# --NUMBERS—

## >>>>>Types of Numbers<<<<<

## integers (1, 3, 5, 99.. etc.)
## floating's (1.5, 2.6, 3.9 ,44.55.. etc.)
## complex numbers (x = a + bj)
##Boolean (True or False)

```
example_complex = 1+2j
print(example_complex)
```

### >>Know the Types of numbers:

```
print(type(4)) #integer
print(type(4.5)) #floating
print(type(1+2j)) #complex
```

### >> Type conversion:

```
x = input(" x: ") #Takes a string as input
y = int(x)+1 # "x" is converted into integer and added with an integer
print(f"X: {x} , Y: {y}") #Printing the values
```

# >>>>>Mathematical Operations<<<<<

## >>Standard arithmetic operations:

```
print(2+2) #addition
print(4-2) #subtraction
print(5*6) #multiplication
print(8/5) #classic division (returns a float)
print(17//3) #floor division (discards the fractional part)
print(17 % 3) #remainder
print(5**2) #five to the power two
print(2**7) #two to the power seven
```

## >> Operator Precedence:

# #python follows normal math precedence:
# # brackets ()
# # power (**)
# # division (/)
# # multiplication (*)
# # addition & subtraction (+ -)

```
print(20+3*4)
print(20/5*4+5-1)
print(2+10*10+3)
print((2+10)*(10+3))
print(50-5*6)
print((50-5*6)/4)
print(5*3+2)
print(4 * 3.75 - 1) #mixing of floating & integer operator
print(456/0) # ZeroDivisionError
```

## >>Functions with Numbers:

#built-in functions:

```
print(round(2.9)) # round up/down the value
print(abs(-2.9)) #absolute value
print( pow(2, 4)) #two to the power four
print(min(1,2,3,4,5,6,7,8,9)) #Minimum
print(max(1,2,3,4,5,6,7,8,9)) #Maximum
print( divmod(10, 3)) #(division, remainder)
```

#For complex mathematical operations we import the "math" module
# a module is like a separate file like a python file
#"math" module has lots of complex mathematical functions we can use

```
import math #importing math module
```

#Now we can use dot(.) notation to use the functions/methods
#all the math module functions: google (python3 math module)

```
print(math.ceil(2.2)) #output is 3 (ceiling the floating number)
print(math.floor(2.5)) #output is 2 (flooring the floating number)
print(math.sqrt(16)) #square root of a number
```

# >>>>>Assigning Variables<<<<<

#Example 01:

```
a = 10
 print(a)
b = 20
print(b)
c = a+b
print(c)
x = 20
print(x)
del x   #delete the entire variable + value
print(x)  #x is not defined (NameError)
```

#Example 02:

```
width = 20
height = 5 * 9
print(width * height)
```

#Example 03:

```
tax = 12.5 / 100
price = 100.50
print(price * tax)
```

**>>Assign multiple variables at a time:**

```
a = b = c = 10
print(a)
print(b)
print(c)

x, y, z = 1, 2, 3
print(x)
print(y)
 print(z)
```

**>>Augmented assignment operator:**

```
age = 32
print(age)
age = age + 1
print(age)
age+=1
print(age)
age+=4
print(age)
age-=4
print(age)
age*=2
print(age)
age//=2
print(age)
```

**>> reassigning the value of a variable:**

```
a = 5
print(a)
a = 20
print(a)
a = a+a
print(a)
```

# >>>>>Boolean Data Type<<<<<

# A Boolean expression is a logical statement that is either TRUE or FALSE
#Falsy Boolean values in Python:
# "" (empty string)
# 0 (number zero)
#None (absence of a value)
#no arguments ()
#anything else is True

```
print(bool()) #no arguments = False
print(bool("")) #empty string = False
print(bool(0)) #Number zero = False
print(bool(None)) #None = False
print(bool(" ")) #Space = True
print(bool(5)) #any integer (without number zero) = True
print(bool(-1)) #any integer (even negative) = True
print(bool("a")) #Any letters = True
print(bool("_")) #any characters = True
print(bool("BlaBlaBla")) #any strings = True
```

#a true value is converted into (1)
#a false value is converted into (0)

```
print(int(bool("A"))) #True = 1
print(int(bool(""))) #False = 0
```

## >>>>>Number system representation<<<<<

**>> Decimal to others:**

```
print(bin(5)) #binary
print(hex(10)) #hexadecimal
print(oct(20)) #octal
```

**>> Others to decimal:**

# syntax is (number type, base)
# binary base is 2
# octal base is 8
# hexadecimal base is 16

```
print(int("0b101", 2)) #binary to decimal (integer)
print(int("0o24", 8)) #octal to decimal (integer)
print(int("0xa", 16)) #hexadecimal to decimal (integer)
```

--END --