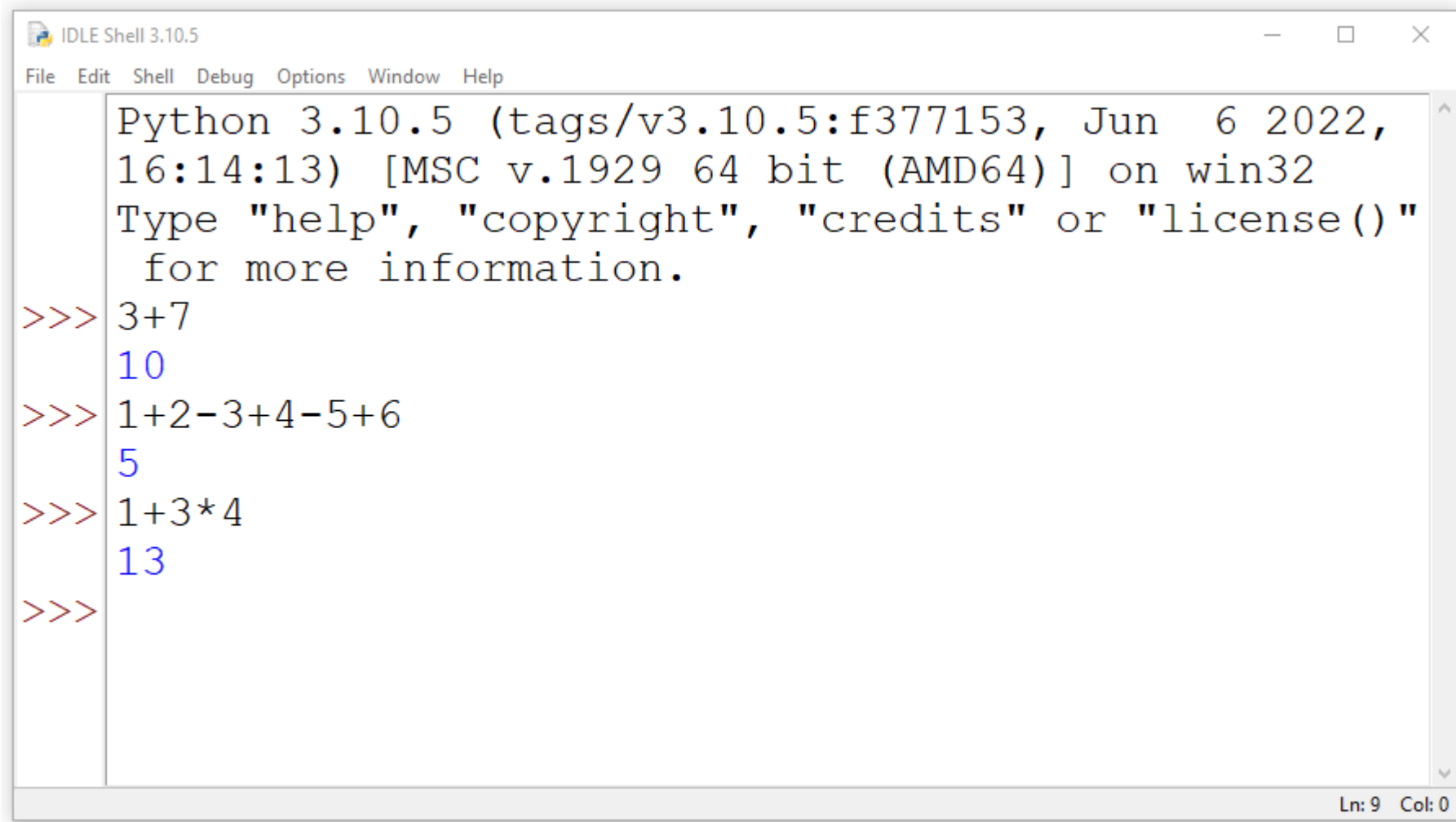
Start the Python IDLE Python **Shell**

A screenshot of the Python IDLE Shell 3.10.5 window. The window has a title bar with the text "IDLE Shell 3.10.5" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a text editor with a white background. It contains the following text: "Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", and a red prompt ">>>" on the left side of the text area.

```
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022,
16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>>
```

The shell is a command line interface to your computer.

Use the Shell as a Calculator

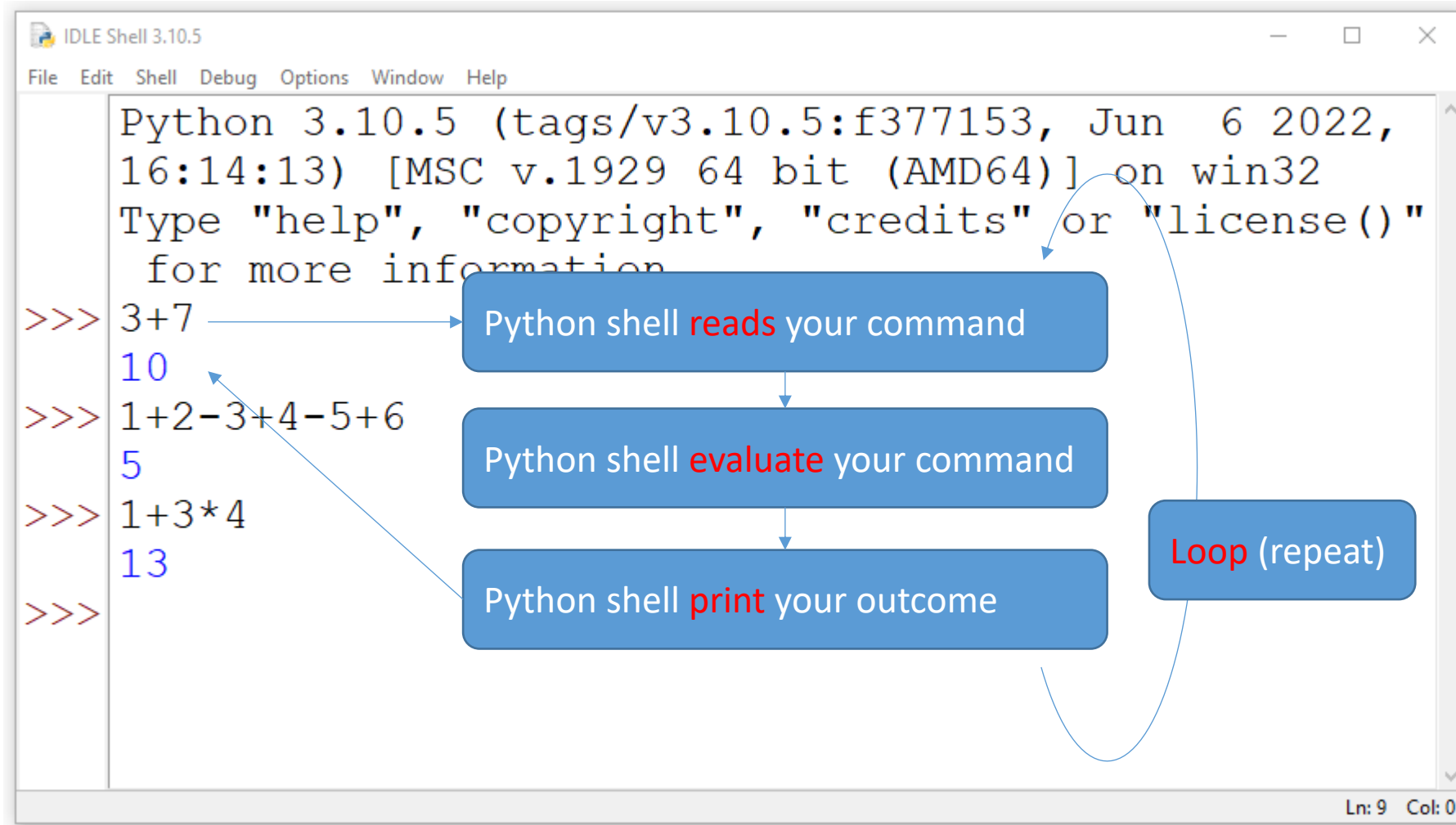


The screenshot shows the IDLE Shell 3.10.5 window. The title bar reads "IDLE Shell 3.10.5". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022,
16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>> 3+7
10
>>> 1+2-3+4-5+6
5
>>> 1+3*4
13
>>>
```

The status bar at the bottom right indicates "Ln: 9 Col: 0".

Read-eval-print loop (REPL)



Try to solve the following problem by Python

- You put \$10000 into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?

Try to solve the following problem by Python

- You put **\$15000** into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?

Try to solve the following problem by Python

- You put \$23000 into your fixed deposit bank account and the annual interest is 4.5%
 - How much do you have after one year?

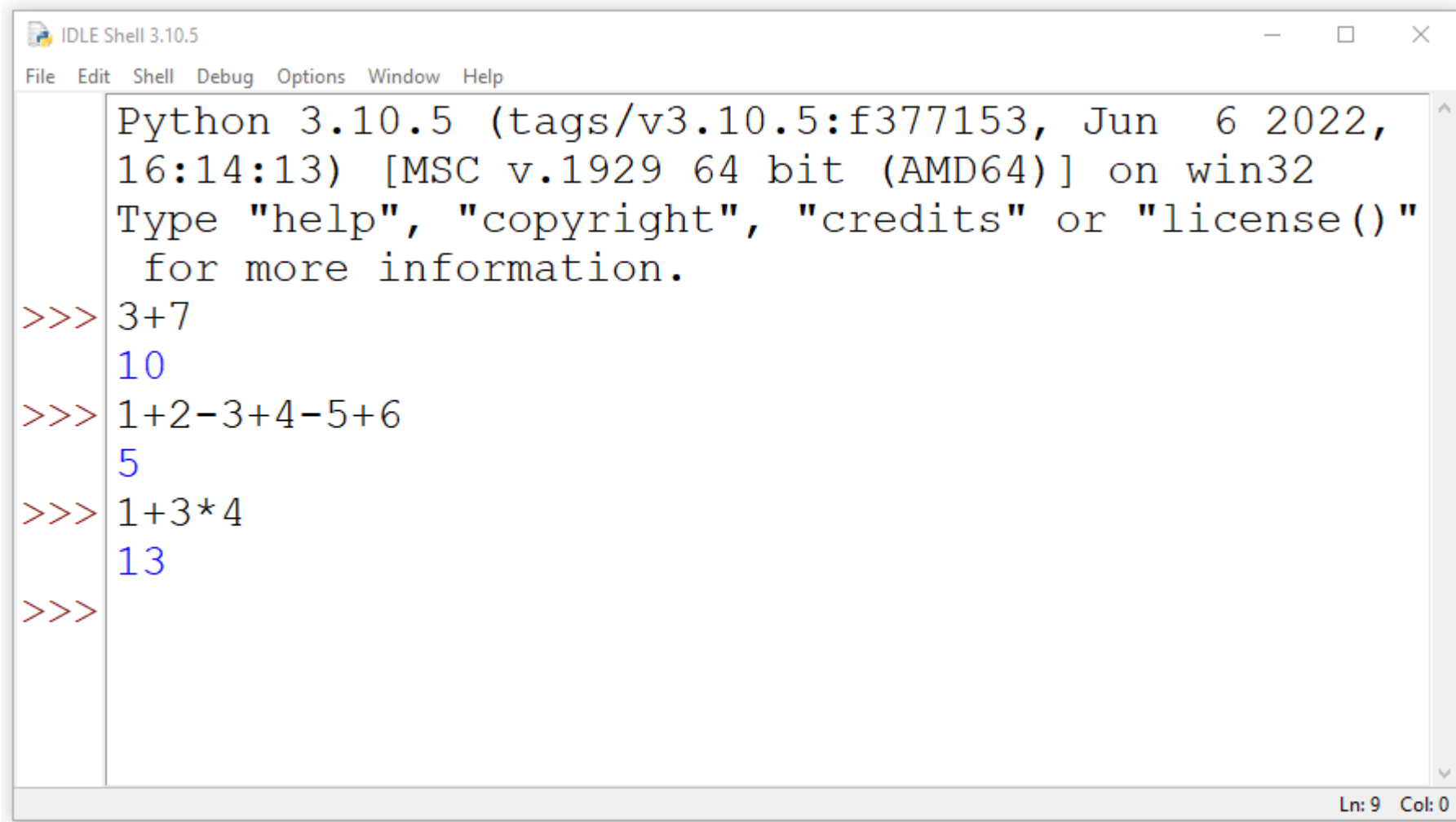
Try to solve the following problem by Python

- You put \$10000 into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?
- You put **\$15000** into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?
- You put **\$23000** into your fixed deposit bank account and the annual interest is **4.5%**
 - How much do you have after one year?

Did you type in the formula 3 times?
What if you need to do it 10 times? 100
times?

Code Editor

Using the Shell, you cannot “save” your work

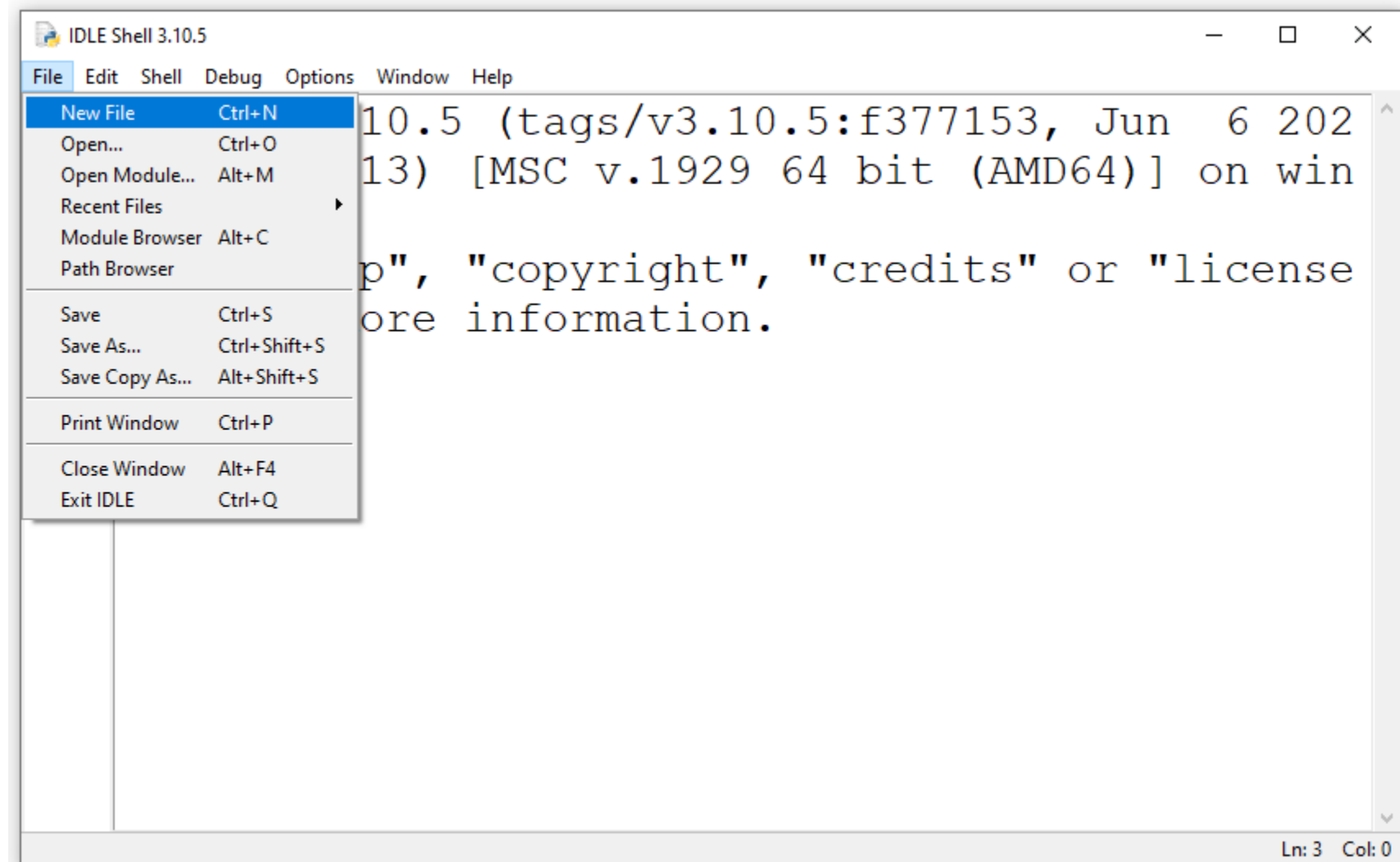


The screenshot shows the IDLE Shell 3.10.5 window. The title bar reads "IDLE Shell 3.10.5". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

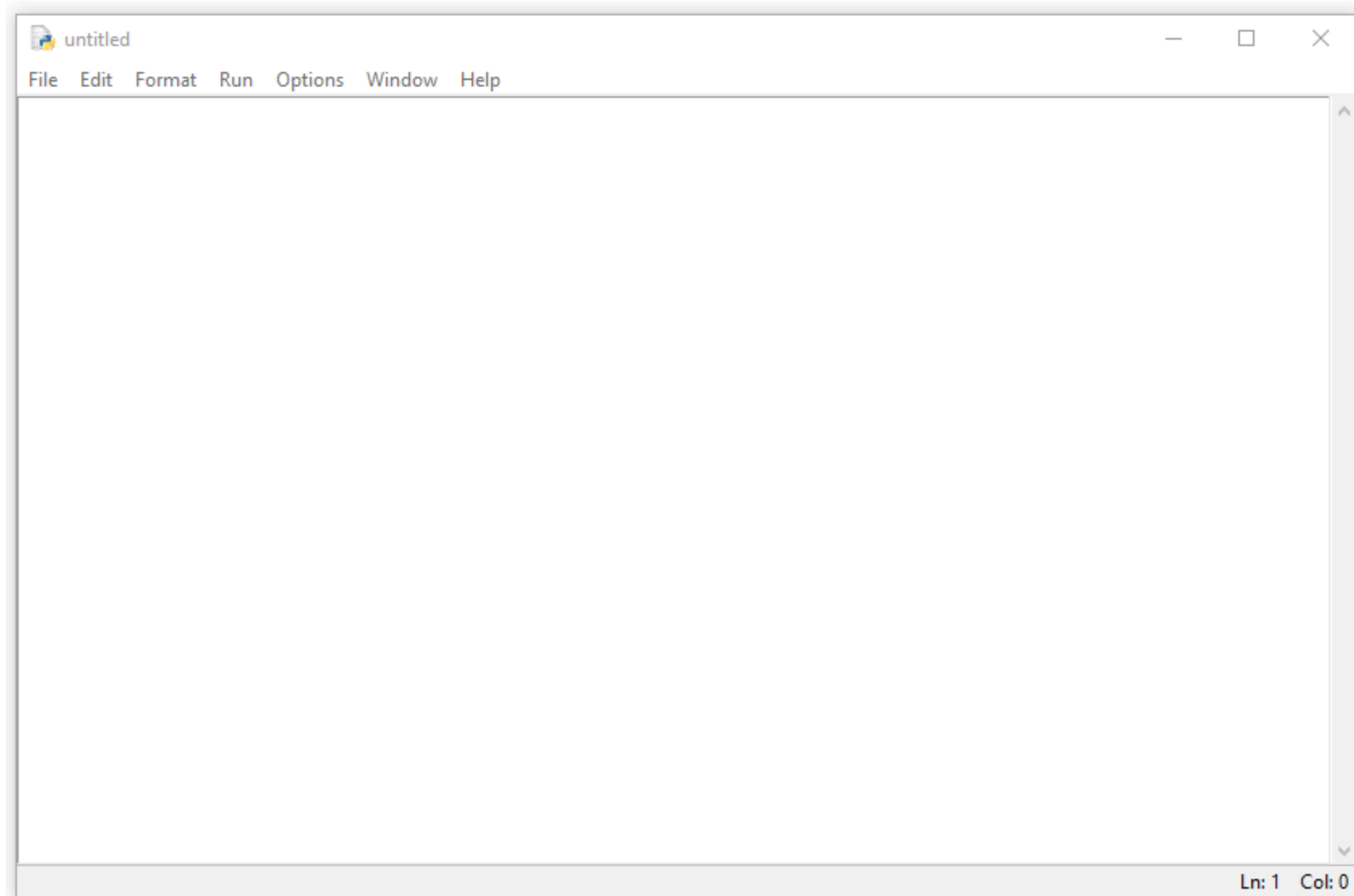
```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022,  
16:14:13) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()"   
for more information.  
>>> 3+7  
10  
>>> 1+2-3+4-5+6  
5  
>>> 1+3*4  
13  
>>>
```

The status bar at the bottom right indicates "Ln: 9 Col: 0".

What you should do: Create a program file



After Created a “New” file

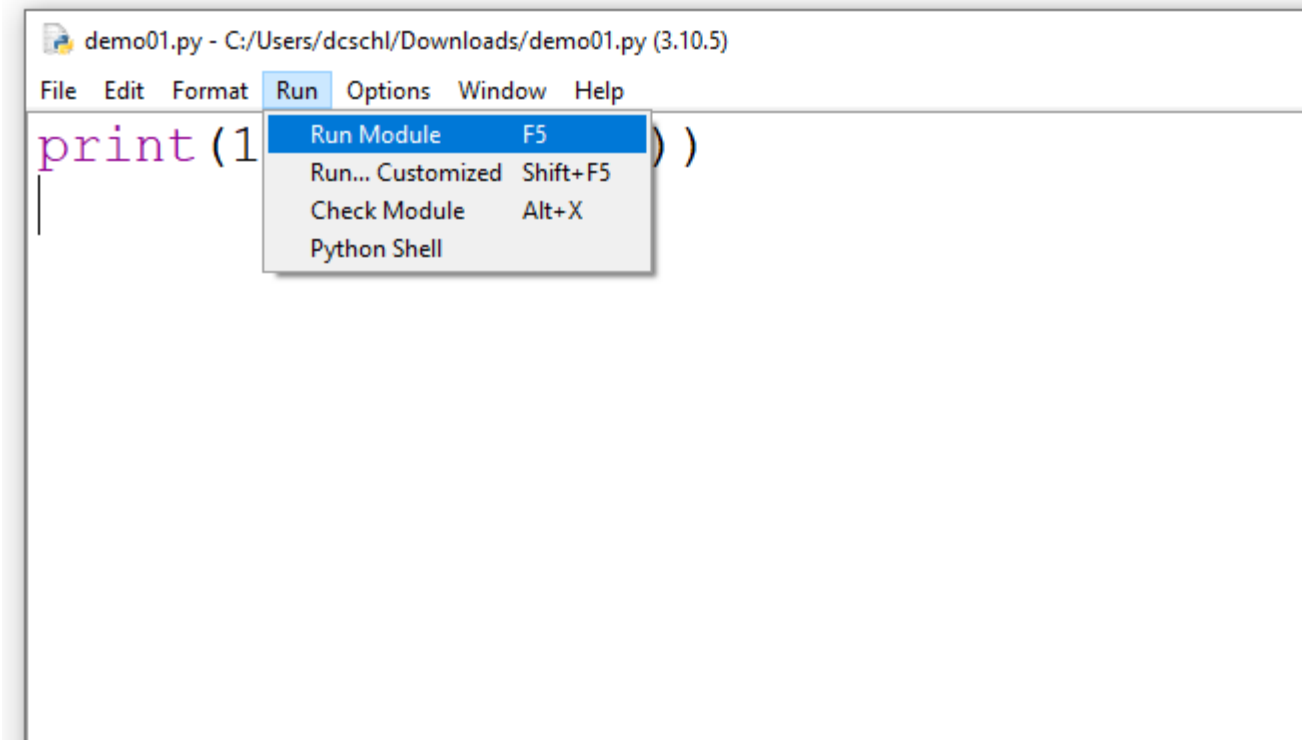


Try

- type:

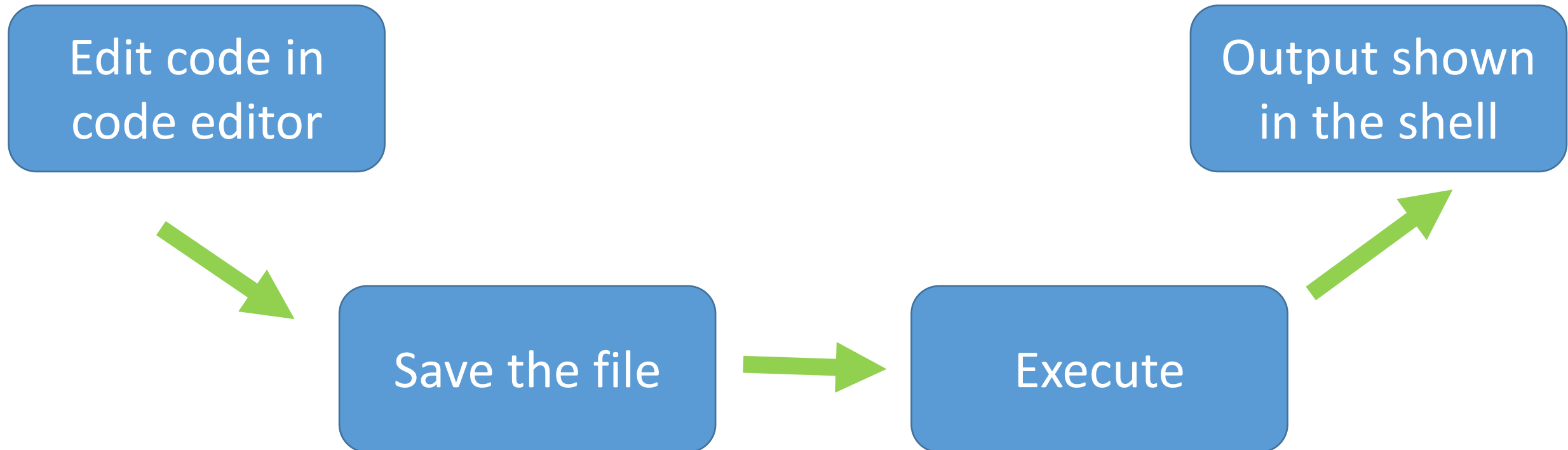
```
print(10000*(1+0.04))
```

- Save the file
- Anything happen?
- You need to “execute” the code
 - Or just press F5 in Windows

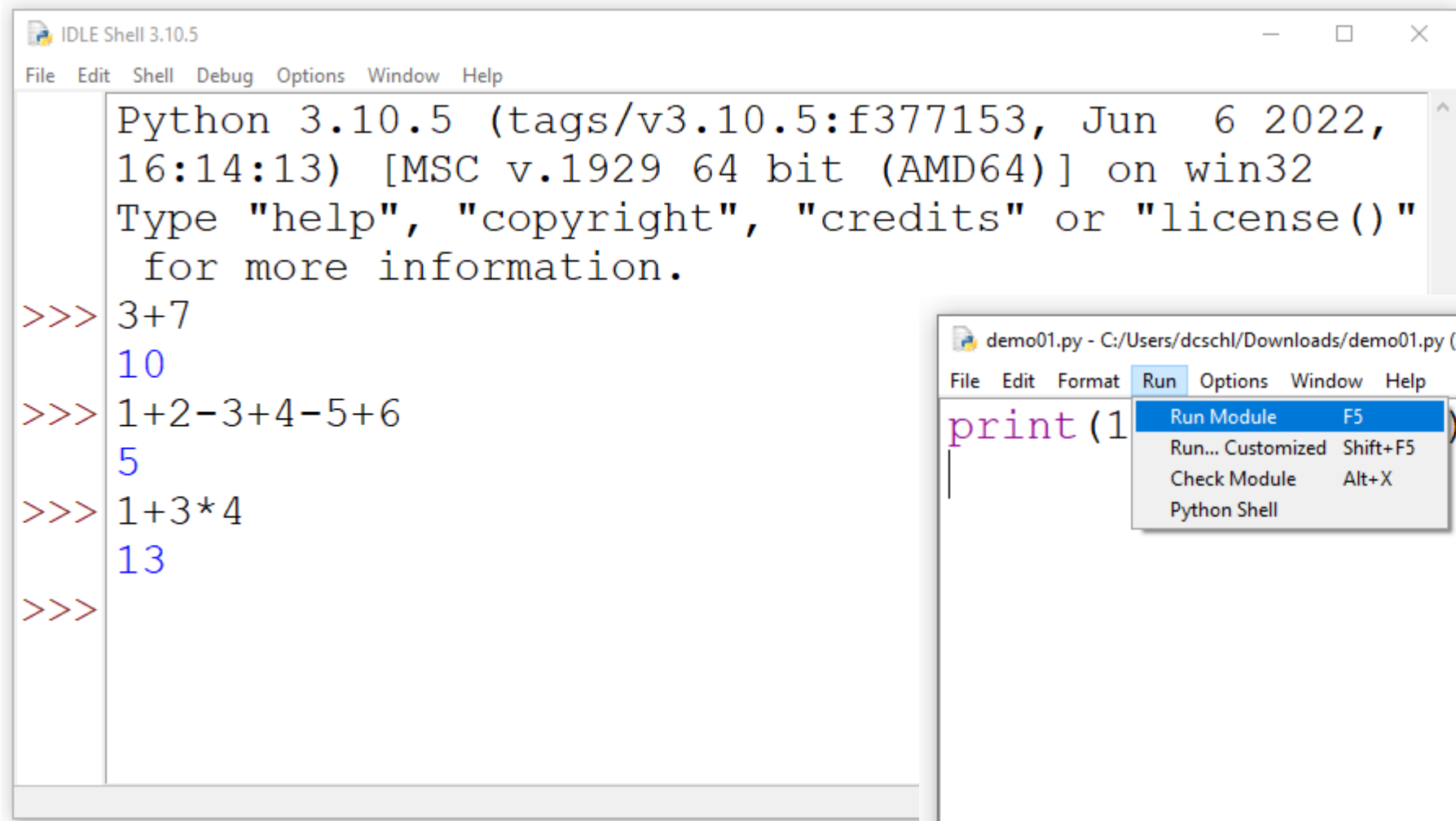


Execution of code

- The execution of code refers to the process of a *computer program* being run or carried out by a computer's central processing unit (CPU).

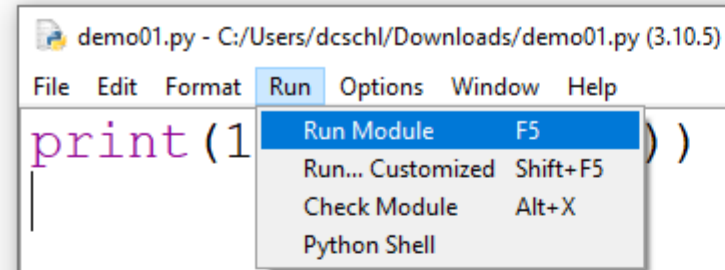


Shell vs Code Editor



The screenshot shows the IDLE Shell 3.10.5 window. The title bar reads "IDLE Shell 3.10.5". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the Python 3.10.5 startup message: "Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32". It then prompts the user to type "help", "copyright", "credits", or "license()" for more information. Below this, three arithmetic expressions are entered at the prompt: `>>> 3+7` resulting in `10`, `>>> 1+2-3+4-5+6` resulting in `5`, and `>>> 1+3*4` resulting in `13`. A fourth prompt `>>>` is shown without an input or output.

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 3+7
10
>>> 1+2-3+4-5+6
5
>>> 1+3*4
13
>>>
```

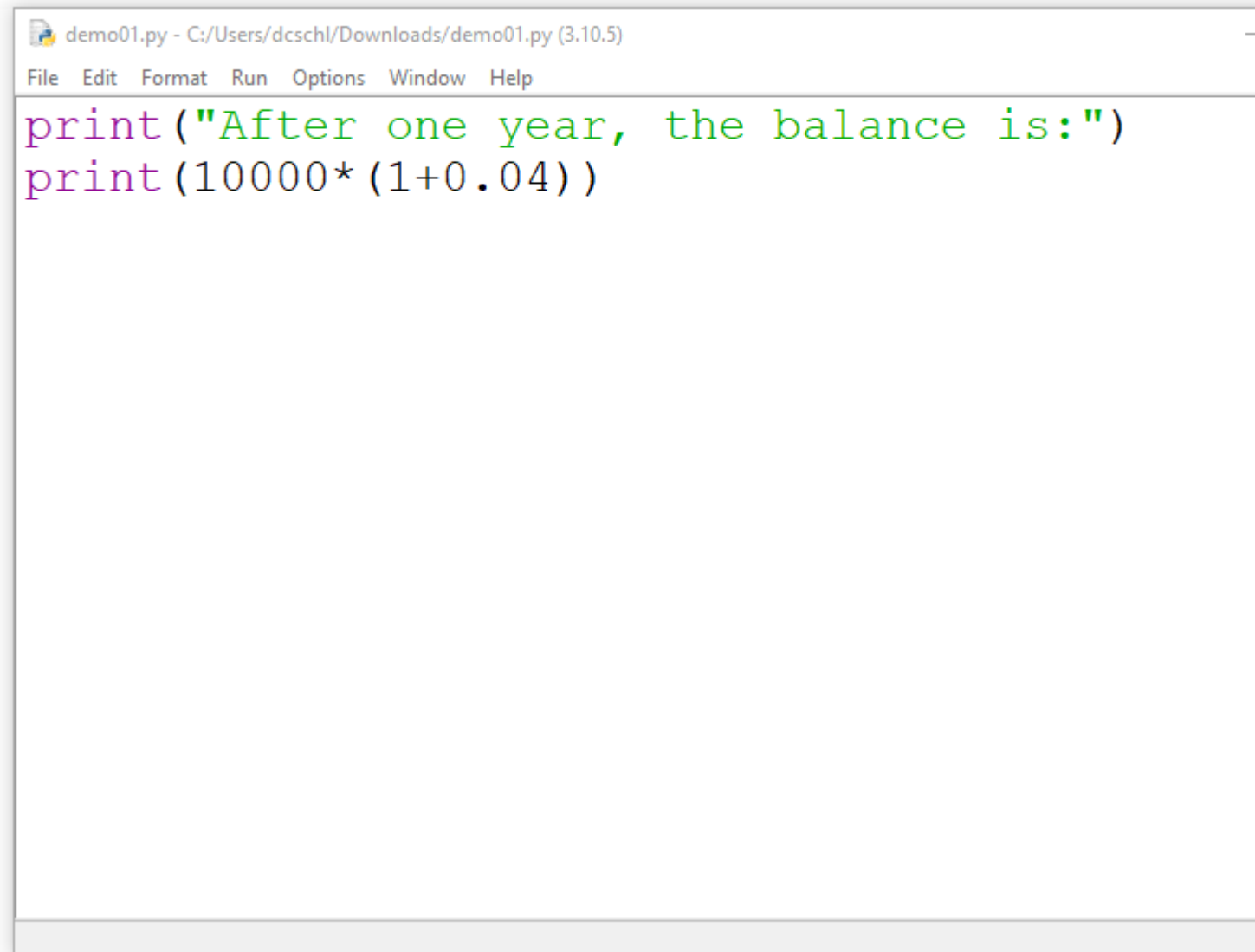


The screenshot shows a code editor window titled "demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the text `print (1`. The "Run" menu is open, showing options: "Run Module" (F5), "Run... Customized" (Shift+F5), "Check Module" (Alt+X), and "Python Shell".

```
demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)
File Edit Format Run Options Window Help
print (1
```

Code Editor

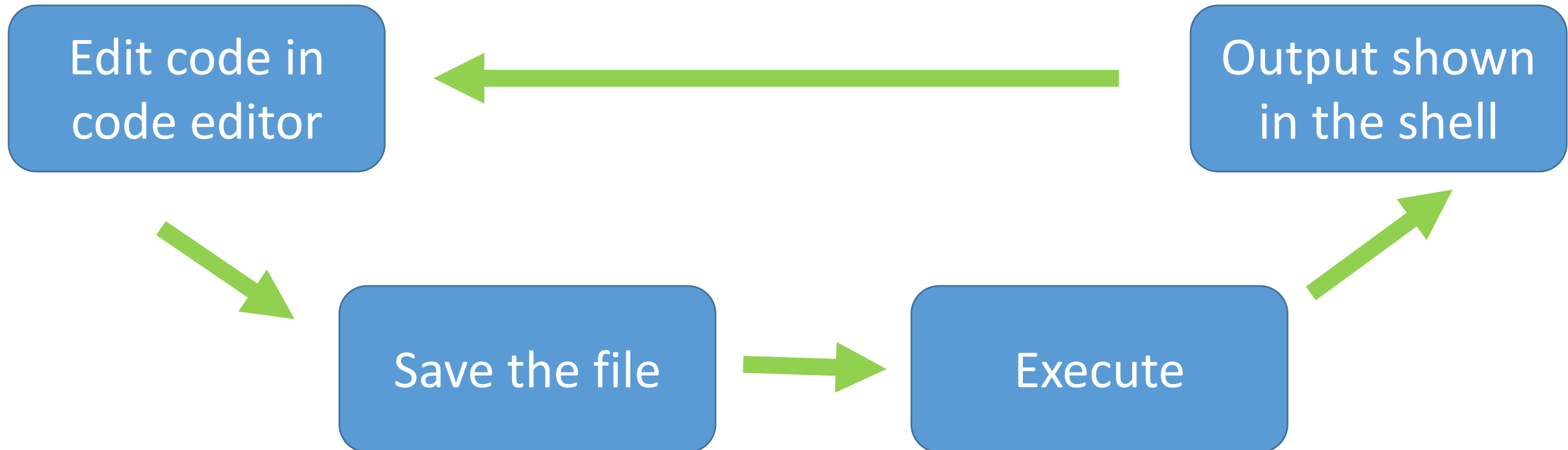
- Able to be
 - Saved
 - Edited
 - Reused
- Execute multiple lines
- Shell is mainly for testing



The image shows a screenshot of a Python code editor window. The title bar at the top reads "demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)". Below the title bar is a menu bar with the following options: File, Edit, Format, Run, Options, Window, and Help. The main editing area contains two lines of Python code. The first line is `print("After one year, the balance is:")` and the second line is `print(10000*(1+0.04))`. The code is color-coded: the `print` function names are in purple, the strings and parentheses are in green, and the numbers and operators are in black.

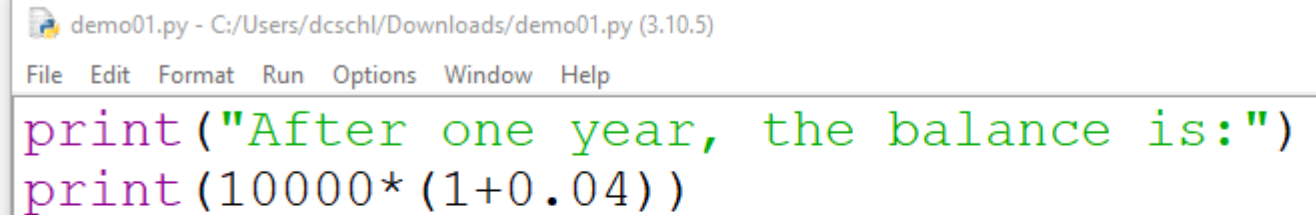
```
demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)
File Edit Format Run Options Window Help
print("After one year, the balance is:")
print(10000*(1+0.04))
```

Code Development Cycle



What is “print ()”?

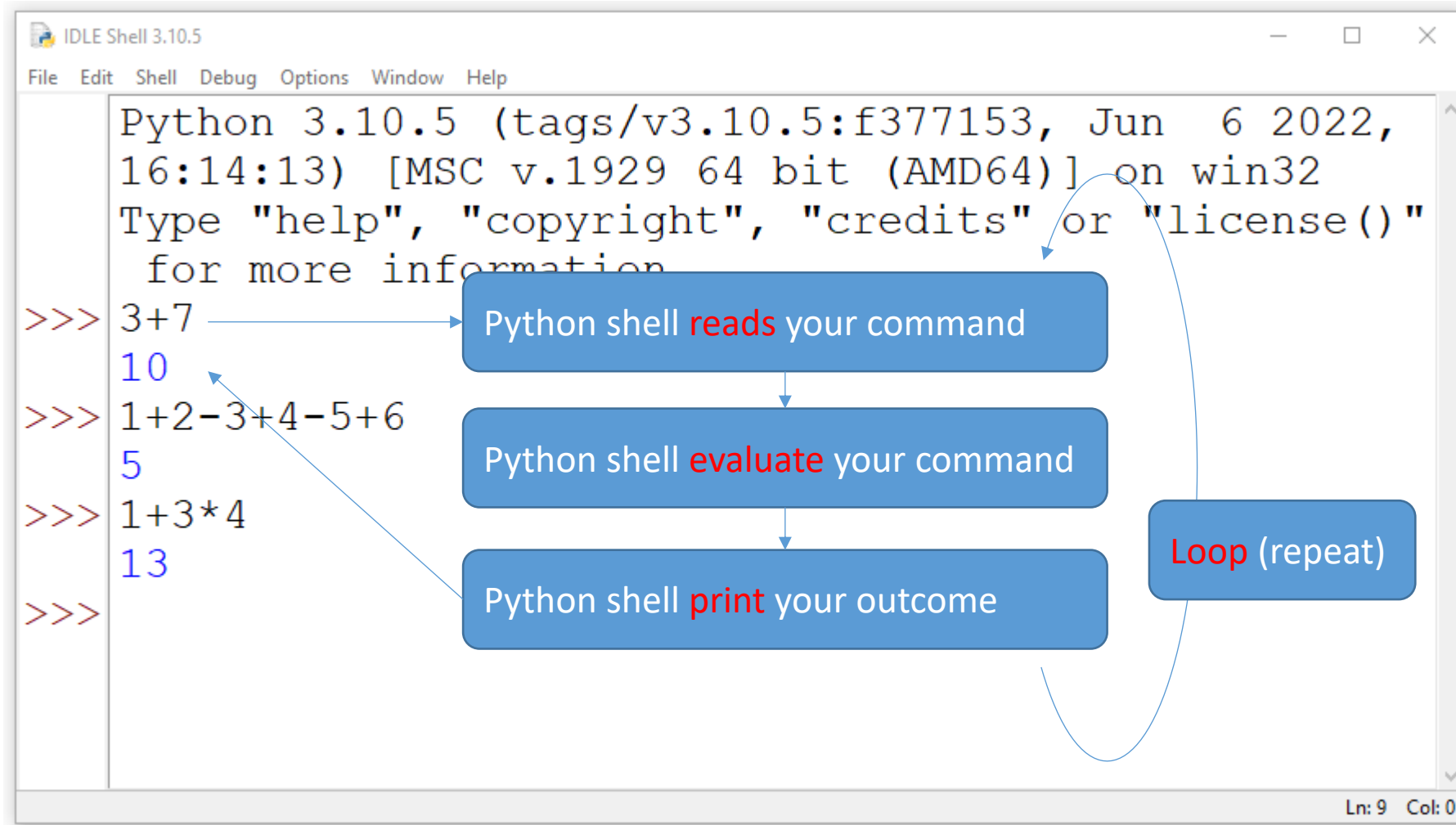
- print () is a function
- It “commands” Python to display what is in the bracket.
- But why we don’t need it when we run it in the shell?



The screenshot shows a window titled "demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code in the editor consists of two lines: `print("After one year, the balance is:")` and `print(10000*(1+0.04))`. The first line is green and the second is purple.

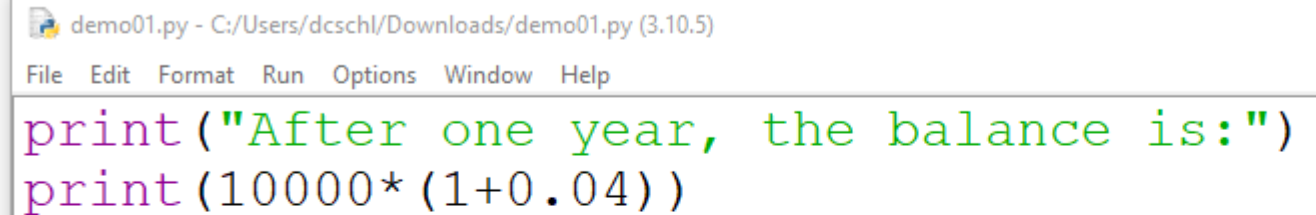
```
demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)
File Edit Format Run Options Window Help
print("After one year, the balance is:")
print(10000*(1+0.04))
```

Recap in Shell: Read-eval-print loop (REPL)



What is “print ()”?

- print () is a function
- It “commands” Python to display what is in the bracket.
- But why we don’t need it when we run it in the shell?
 - When you execute a program from the code editor, Python will only “execute” every line of code, but it will not “print”



The screenshot shows a code editor window titled "demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code content consists of two lines: `print("After one year, the balance is:")` and `print(10000*(1+0.04))`. The first line is highlighted in green, and the second line is highlighted in purple.

```
demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)
File Edit Format Run Options Window Help
print("After one year, the balance is:")
print(10000*(1+0.04))
```

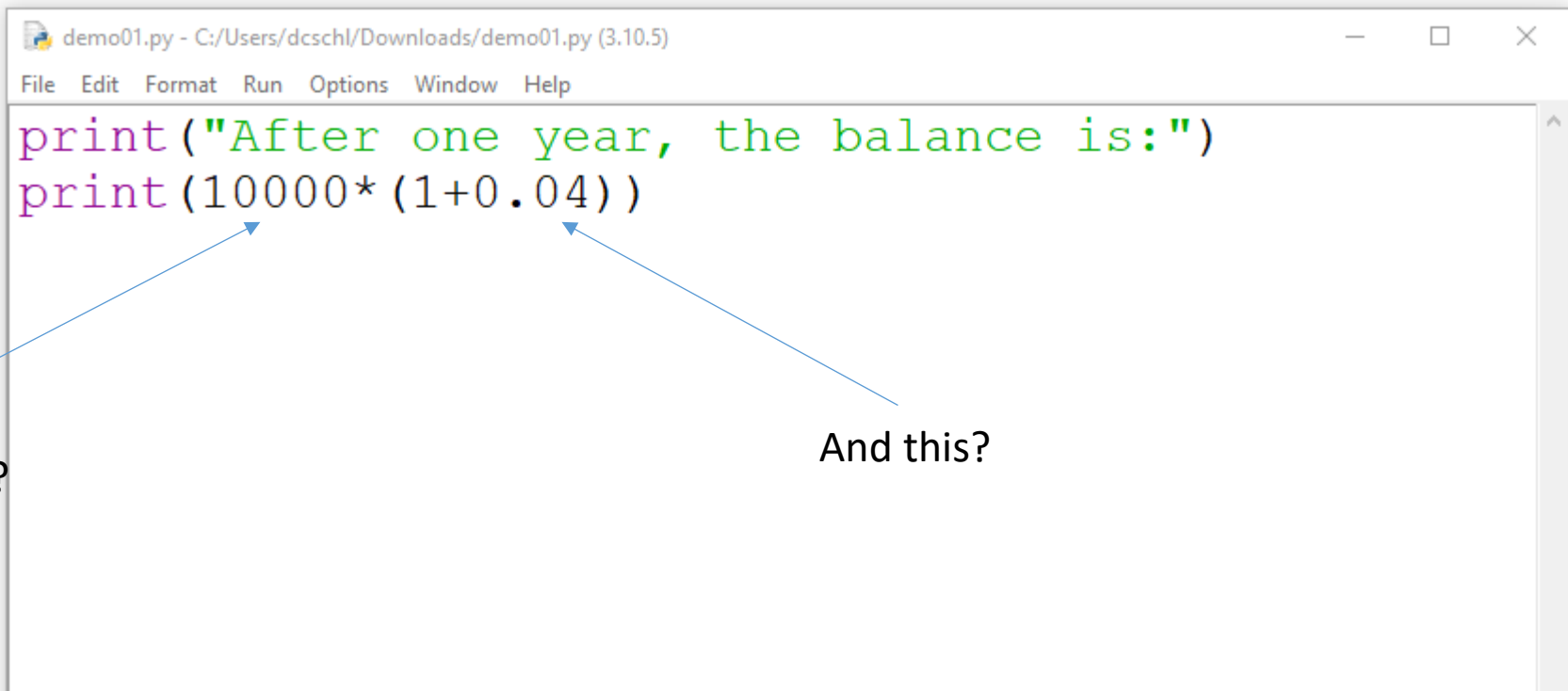
Break

Recap

- You put \$10000 into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?

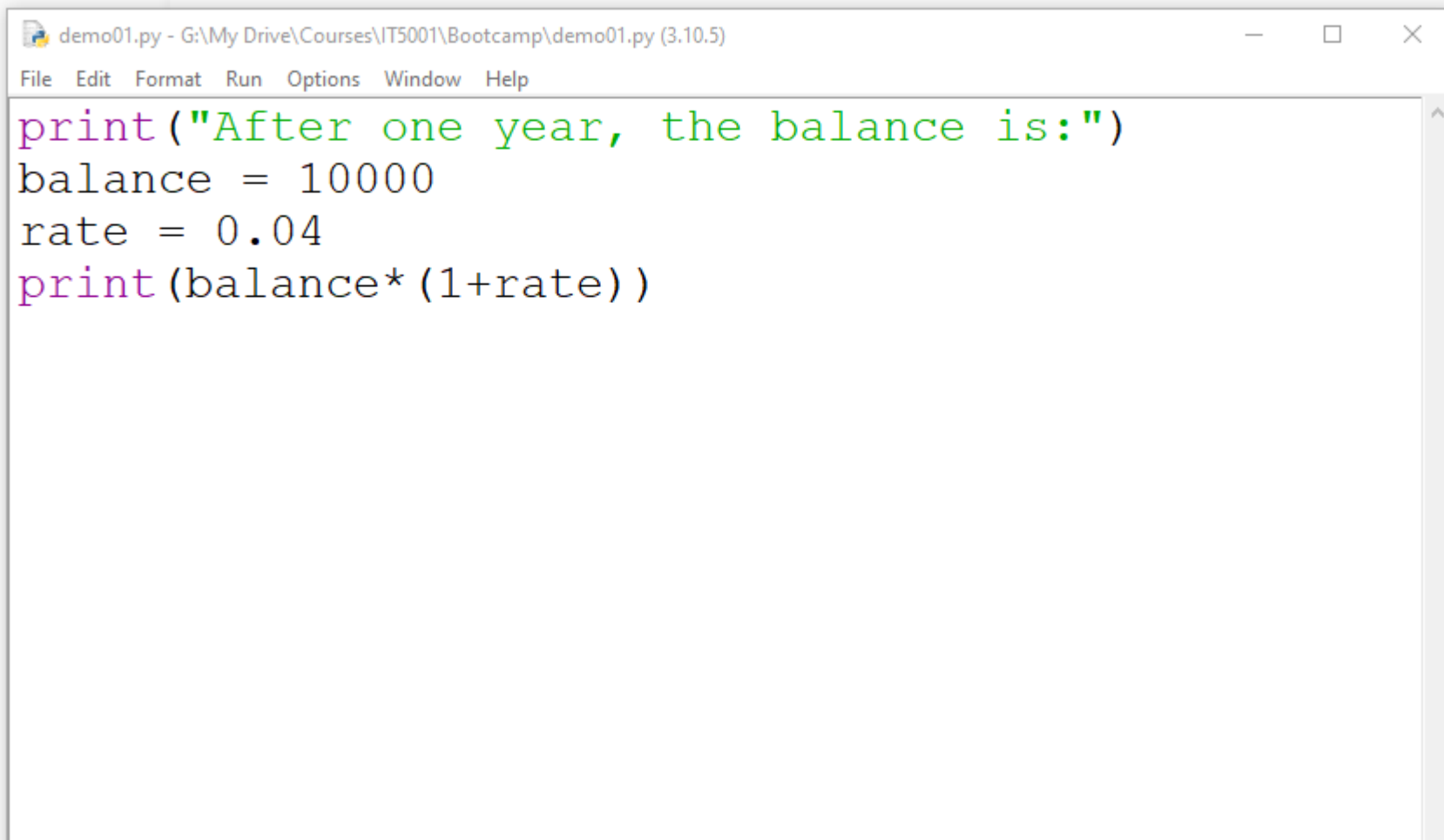
What is this?

And this?



```
demo01.py - C:/Users/dcschl/Downloads/demo01.py (3.10.5)
File Edit Format Run Options Window Help
print("After one year, the balance is:")
print(10000*(1+0.04))
```


A better way to do it: Using **Variables**



The image shows a screenshot of a Python IDE window titled "demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area contains the following Python code:

```
print("After one year, the balance is:")
balance = 10000
rate = 0.04
print(balance*(1+rate))
```

Try to solve the following problem by Python

- You put \$10000 into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?
- You put **\$15000** into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?
- You put **\$23000** into your fixed deposit bank account and the annual interest is **4.5%**
 - How much do you have after one year?

Did you type in the formula 3 times?
What if you need to do it 10 times? 100
times?

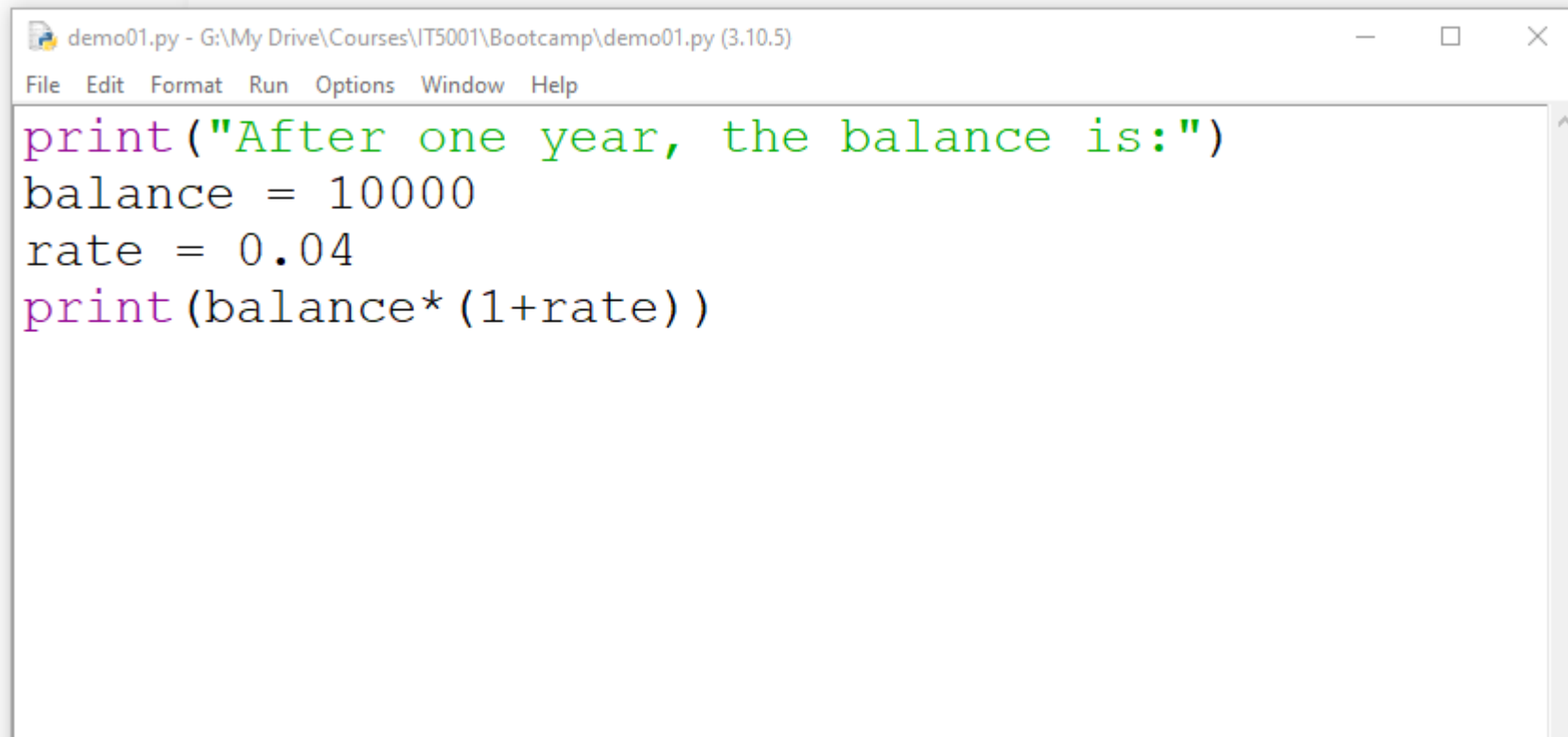
A better way to do it: Using Variables

```
demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)
File Edit Format Run Options Window Help
print("After one year, the balance is:")
balance = 10000
rate = 0.04
print(balance*(1+rate))
```

```
demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)
File Edit Format Run Options Window Help
print("After one year, the balance is:")
balance = 23000
rate = 0.045
print(balance*(1+rate))
```

Problem

- You put \$10000 into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?

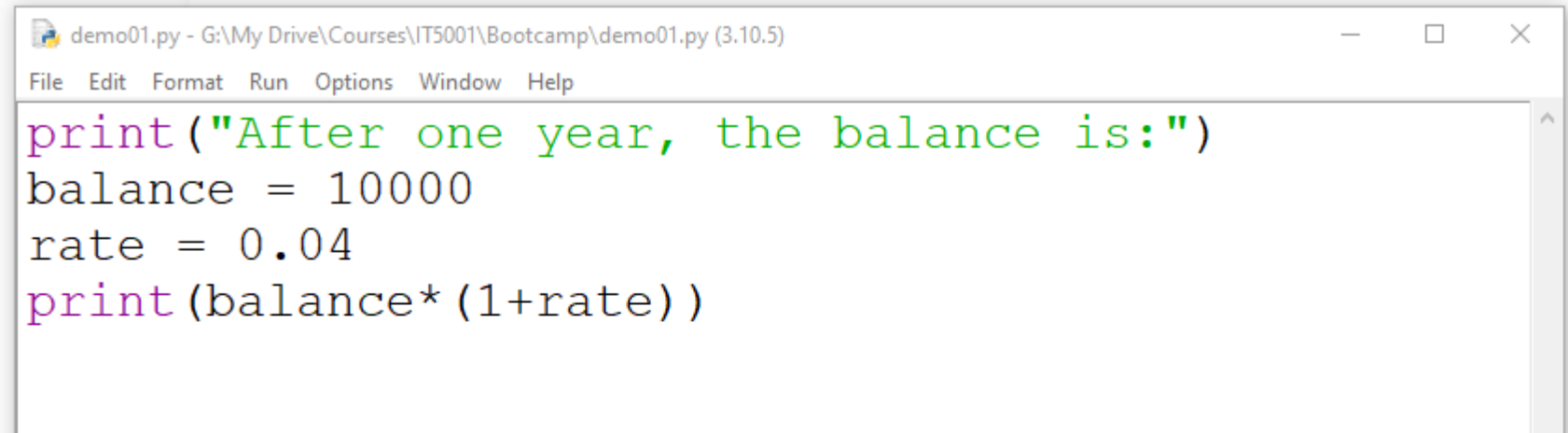


The image shows a screenshot of a Python IDE window titled "demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
print("After one year, the balance is:")
balance = 10000
rate = 0.04
print(balance*(1+rate))
```

Problem

- You put \$10000 into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year?
 - How much do you have after two years?
 - How much do you have after three years?
- We need to use the balance result AFTER one year to calculate the next year



```
demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)
File Edit Format Run Options Window Help
print("After one year, the balance is:")
balance = 10000
rate = 0.04
print(balance*(1+rate))
```

Problem

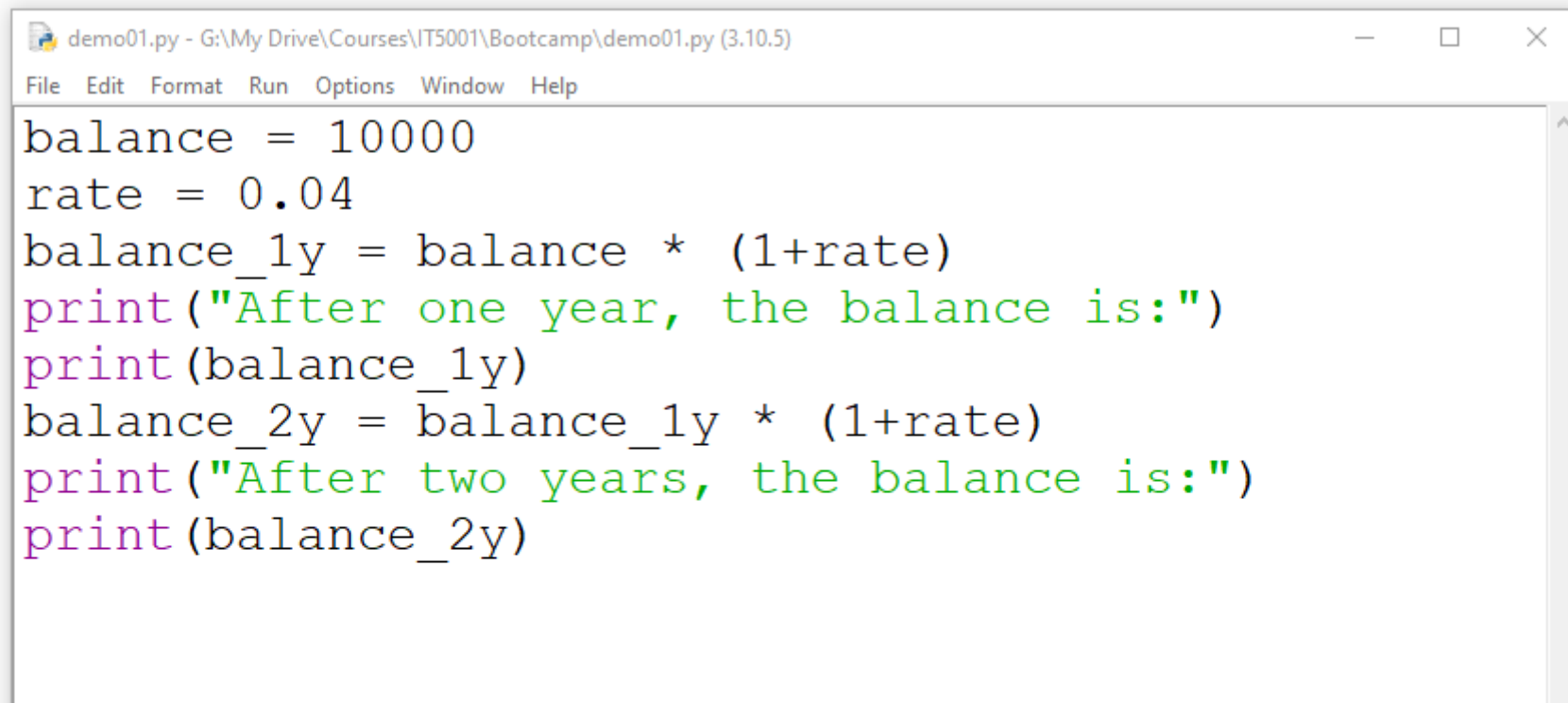
- You put \$10000 into your fixed deposit bank account and the annual interest is 4%
 - How much do you have after one year
 - How much do you have after two years
 - How much do you have after three years?
- We need to use the balance result AFTER one year to calculate the next year

```
demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)
File Edit Format Run Options Window Help

balance = 10000
rate = 0.04
balance_1y = balance * (1+rate)
print("After one year, the balance")
print(balance_1y)
balance_2y = balance_1y * (1+rate)
print("After two years, the balance")
print(balance_2y)
```

Calculating Balance

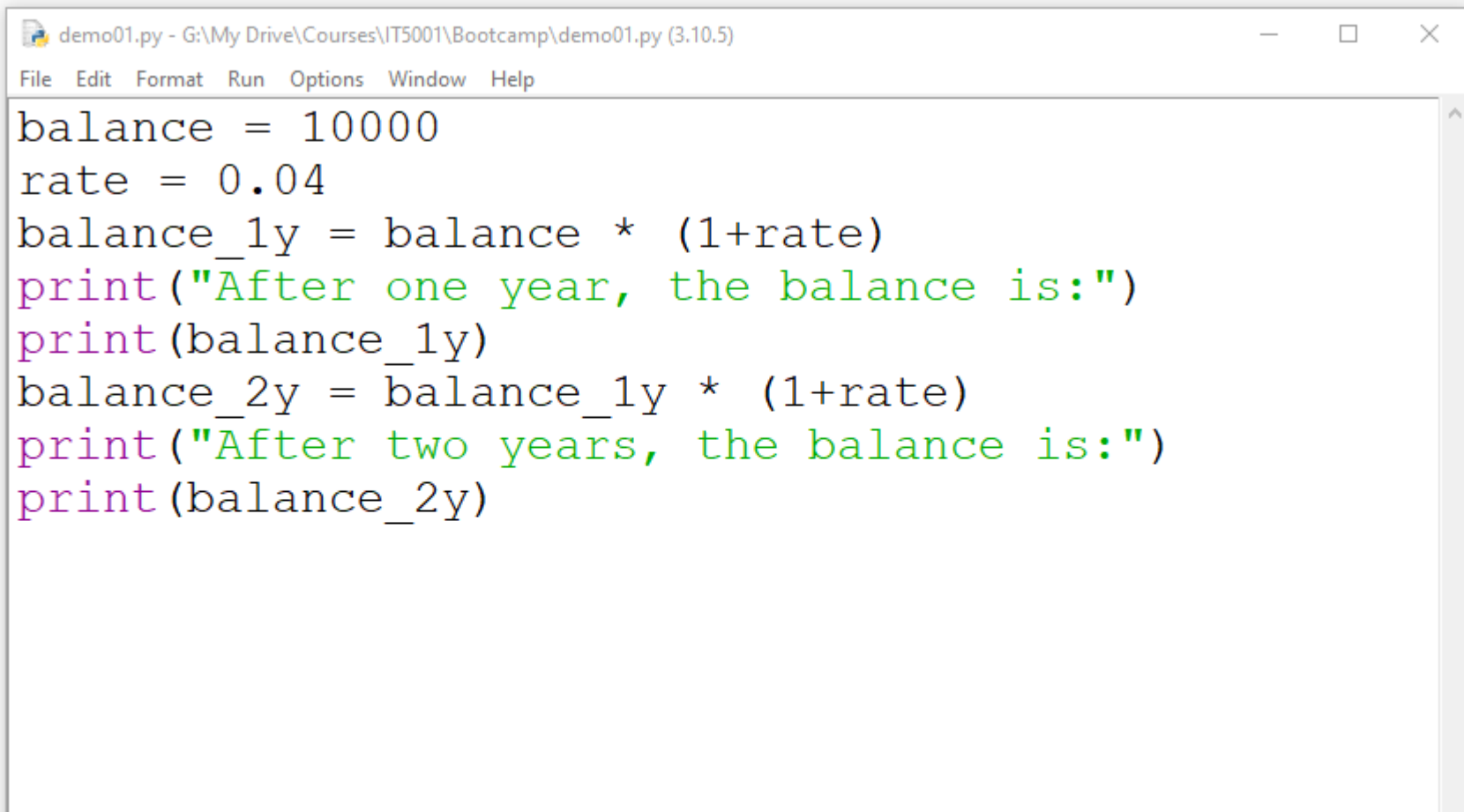
- In computing, a variable is a piece of memory that stores a value that can be *changed*.

A screenshot of a Python script editor window titled 'demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The script content is as follows:

```
balance = 10000
rate = 0.04
balance_1y = balance * (1+rate)
print("After one year, the balance is:")
print(balance_1y)
balance_2y = balance_1y * (1+rate)
print("After two years, the balance is:")
print(balance_2y)
```

Your job now

- Print out the balance in the next 3 years



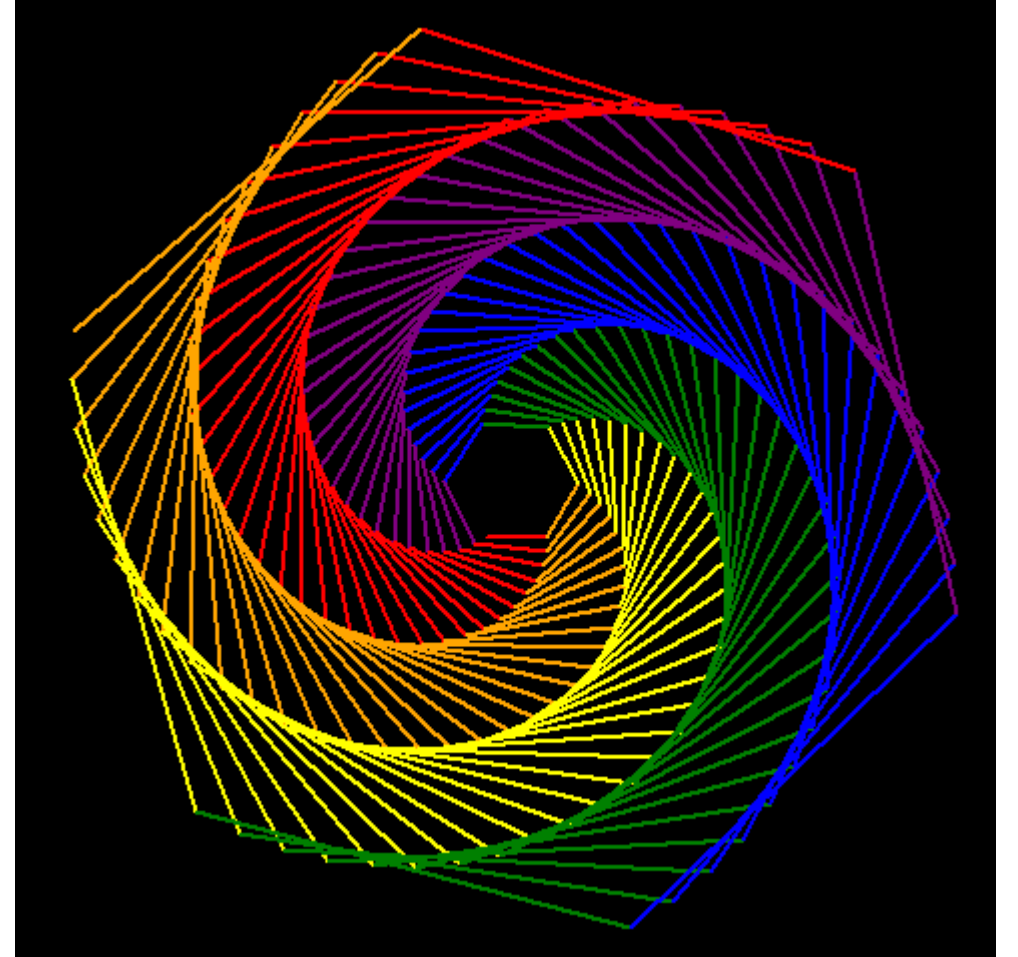
```
demo01.py - G:\My Drive\Courses\IT5001\Bootcamp\demo01.py (3.10.5)
File Edit Format Run Options Window Help
balance = 10000
rate = 0.04
balance_1y = balance * (1+rate)
print("After one year, the balance is:")
print(balance_1y)
balance_2y = balance_1y * (1+rate)
print("After two years, the balance is:")
print(balance_2y)
```


Turtle Graphics

Let's get familiar with Programming

Turtle Graphics

- “A picture is worth a thousand words”



Turtle Graphics

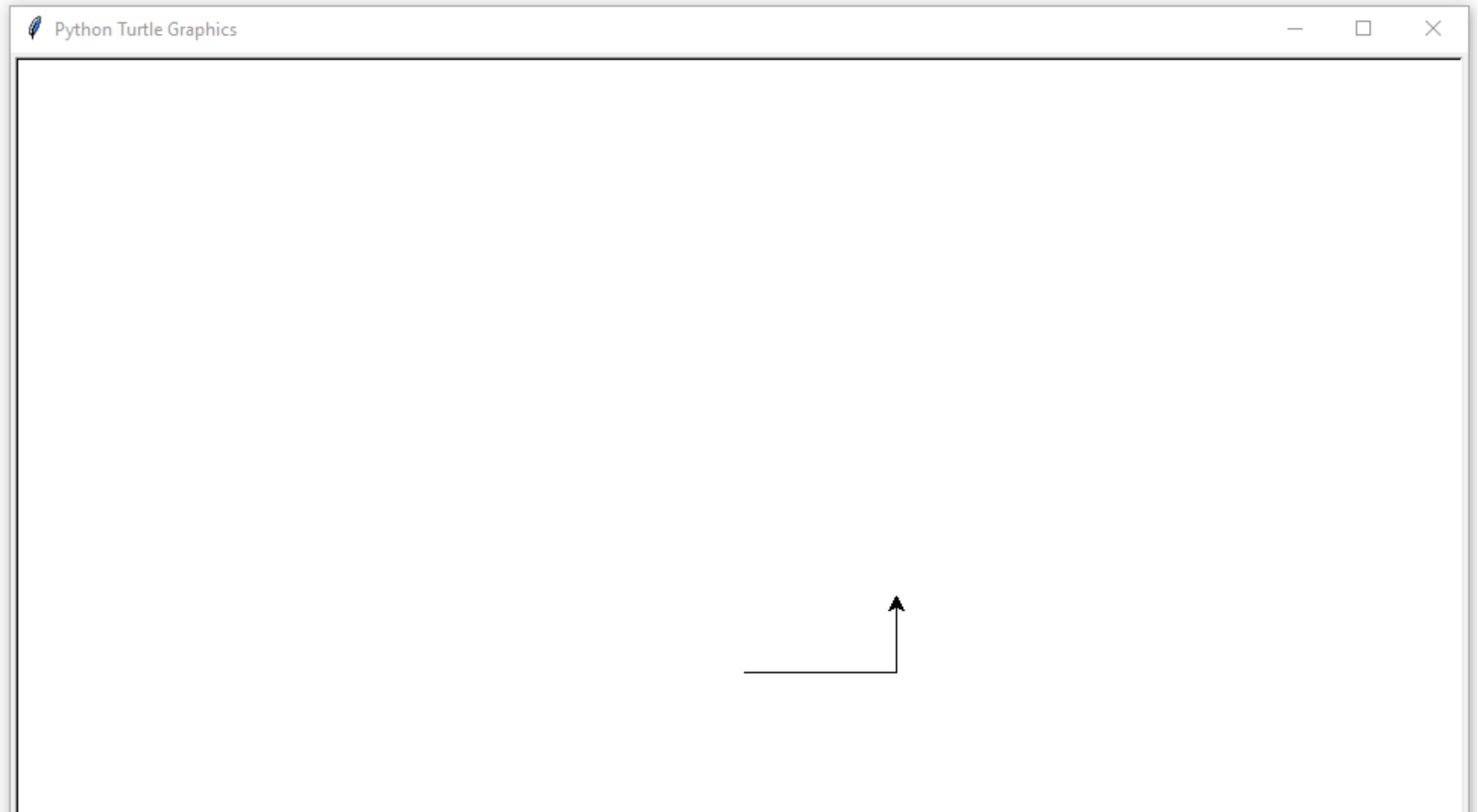
- “Turtle” is a Python feature like a drawing board, which lets us command a turtle to draw all over it!
- Imagine you have a little turtle on the screen and facing East
 - *picture*
- Then you command it to go forward 100 pixels
- Then turn left 90 degree and walk another 50 pixels
 - *picture*



Then you command it to go forward 100 pixels

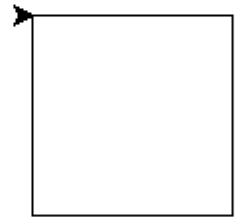


Then turn left 90 degree and walk another 50 pixels



Drawing a Square

```
forward(100)  
right(90)  
forward(100)  
right(90)  
forward(100)  
right(90)  
forward(100)  
right(90)
```

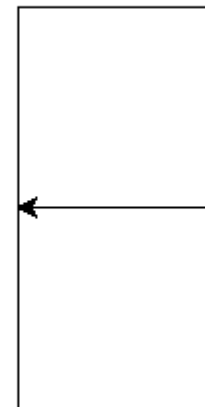
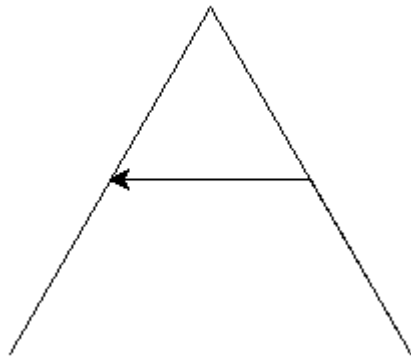


Assignment 1

- Draw a triangle
- And share how you draw it
 - (You may be tempted to use for-loop now, but let's try without it now)

Assignment 2

- Draw a letter A
- Share how you draw it
- Or did you draw other things?



How to learn programming in general?

- Google is your good friend
- E.g. how to get more commands in Turtle?
 - Google "turtle Python"

