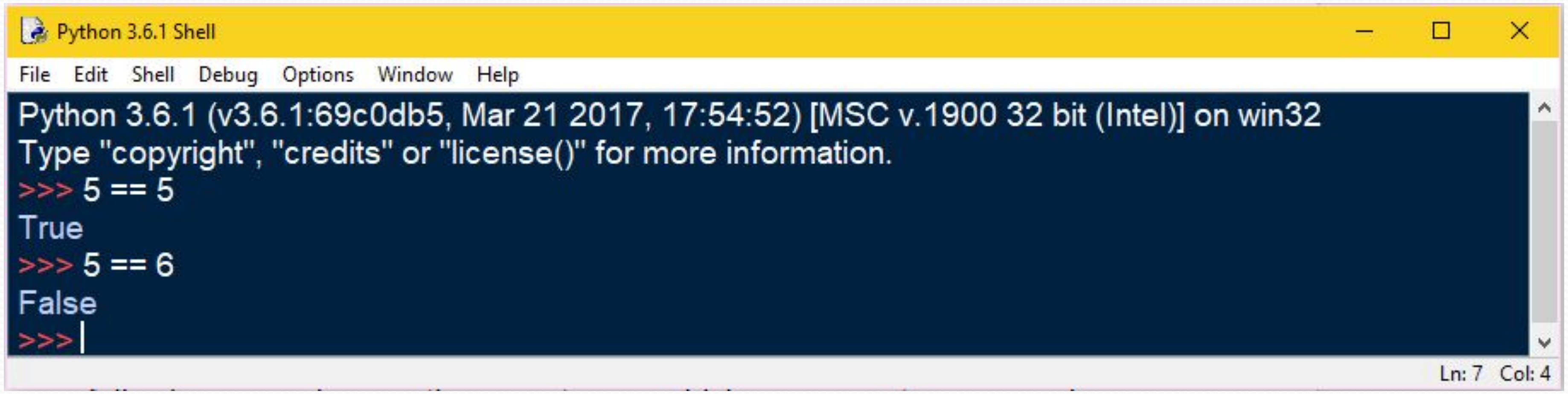
CONDITION EXECUTION

Agenda

- How to use “if condition” in conditional Structures
- How to use “elif” condition
- Nested IF Statements
- Debugging

Condition Execution

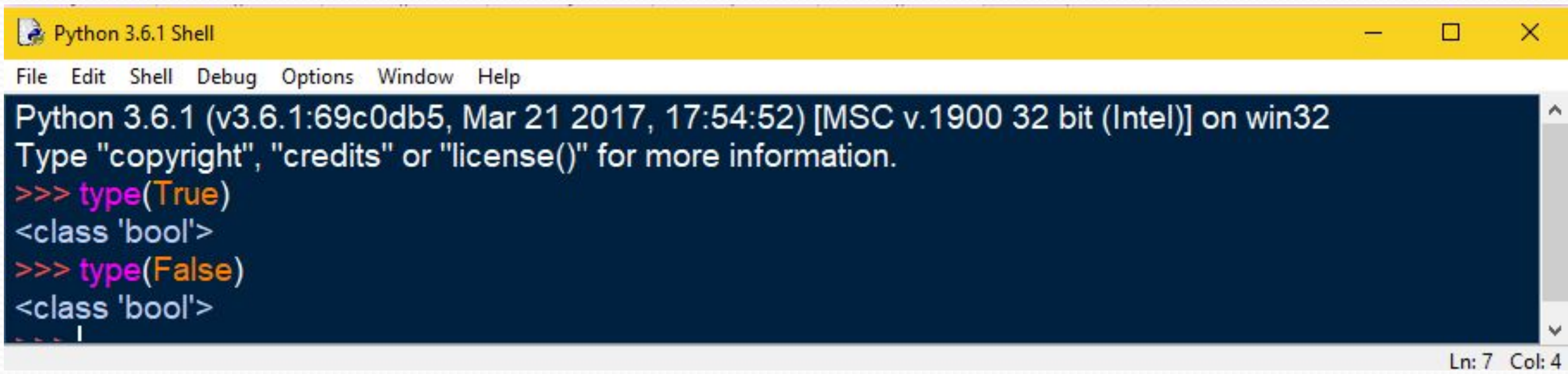
- Boolean Expression:A Boolean expression is an expression that is either true or false. The following examples use the operator ==, which compares two operands and produces True if they are equal and False otherwise:



```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 5 == 5
True
>>> 5 == 6
False
>>>
```

Continue…

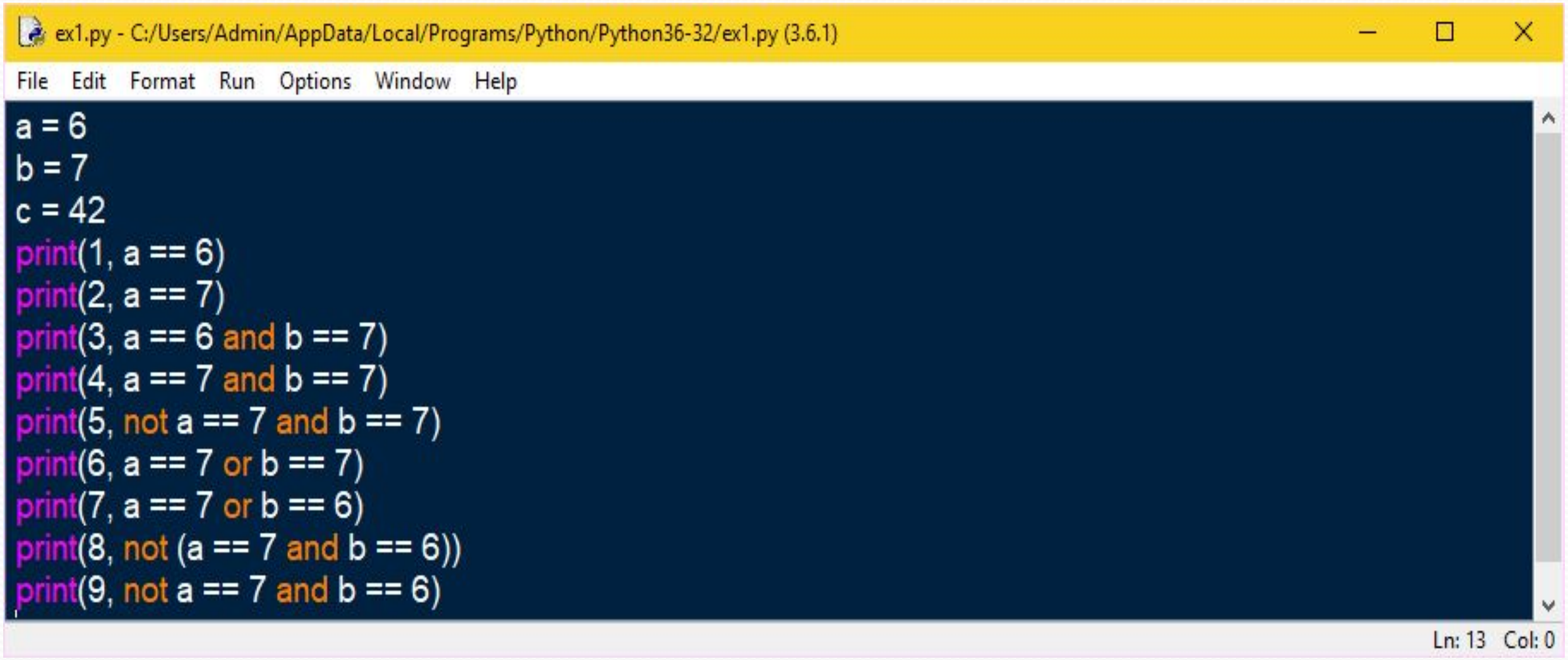
- True and False are special values that belong to the type bool; they are not Strings



```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
>>> |
```

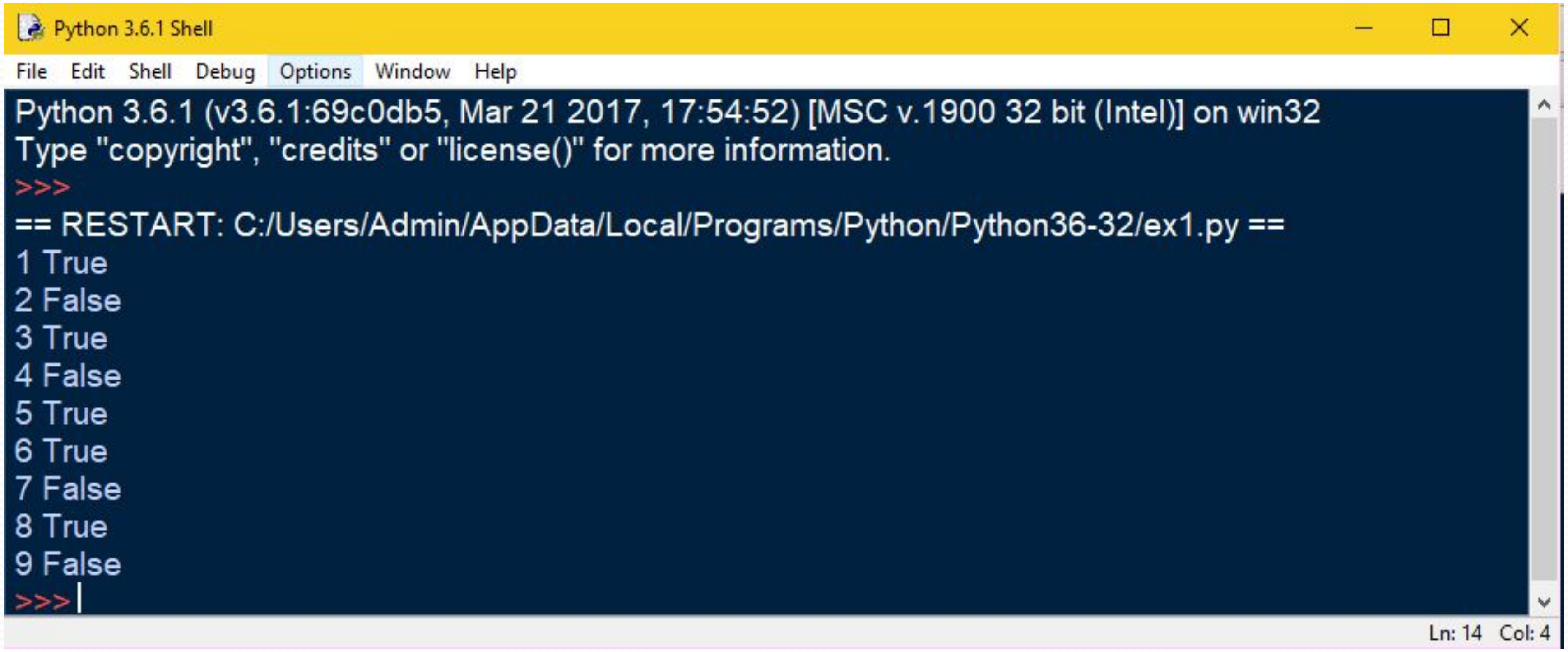
Continue…

- Some Boolean expressions are given below
- Example 1:


ex1.py - C:/Users/Admin/AppData/Local/Programs/Python/Python36-32/ex1.py (3.6.1)
File Edit Format Run Options Window Help
Ln: 13 Col: 0

```
a = 6
b = 7
c = 42
print(1, a == 6)
print(2, a == 7)
print(3, a == 6 and b == 7)
print(4, a == 7 and b == 7)
print(5, not a == 7 and b == 7)
print(6, a == 7 or b == 7)
print(7, a == 7 or b == 6)
print(8, not (a == 7 and b == 6))
print(9, not a == 7 and b == 6)
```

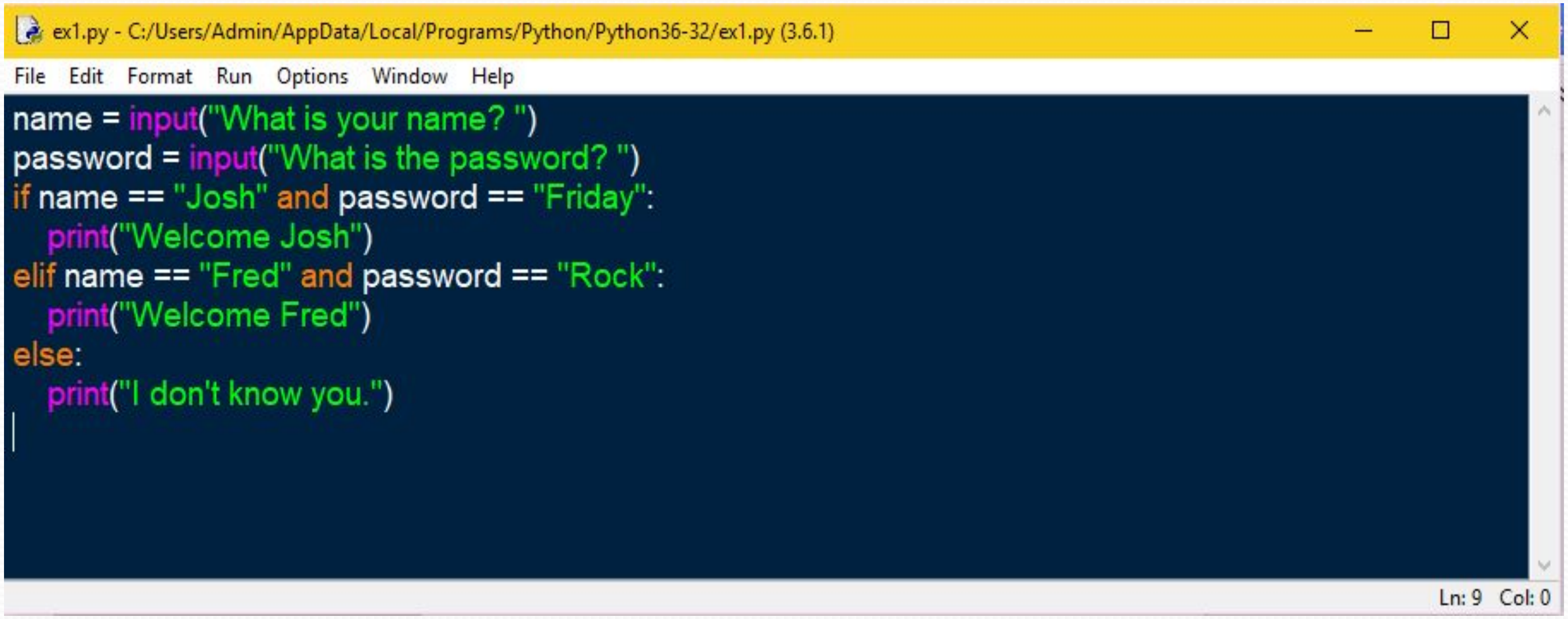
Continue…



```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python36-32/ex1.py ==
1 True
2 False
3 True
4 False
5 True
6 True
7 False
8 True
9 False
>>>
```

Continue….

- Example 2: This program asks a user for a name and a password it then check them to make sure that the user is allowed in.
- Program name: Sample.py


ex1.py - C:/Users/Admin/AppData/Local/Programs/Python/Python36-32/ex1.py (3.6.1)
File Edit Format Run Options Window Help
Ln: 9 Col: 0

```python
name = input("What is your name? ")
password = input("What is the password? ")
if name == "Josh" and password == "Friday":
    print("Welcome Josh")
elif name == "Fred" and password == "Rock":
    print("Welcome Fred")
else:
    print("I don't know you.")
```

Logical operators

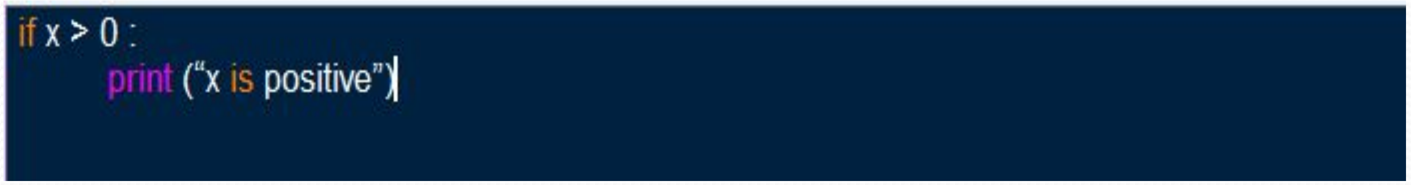
- There are three logical operators: and, or, and not. The semantics (meaning) of these operators is similar to their meaning in English.
- For example x>0 and x<10, is true only if x is greater than 0 and less than 10.
- n%2==0 or n%3==0, is true if either of the conditions is true, that is, if the number is divisible by 2 or 3.
- Finally, the not operator negates a Boolean expression, so not (x > y) is true if x > y is false, that is, if x is less than or equal to y.

Conditional execution

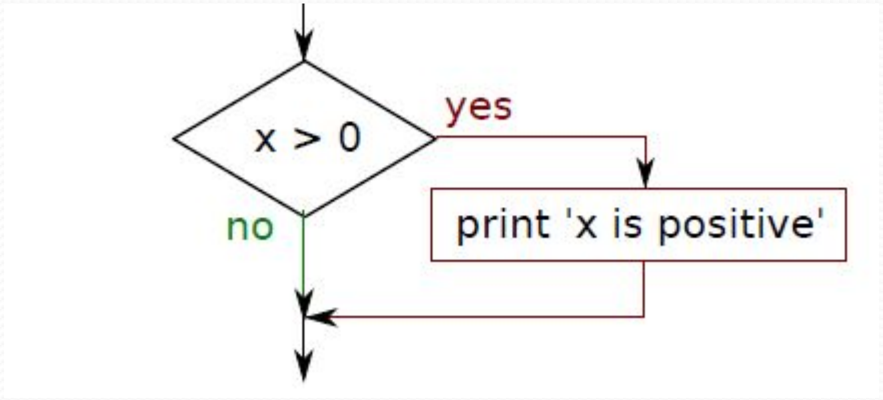
- In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly. Conditional statements give us this ability.
- The simplest form is the if statement
- An **if statement** consists of a boolean expression followed by one or more statements.

If Statement

- An **if statement** consists of a boolean expression followed by one or more statements.
- The Boolean expression after the if statement is called the condition. We end the if statement with a colon character (:) and the line(s) after the if statement are indented.

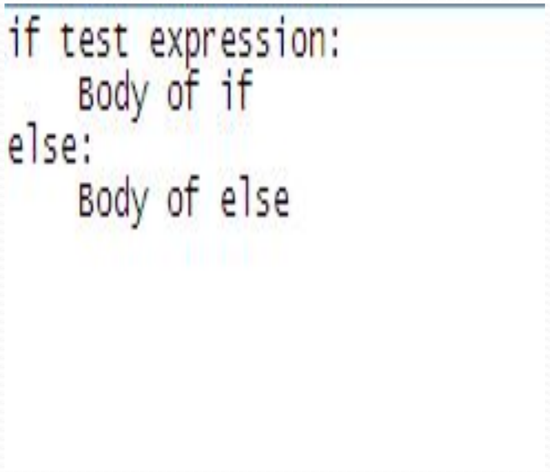


```
if x > 0 :
        print ("x is positive")
```

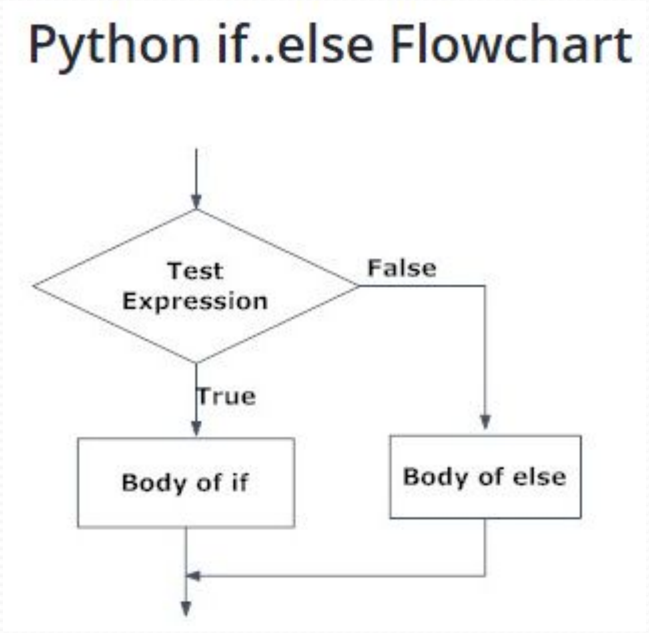


If else Statement

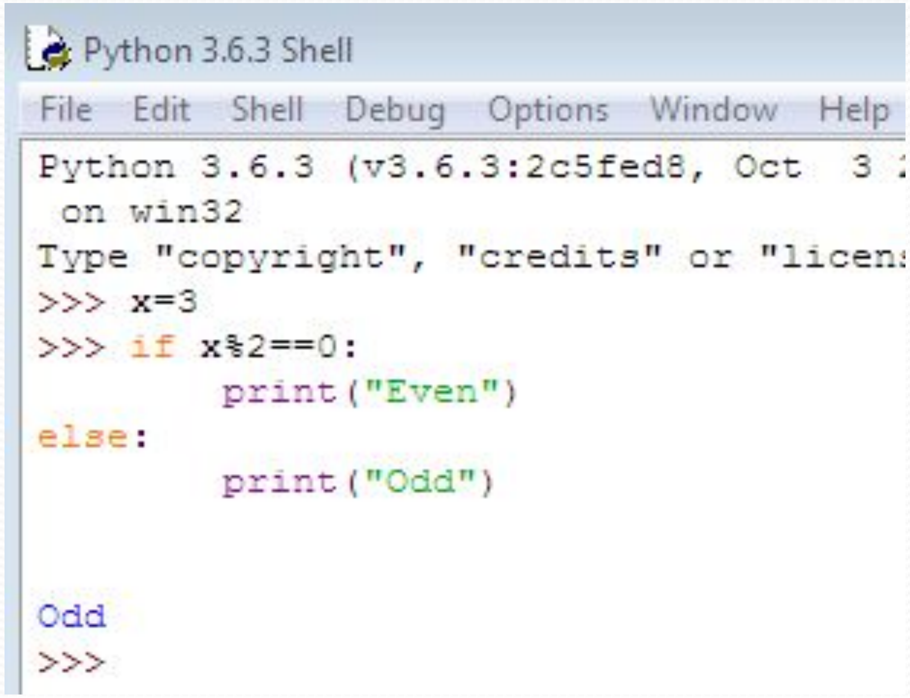
- The if..else statement evaluates test expression and will execute body of if only when test condition is True.
- If the condition is False, body of else is executed. Indentation is used to separate the blocks.



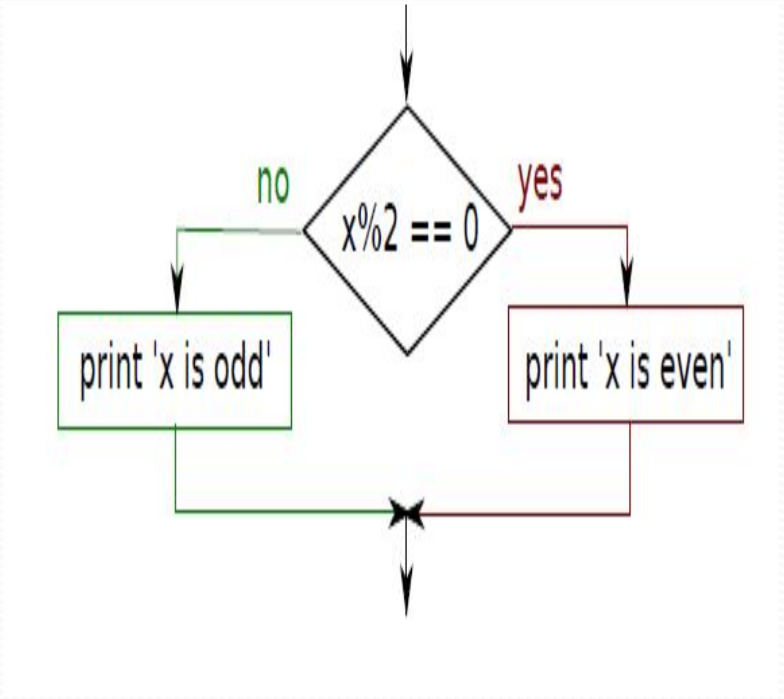
```
if test expression:
    Body of if
else:
    Body of else
```



Continue…

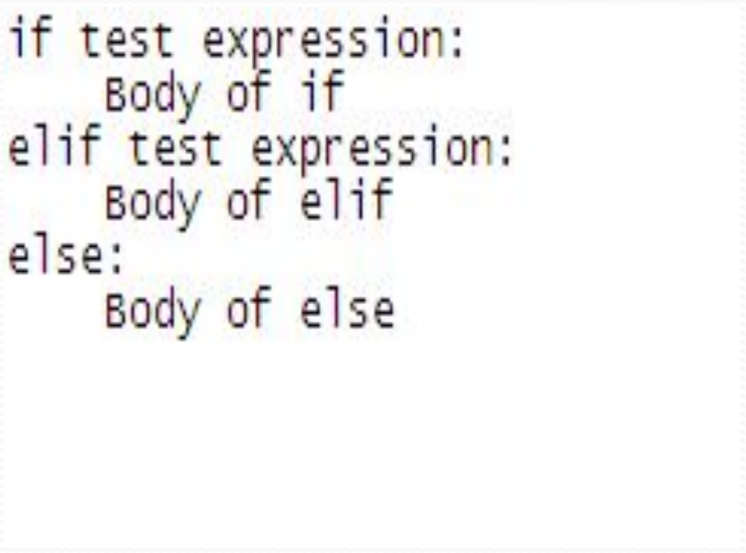


```
Python 3.6.3 (v3.6.3:2c5fed8, Oct  3 
 on win32
Type "copyright", "credits" or "licen
>>> x=3
>>> if x%2==0:
        print("Even")
else:
        print("Odd")


Odd
>>>
```

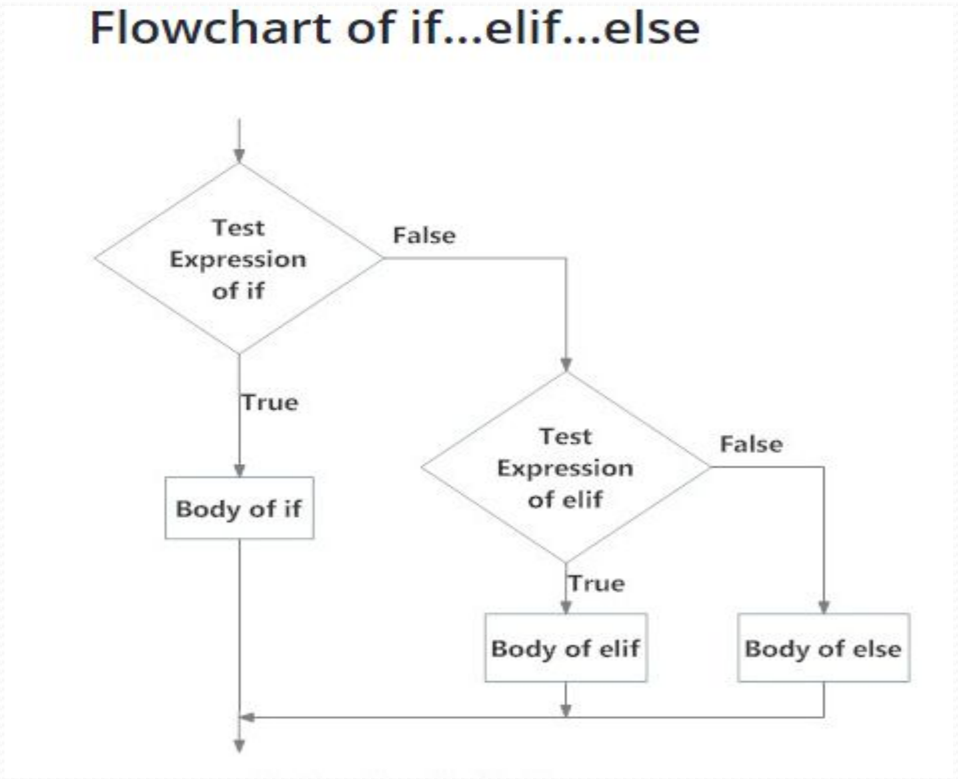


If..elif..else

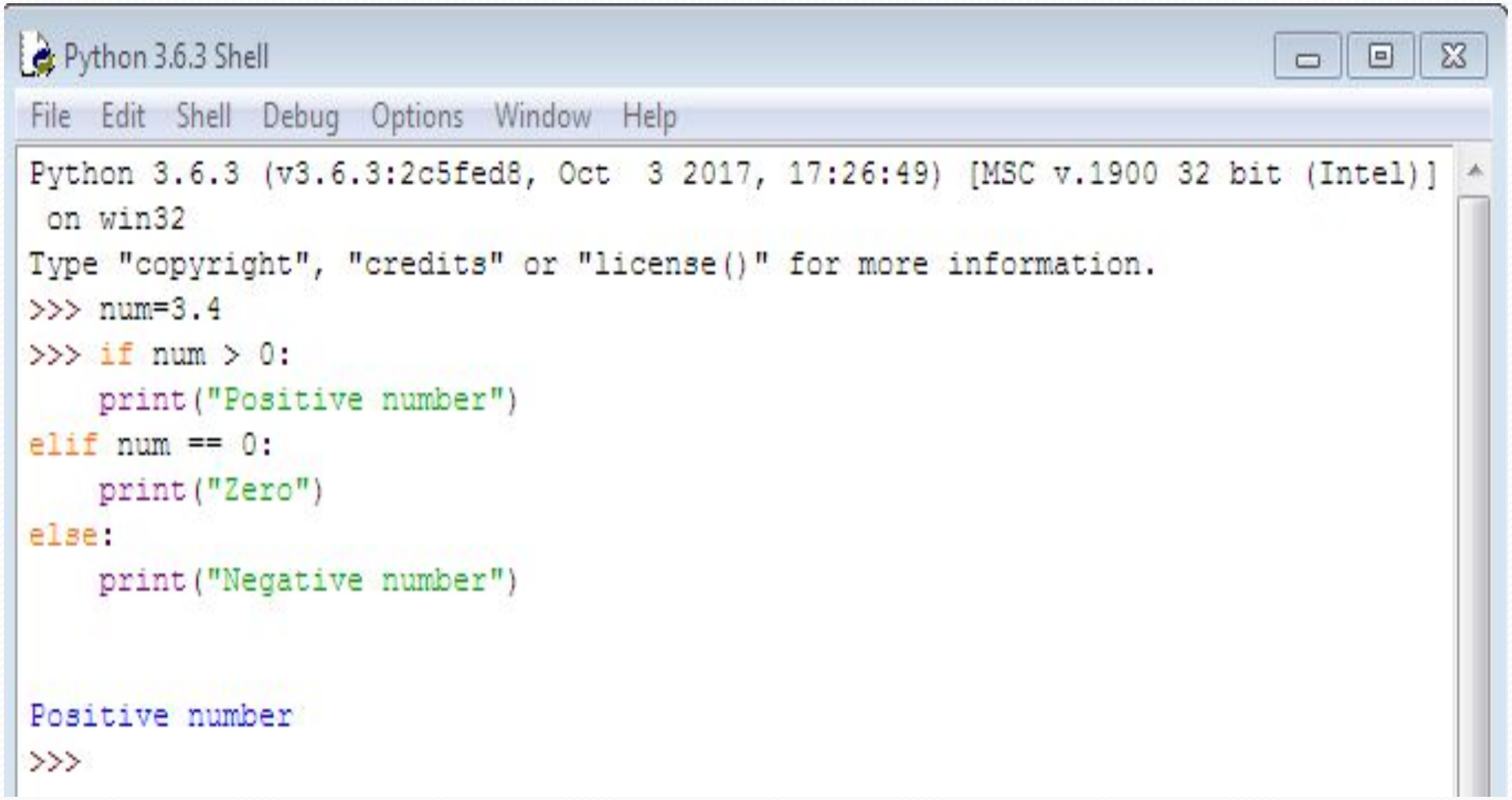
- The elif is short for else if. It allows us to check for multiple expressions.
- If the condition for if is False, it checks the condition of the next elif block and so on. If all the conditions are False, body of else is executed.



```
if test expression:
    Body of if
elif test expression:
    Body of elif
else:
    Body of else
```



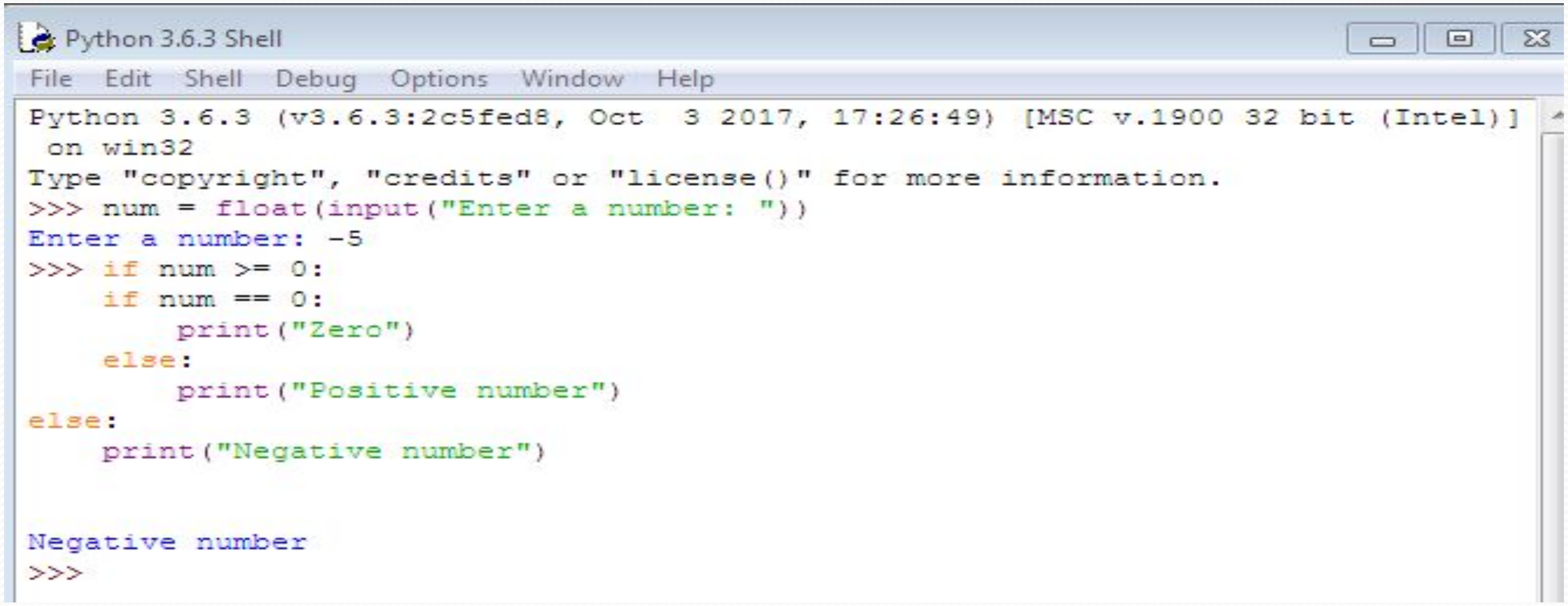
Example



```
Python 3.6.3 (v3.6.3:2c5fed8, Oct  3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>> num=3.4
>>> if num > 0:
	print("Positive number")
elif num == 0:
	print("Zero")
else:
	print("Negative number")



Positive number
>>>
```

Nested If Statement

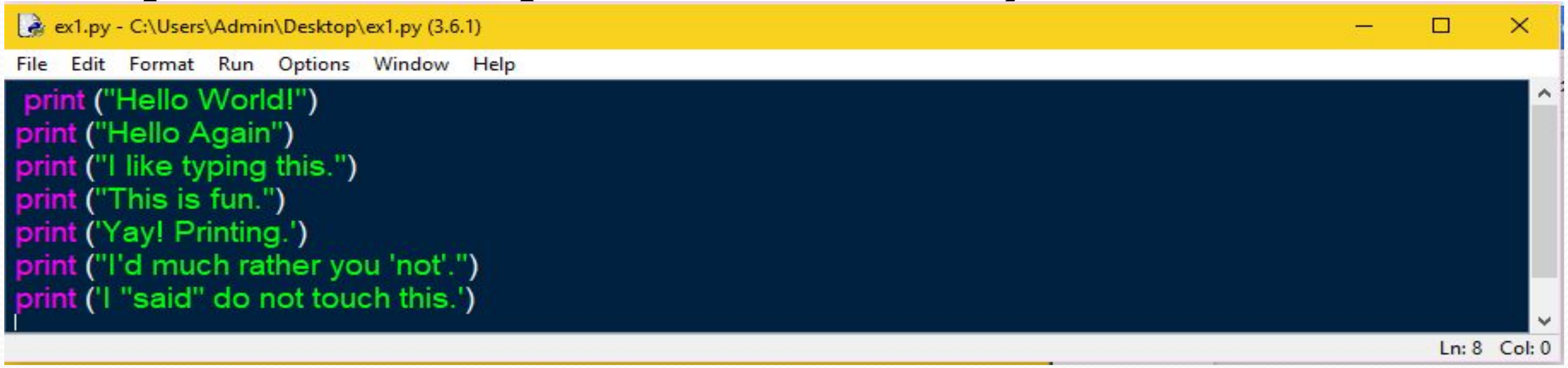
- We can have a if...elif...else statement inside another if...elif...else statement. This is called nesting in computer programming.



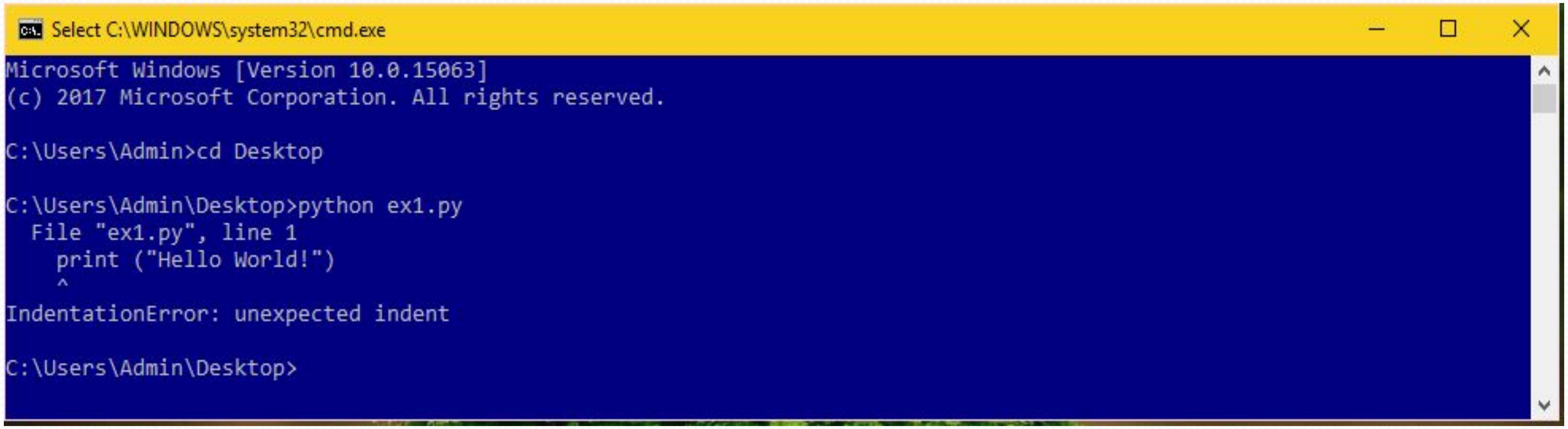
```
Python 3.6.3 (v3.6.3:2c5fed8, Oct  3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>> num = float(input("Enter a number: "))
Enter a number: -5
>>> if num >= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")


Negative number
>>>
```

Debugging

- **Debugging** is the process of finding and resolving defects or problems within the program that prevent correct operation of computer software or a system


ex1.py - C:\Users\Admin\Desktop\ex1.py (3.6.1)
File Edit Format Run Options Window Help
Ln: 8 Col: 0

```
 print ("Hello World!")
print ("Hello Again")
print ("I like typing this.")
print ("This is fun.")
print ('Yay! Printing.')
print ("I'd much rather you 'not'.")
print ('I "said" do not touch this.')
```



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

C:\Users\Admin\Desktop>python ex1.py
  File "ex1.py", line 1
    print ("Hello World!")
    ^
IndentationError: unexpected indent

C:\Users\Admin\Desktop>
```