Introduction to Computer Programming

Week 2 More on variables types and conditional

Part and variabes

Exercise 1: Overwriting Variable

1

In [1]:

```
x = 5
x = x + 7
x = x -3
print(x)
```

```
9
```

2

In [2]:

```
y = 2
y = y - 2
y += 2
print(y)
```

```
2
```

As you can see if x += 4 is equivalent to writing x = x + 4. The principles is the same for -=

3

In [3]:

```
x = 5
x += 7
x -= 3
print(x)
```

```
9
```

4

In [4]:

```
y = 2
y += y
print(y)
y = 2
y *= 2
print("You could also do y *= 2 to give you the same results:", y)
```

```
4
You could also do y *= 2 to give you the same results: 4
```

5

In [5]:

```
z += 1
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-5-b1ebc95d97f2> in <module>
----> 1 z += 1

NameError: name 'z' is not defined
```

As you can see since the variable z is not defined we can't add to it

6

In [39]:

```
x = 3
x *= 10 # to multiply
x = 15
x /= 5 # to divide
```

7

In [40]:

```
print(8/2*(2+2))
```

```
16.0
```

Python uses linear notation which is computed using the traditional order of operation. However, the operation is executed from left to right. Hence why the result is 16 and not 1. One could assume that the in the equation $8 \div 2(2+2)$ the multiplication might take place before the division. Moreover, if your intent is to have the equation interpreted as $\frac{8}{2*(2+2)}$ then in linear notation you would need to add extra parenthesis to make it explicit.

Exercise 2

-

In [6]:

```
cat name = "Whiskers"
```

```
  File "<ipython-input-6-bbf242b3df73>", line 1
    cat name = "Whiskers"
           ^
SyntaxError: invalid syntax
```

In [7]:

```
cat-name = "Whiskers"
```

```
  File "<ipython-input-7-5c1c44bde901>", line 1
    cat-name = "Whiskers"
    ^
SyntaxError: cannot assign to operator
```

In [8]:

```
1catname = "Whiskers"
```

```
  File "<ipython-input-8-8e68bb9f2627>", line 1
    1catname = "Whiskers"
     ^
SyntaxError: invalid syntax
```

In [9]:

```
$catname = "Whiskers"
```

```
  File "<ipython-input-9-66ce12627a9f>", line 1
    $catname = "Whiskers"
    ^
SyntaxError: invalid syntax
```

In [10]:

```
catname = "Whiskers"
```

3

In [11]:

```
CATNAME = "Felix"
print("The variable CATNAME contains:", CATNAME)
print("The variable catname contains:", catname)
```

```
The variable CATNAME contains: Felix
The variable catname contains: Whiskers
```

As you can see the variable names are case sensitive so "catname was not overwritten"

5

In [12]:

```
len = 10
len()
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-12-3638ca34007c> in <module>
      1 len = 10
----> 2 len()

TypeError: 'int' object is not callable
```

We have redefined len to an int, this is allowed as length is not restricted we now need to perform a delted operation to len to use the function again

In [13]:

```
del len
len([1,2,3]) # we can now use the function again
```

Out[13]:

```
3
```

Exercise 3

1

In [14]:

```
a = 5
b = 1.618
c = 1 + 2j
d = 9j
e = 0
f = 2.0 + 0j
ListOfVar = [a , b, c, d, e, f]
ListOfVarStr = ['a' , 'b', 'c', 'd', 'e', 'f']
list(map(lambda x,y: (y + " is of type", type(x)), ListOfVar, ListOfVarStr))
```

Out[14]:

```
[('a is of type', int),
 ('b is of type', float),
 ('c is of type', complex),
 ('d is of type', complex),
 ('e is of type', int),
 ('f is of type', complex)]
```

2

In [42]:

```
z = 1 / 2
y = 5 * 20
x = 5.0 * 20.0
w = (2.0+1j) * (2.0-1j)
ListOfVar2 = [z, y, x, w]
ListOfVar2Str = ['z', 'y', 'x', 'w']
list(map(lambda x,y: (y + " is of type", type(x)), ListOfVar2, ListOfVar2Str))
```

Out[42]:

```
[('z is of type', float),
 ('y is of type', int),
 ('x is of type', float),
 ('w is of type', complex)]
```

In [16]:

```
list(map(lambda x,y: print("Is Var",y,"int ?",isinstance(x, int)), ListOfVar2, ListOfVa
r2Str))
```

```
Is Var z int ? False
Is Var y int ? True
Is Var x int ? False
Is Var w int ? False
```

Out[16]:

```
[None, None, None, None]
```

3

In [17]:

```
list(map(lambda x,y: print("Type Casting to Int the variable",y,"now becomes",int(x)),
ListOfVar2, ListOfVar2Str))
```

```
Type Casting to Int the variable z now becomes 0
Type Casting to Int the variable y now becomes 100
Type Casting to Int the variable x now becomes 100

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-17-7129cd31058d> in <module>
----> 1 list(map(lambda x,y: print("Type Casting to Int the variable",y,"n
ow becomes",int(x)), ListOfVar2, ListOfVar2Str))

<ipython-input-17-7129cd31058d> in <lambda>(x, y)
----> 1 list(map(lambda x,y: print("Type Casting to Int the variable",y,"n
ow becomes",int(x)), ListOfVar2, ListOfVar2Str))

TypeError: can't convert complex to int
```

As you can see we cannot type cast complex to integer. You could argue that a complex number is a list of two floats (or integer) each representing the real and imaginary part

In [18]:

```
list(map(lambda x,y: print("Type Casting to float the variable",y,"now becomes",float(x
)), ListOfVar2, ListOfVar2Str))
```

```
Type Casting to float the variable z now becomes 0.5
Type Casting to float the variable y now becomes 100.0
Type Casting to float the variable x now becomes 100.0
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-18-4b1e481db4a5> in <module>
----> 1 list(map(lambda x,y: print("Type Casting to float the variable",y,
"now becomes",float(x)), ListOfVar2, ListOfVar2Str))

<ipython-input-18-4b1e481db4a5> in <lambda>(x, y)
----> 1 list(map(lambda x,y: print("Type Casting to float the variable",y,
"now becomes",float(x)), ListOfVar2, ListOfVar2Str))

TypeError: can't convert complex to float
```

Similar to the previous type cast we can't convert complex to float

In [19]:

```
list(map(lambda x,y: print("Type Casting to float the variable",y,"now becomes",complex
(x)), ListOfVar2, ListOfVar2Str))
```

```
Type Casting to float the variable z now becomes (0.5+0j)
Type Casting to float the variable y now becomes (100+0j)
Type Casting to float the variable x now becomes (100+0j)
Type Casting to float the variable w now becomes (5+0j)
```

Out[19]:

```
[None, None, None, None]
```

As you can see it is possible to convert inteeers and floats to complex types, the imaginary part is 0.

In [20]:

```
list(map(lambda x,y: print("Type Casting to float the variable",y,"now becomes",bool(x
)), ListOfVar2, ListOfVar2Str))
```

```
Type Casting to float the variable z now becomes True
Type Casting to float the variable y now becomes True
Type Casting to float the variable x now becomes True
Type Casting to float the variable w now becomes True
```

Out[20]:

```
[None, None, None, None]
```

Values not equal to zero get type cast as boolean True (False in the zero case)

In [21]:

```
list(map(lambda x,y: print("Type Casting to float the variable",y,"now becomes",str(x
)), ListOfVar2, ListOfVar2Str))
```

```
Type Casting to float the variable z now becomes 0.5
Type Casting to float the variable y now becomes 100
Type Casting to float the variable x now becomes 100.0
Type Casting to float the variable w now becomes (5+0j)
```

Out[21]:

```
[None, None, None, None]
```

4

In [22]:

```
e = 1.0 + 5
f = 3j + 4.6
g = 4 * 2.5
h = 4 * 2.7
ListOfVar3 = [e, f, g, h]
ListOfVarStr3 = ['e', 'f', 'g', 'h']
list(map(lambda x,y: (y + " is of type", type(x)), ListOfVar3, ListOfVarStr3))
```

Out[22]:

```
[('e is of type', float),
 ('f is of type', complex),
 ('g is of type', float),
 ('h is of type', float)]
```

Python takes the most encompassing type to avoid information loss similar to number set memebership: $\mathbb{N} \in \mathbb{Z} \in \mathbb{Q} \in \mathbb{R} \in \mathbb{C}$

5

These functions return the octal, hex, and binary represnetation of integer respectively

In [ ]:

```

```

Exercise 4 Boolean Operators

1, 2

In [23]:

```
A, B, C = 5, 10, 0
```

In [24]:

```
print(A < B < C)
print((A <= B) and (A > C))
print((2*A) == B)
print(A and B and C)
print((A != B) or C)
print((A > 0) and (B > 0) or not (C == 0))
```

```
False
True
True
0
True
True
```

Part 2 List and Strings

Exercise 5 Further Strings

1

In [25]:

```
str(4)
str(3.14159)
str(2+9j)
```

Out[25]:

```
'(2+9j)'
```

2

In [26]:

```
str()
```

Out[26]:

```
''
```

As you can see this returns the empty string

3

```
In [27]:
```

```
a = "The meaning of life is"
a = a + " " + 42
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-27-6269b8d035c4> in <module>
      1 a = "The meaning of life is"
----> 2 a = a + " " + 42

TypeError: can only concatenate str (not "int") to str
```

```
In [28]:
```

```
# Let's fix this by converting 42 to a string
a = "The meaning of life is"
a = a + " " + str(42)
print(a)
```

```
The meaning of life is 42
```

4

```
In [29]:
```

```
print(a[len(a) -1])
```

```
2
```

Python arrays start at index 0 this means that to access the last element we need to get the length of the string and substract 1. Another Python-esque approach is to peruse a list in reverse starting from index -1 to obtain the last element. This is demonstrated below

```
In [30]:
```

```
print(a[-1])
```

```
2
```

5

```
In [31]:
```

```
a.replace("42","41.99999")
```

```
Out[31]:
```

```
'The meaning of life is 41.99999'
```

replace() takes a given sequence in the data and its repalcement. Please refer to the docs for more information

Part 3 Control Flow

Exercise 7 Conditionals

1, 2, 3, 4, 5

In [32]:

```
AgeOfWhiskers = 5
if AgeOfWhiskers <= 1.5:
    print("Part 1 Whiskers is a kitten")
```

Since Whiskers' age is greater than 1.5 he isn't a kitten

In [33]:

```
AgeOfWhiskers = 5
if AgeOfWhiskers <= 1.5:
    print("Part 2 Whiskers is a kitten")
else:
    print("Part 2 Whiskers is an Adult Cat")
```

```
Part 2 Whiskers is an Adult Cat
```

In [34]:

```
AgeOfWhiskers = 5
if AgeOfWhiskers <= 1.5:
    print("Part 3 Whiskers is a kitten")
elif AgeOfWhiskers > 12:
    print("Part 3 Whiskers is an old cat")
else:
    print("Part 3 Whiskers is an Adult Cat")
```

```
Part 3 Whiskers is an Adult Cat
```

In [35]:

```
AgeOfWhiskers = 5
if AgeOfWhiskers <= 1.5:
    print("Part 3 Whiskers is a kitten")
elif AgeOfWhiskers > 12:
    print("Part 3 Whiskers is an old cat")
else:
    print("Part 3 Whiskers is an Adult Cat")
```

```
Part 3 Whiskers is an Adult Cat
```

In [43]:

```
AgeOfWhiskers = 5
CatAge =  6
if AgeOfWhiskers <= CatAge:
    print("Part 3 Whiskers is a kitten")
elif AgeOfWhiskers > CatAge*8:
    print("Part 3 Whiskers is an old cat")
else:
    print("Part 3 Whiskers is an Adult Cat")
```

```
Part 3 Whiskers is a kitten
```

6

In [37]:

```
# First let's save the center and radius of each circles
import math

CirclesX = [0, 2, -5]
CirclesY = [0, 1, 0]
CirclesRad = [5, 2, 3]

# I am going to defined a function that takes coordinates x and y which we will then co
mpute the distance from the from the center and check if it is less or equal than the r
adius
def isItinCircle(x,y):
    TempX = [a - x for a in CirclesX]
    TempY = [b - y for b in CirclesY]
    TempRad = list(map(lambda a, b, c: (math.sqrt(a**2 + b**2) - c) <= 0, TempX, TempY,
CirclesRad))
    for i in range(0, len(CirclesX)):
        if TempRad[i]:
            print("This point is in Range of Circle C",i+1)
```

IMPORTANT: I USED A SHORTCUT TO COMPUTE THE DISTANCE FROM THE CENTERS BUT YOU CAN DO IT MANUALLY OR IN A FOR LOOP. IN EACH IF STATEMENT YOU COULD COMPUTE THE CHECK FOR THE DISTANCE BEING LESS THAN THE RADIUS.

FINALLY I USED A LOOP WITH IF IN THIS EXERCISE USING IF ELSE AND ELIF IS REQUIRED.

As Xanthe mentioned you compute if a point is within the intersections to be more efficient and use less conditional statements

In [38]:

```
isItinCircle(-6,1)
```

```
This point is in Range of Circle C 3
```

In [ ]:

```
```