print(1/3) #devision returns float even when dividing two ints

# highlight something and use ctrl + / to comment all of that

all0, at, once = 5, 10 , 10

print (all0)

Naming Conventions

Most Python programmers prefer to use standard style conventions   
when naming things:

* Most variables should be **snake case** (underscores between words)
* Most variables should also be **lowercase,** with some exceptions: **CAPITAL\_SNAKE\_CASE** usually refers to constants (e.g. PI = 3.14) **UpperCamelCase** usually refers to a class (more on that later)
* Variables that start and end with two underscores (called "dunder"   
  for double underscore) are supposed to be private or left alone

\_\_no\_touchy\_\_

num\_1 = input("Please input miles ")

num\_2 = float(num\_1) \* 1.60934

print('{} miles equals to {} kilometers' .format(num\_1, num\_2))

######################

num\_3, num\_4 = input('enter 2 numbers: ').split()

num\_3 = int(num\_3)

num\_4 = int(num\_4)

sum\_1 = num\_3 + num\_4

difference = num\_3 - num\_4

print("{} + {} = {}".format(num\_3, num\_4, sum\_1))

print("{} - {} = {}".format(num\_3, num\_4, difference))

import math

print("ceil(4.4) = ", math.ceil(4.4))

print("floor(4.4) = ", math.floor(4.4))

print("fabs(-4.4) = ", math.fabs(-4.4))

print('factorial(4) = ', math.factorial(4))

print("math e", math.e)

print("math.pi", math.pi)

'''

math.fmod -----> reminder of division

math.trunc ------> receive dloat return int

math.pow -------> x^y

math.sqrt

math.log(1000,10)

math.log10(1000)

sin, cos, tan, asin, acos, atan, atan2

atanh,

math.sin(0)

math.degrees(1.555)

math.radian(90)

'''

|  |  |  |
| --- | --- | --- |
| **Code** | **Result** | **Try it** |
| \' | Single Quote | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_escape2) |
| \\ | Backslash | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_backslash) |
| \n | New Line | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_newline) |
| \r | Carriage Return | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_r) |
| \t | Tab | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_t) |
| \b | Backspace | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_b) |

round(4.56569587, 2) = 4.56

X = 1

X is 1 #True

X is 0 #False

False things: number 0 , empty string, and none

Boolean Logics:

animal = input('enter your favourite animal: ')

number = input('input your favourite number: ')

if animal == "" or number == "":

if bool(animal) ^ bool(number): #same as the next line, ^ & | are bitwise

if (animal and not number) or (number and not animal):

Booleans are considered a **numeric** type in Python. This means they’re [numbers](https://realpython.com/python-numbers/) for all intents and purposes. In other words, you can apply arithmetic operations to Booleans, and you can also compare them to numbers:

>>>

>>> True == 1

True

>>> False == 0

True

>>> True + (False / True)

1.0

**Bitwise operators are used to compare (binary) numbers: ( & | ^ ~ << >> )**

**Logical operators: and, or, not**

**Identity operators: is, is not**

**Comparison operators: ==, != , >, <, =<, =>**

**Arithmetic operators: +, -, %, /, \*, \*\*(exponent), //(kharej ghesmat), ^ (xor)**

**Assignment operators:**

**= x = 5 x = 5**

**+= x += 3 x = x + 3**

**-= x -= 3 x = x - 3**

**\*= x \*= 3 x = x \* 3**

**/= x /= 3 x = x / 3**

**%= x %= 3 x = x % 3**

**//= x //= 3 x = x // 3**

**\*\*= x \*\*= 3 x = x \*\* 3**

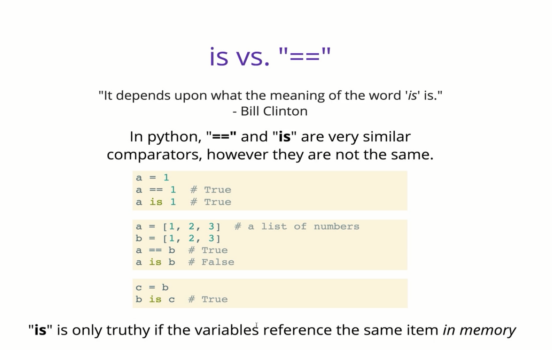
**&= x &= 3 x = x & 3**

**|= x |= 3 x = x | 3**

**^= x ^= 3 x = x ^ 3**

**>>= x >>= 3 x = x >> 3**

**<<= x <<= 3 x = x << 3**

****

**“is” checks whether they are stored in the same place in the memory**

**“==” checks values**

>>> x = 13

>>> y = 13

>>> x == y

True

>>> a = [1,2]

>>> b = [1,2]

>>> a == b

True

>>> a is b

False

>>> clone = a

>>> clone is a

True