1. Run git bash
2. Type **Python**
3. **14 == 7+7** check if 14 equals 7+7
4. **“Paul”** this is a string
5. **“Paul” + “ ’s excellent adventure ”** gives "Paul's excellent adventure"
6. **quit()** exit Python

* 7 and “text” are examples of Python objects. An objects is basically a container for data. It keeps data separate from each other. In computer memory, all objects are stored as zeros and ones. Each object in Python has a type. We can call it a type or a class these are essentially interchangeable in python 3.
* Python is a strongly-typed: it will restrict the operations you can perform, based on an object’s type. In a weakly typed language, the computer will let you do things like add functions together. e.g. you can add two integers together, but not two functions in Python

**EVEYTHING in PYTHON is in OBJECTS and EVERY OBJECTS in PYTHON has a TYPE**

* booleans (bool) take only truth or false

integers (int)

folats (float) any numbers that have a decimal point

strings (str) sequences of txt characters

* **type(-345)** this checks the type of -345 and gives <class ‘int’>
* **type (Ture)** this checks the type of True and gives <class ‘bool’>

in Python, only True and Flase are bool

* **type (“Book”)** this checks the type of “Book” and gives <class ‘str’>

because python is strongly type, you can not do **4 + “5.5”** even though you can do 4+4.6

however, you can do **4 + float(“5.5”)** this is a constructor (type caustic ), a special functions that create an object of a particular type and the name of the function is just the name of the type we want to create . if we do **int(7.5)** it will give us 7. However, the constructor can not do everything, if we do int(“5.5”) this will give us an error. However, I can fix this by **int(float("5.5"))**

**literal is sth that python can read and immediately create an object out of it**

e.g if we type **5**, python will create an int literal.

When you divide, you always get a float in python. E,g **type(15/5)** gives you <class ‘float’>

* division **14 / 5 = 2.8**
* quotient **14 // 5 = 2**
* remainder **14 % 5 = 4**
* power **2 \*\* 5 = 32**
* comparison operator **6 >= 2 + 2** this will gives True

BECAREFUL WHEN YOU COMPARING FLOATING NUMBERS!!!!

e.g. **.1 + .1 + .1 == .3** will give you False

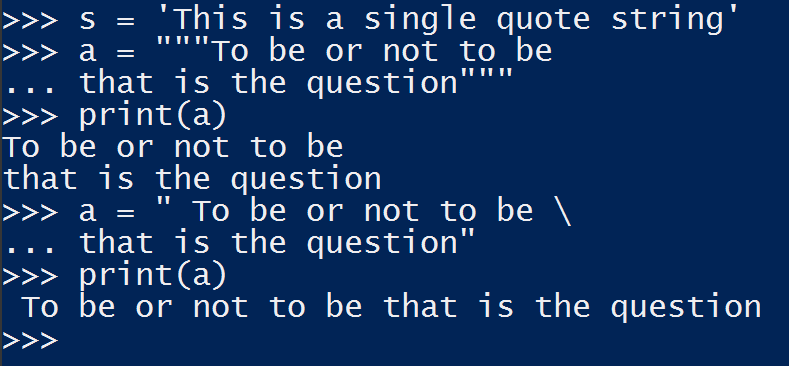
print() ……

**a -= 1** is the same as a = a-1

**b //= 5** is the same as b = b//5

Python is dynamic typed program. This means that an object can be one type somewhere but another type somewhere else. E.g a = 3 a = “char” ,than a is a str type !

Single quote and double quote both create the same string

We use triple quote to do several lines of strings , we can use \ to escape the return 

We can also do constructor to do string , e.g. **C = str(“Hi”)** this gives a C as a string

We can also do **x = "To be or not to be\nThat is a question"** this gives a two line string even though I typed in one line

We can insert a tab by **x = "To be or not to be\tThat is a question"** this gives a tab between the two sentences

We can put quotations inside a string by escape by

**x = "He said \”To be or not to be, that is a question. \””**

We can put a backslash inside the sentences by escape the backslash

X = “I just wanted to use a backslash \\ in this example”

Concatenation **a = “The ” b = “best class” a+b**  gives ‘The best class’ notice concatenation does not add space unless we add it ourselves

Multiple times e.g. **a \* 4** gives ‘The The The The ’

Extract string **s = “Absolute” print (s[0])** gives the first letter which is A

**print(s[-1])** gives the last letter which is e

slice **s[1:4]** will gives ‘bso’

**s[1:6:2]** this will give ‘bou’ because it set step size to be 2

**s[1:]** this will give ‘bsolute’ , the second to the last

**s[:-1]** this will give ‘Absolut’, the first to the second last

**s[::-1]** this will give ‘etulosbA’ the reverse of the string

**s[:]** this will give ‘Absolute’ everything

change the case by

**s.upper()** this will give ‘ABSOLUTE’

**s.lower()**  this will give ‘absolute’

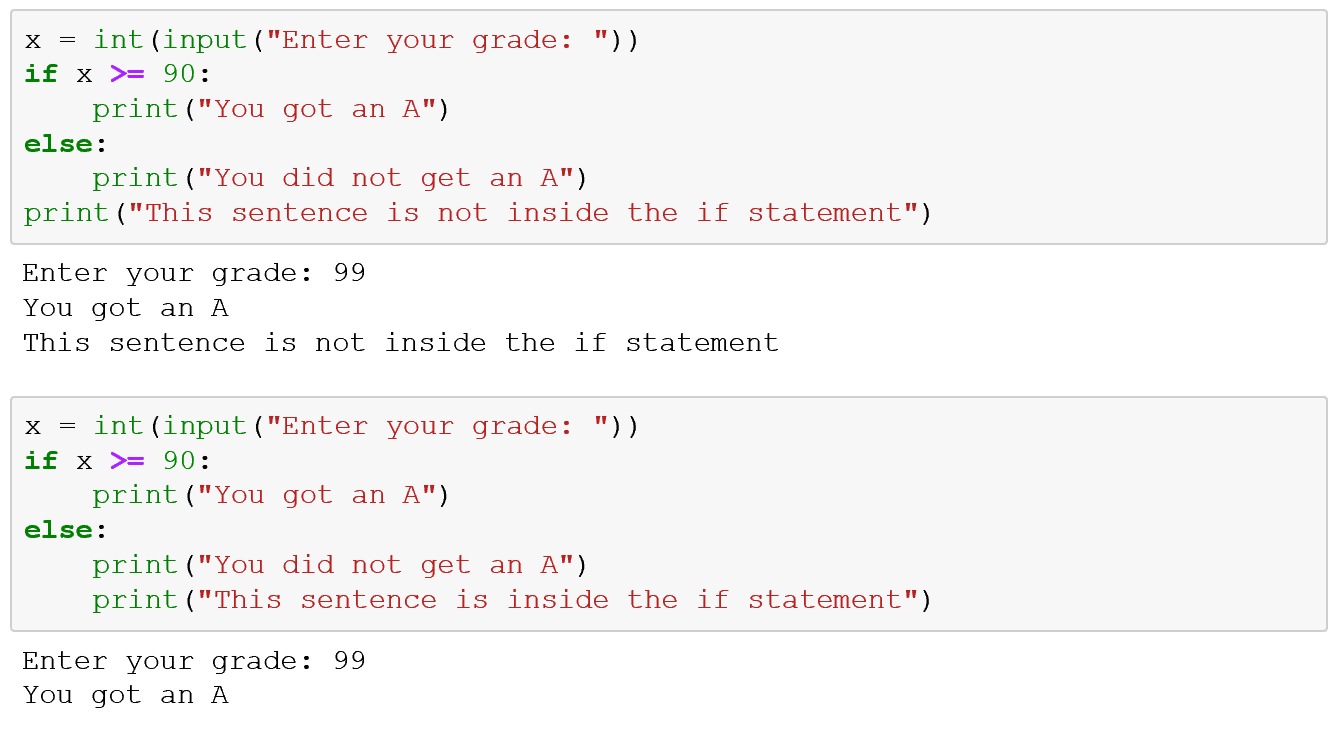
enter prompt

**name = input (“Enter your name : “)**  this will give Enter your name : and wait you to enter

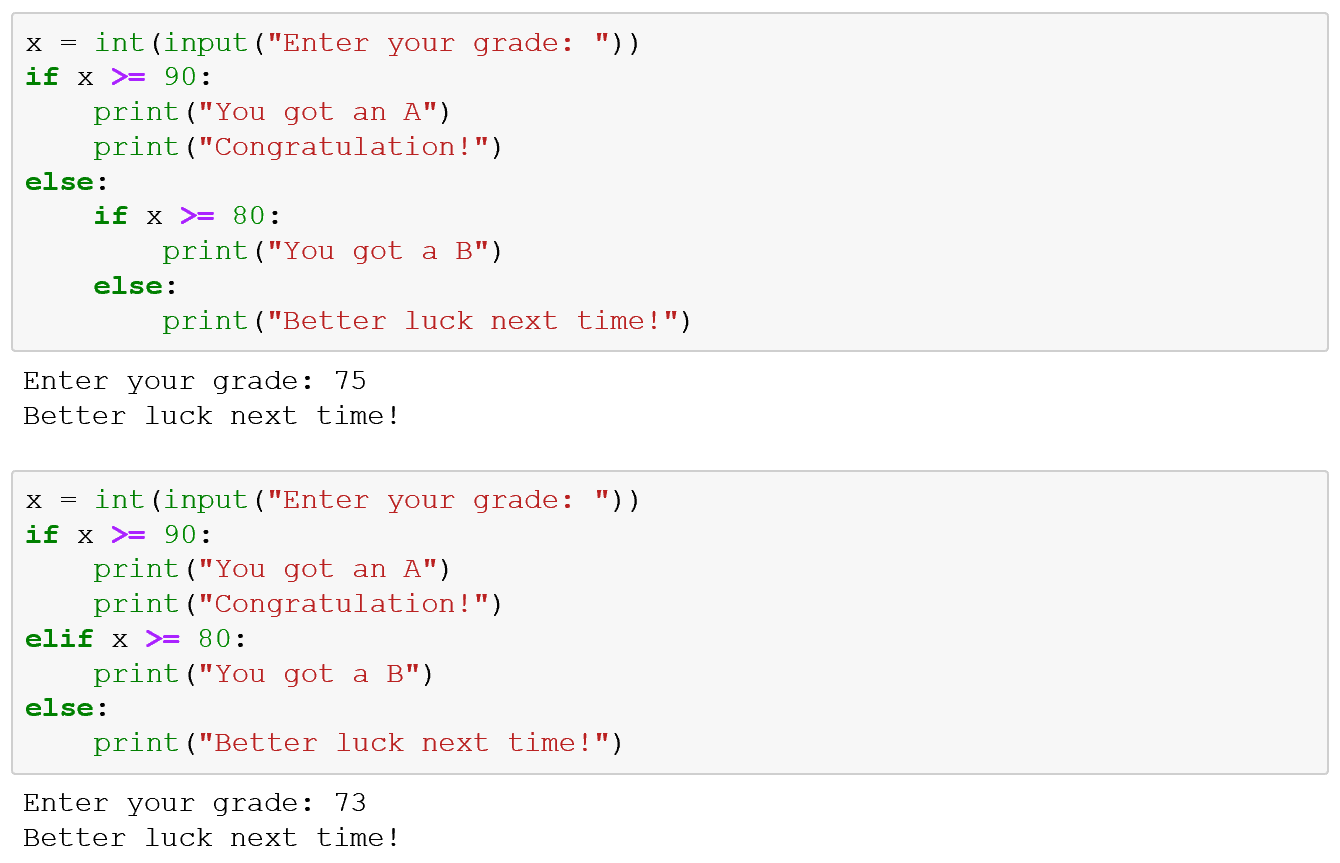
remember, whatever you put into the input will be saved as a str even if you enterer a number

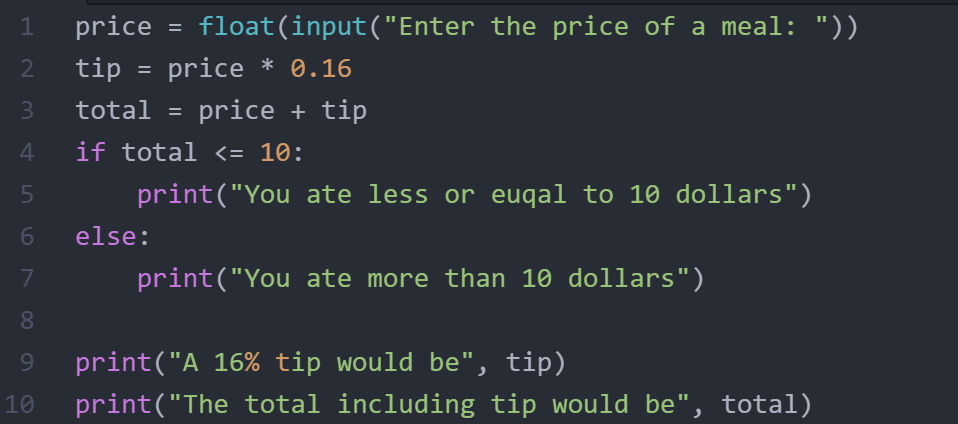
however, you can convert into other types using name = float(name)

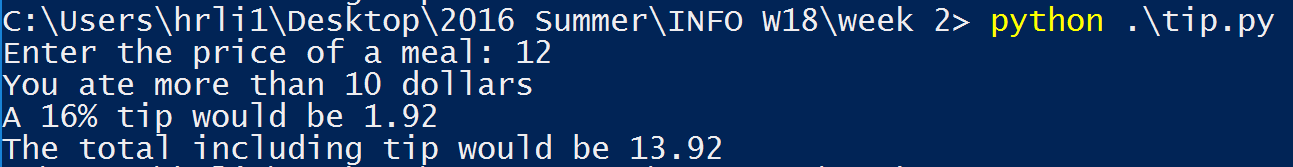
in Python, when you do if statement, there is no {} or end to end sth, there is only indentation

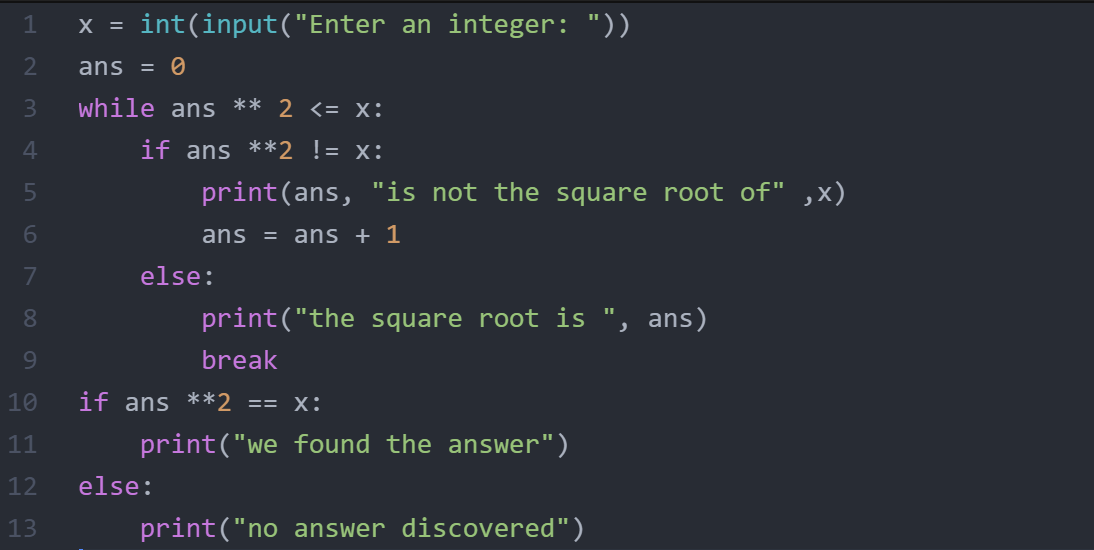


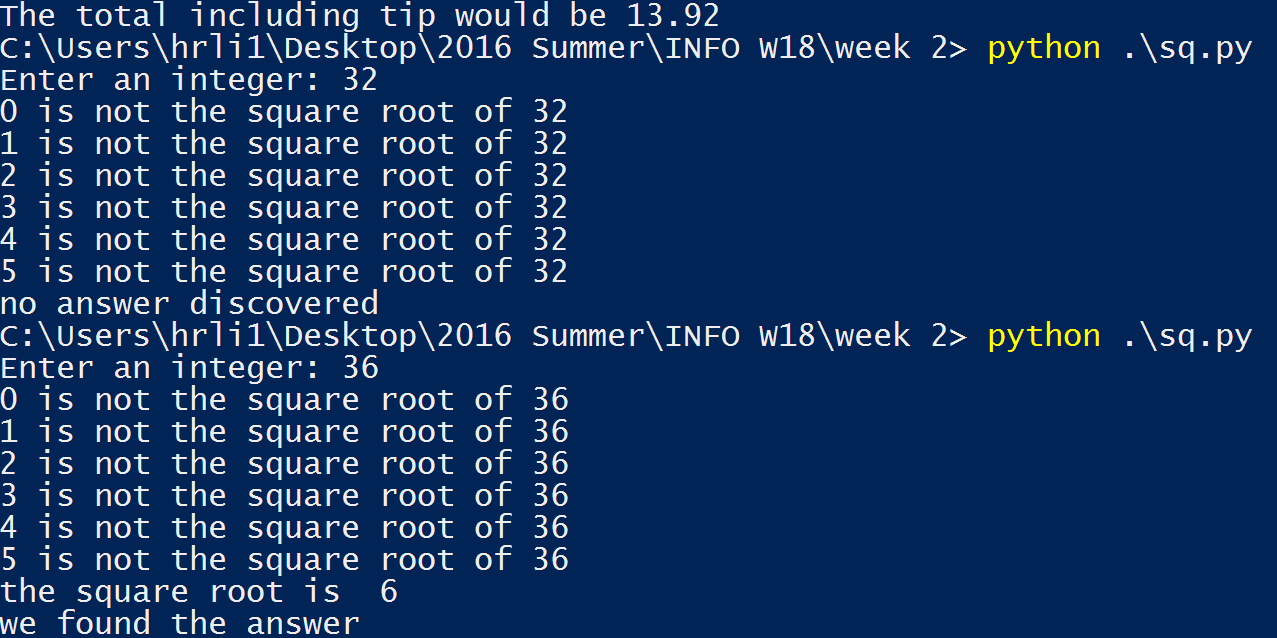
The following two are the same!











Questions:

1. When we say that python is strongly typed, do we mean that we can’t do function or operator overloading in python like in C++?

2. >>> sev\_f = float(7)

>>> sev\_i = int(7)

>>> sev\_f ==sev\_i

True

3. The video about loops is kind of indicating that a variable can be used insied and outside the loop, is it ture for Python? Does Python distiguish local variables and global variables?