Clase1

Unknown Author

July 28, 2016

Part I

Introduction to Python

1 Variable types

```
In [29]:  # This  is  a  comment
          a=5        # integer
          b=4.0      # float
          c=True     # boolean
          d='Hola'   # Character   string
```

2 Arithmetic operators

```
In [34]:  s=a+b # sum
          r=a-b # subtraction
          m=a*b # multiplication
          d=a/b # division
```

3 Different output formats

```
In [36]:  print  s # most   symple    output
          print  "Hello" # print  a   character   string
          print  "Hello"+"\t"+"the sum result is :"+str(s)+"\n" # print  using  string concatena
          print  "The sum  result is: %.8f" %(s)   # print using space reservation to variables
          print  "The sum  result is: %.2e" %(f)   # print using space reservation to variables
          print """print for texts with several
                  lines, in this case we dont
                  employed return command
              """
          9.0
          Hello
          Hello   the sum result is :9.0

          The sum  result is: 9.00000000
          The sum  result is: 1.29e+02
          print for texts with several
```

```
                lines, in this case we dont
                employed return command
```

Different format commant in python are given:

4 Integer division

```
          a=5
          b=4
In [37]:  c=a/b
          c
          1
```
Out [37]:
¿What is the problem in the result? ¿How we can resolve it?

4.1 Solution 1

```
          a=5
          b=4
In [38]:  c=a/(1.0*b)
          c
          1.25
```
Out [38]:

4.2 Solution 2

```
          from __future__ import division
          a=5
In [39]:  b=4
          c=a/b
          c
          1.25
```
Out [39]:
¿What mean the word from in python?

In python the reserved word "from" is employed to import functions or objects contained in different packages.

5 Other Basic operators

```
          a=10.0%4.0 # rest   of the   integer division between   two   numbers
In [56]:  p=2**4      # power  operator
```

6 Other variables types in python

```
          c=  1.0+2.0j #complex number
In [57]:  l=  [1.0,2.0,3.0,"hello"]  # var type list
          t=  (1.0,2.0,3.0,"hello")  # var type tuple
```

¿What is the difference between lists and the tuples ?

The main difference between tuples and lists in python is that the tuples are immutable.

6.1 Manipulation lists and tuples

Lists

In [59]:
```
l1=[1.0,2.0,True,"blue"]
l1[0]    # to access to   the   element   of index    0   in the   list
l1[1]="change" # to   assigne   new   valued   to   element   of  index    1  in the   list
print l1[1]
```
```
change
```

In this point is important highlight that the index in the lists and the tuples begin in 0

Tuples

In [62]:
```
t1=(1.0,2.0,True,"blue")
t1[0]    # to access to   the   element   of index    0   in the   tuple
t1[1]= "Change"
```

```
---------------------------------------------------------------
-------------
    TypeError                                Traceback (most recent
call last)


        <ipython-input-62-a1a2bae551ba> in <module>()
          1 t1=(1.0,2.0,True,"blue")
          2 t1[0]   # to access to   the   element   of index    0   in the
tuple
    ----> 3 t1[1]= "Change"



        TypeError: 'tuple' object does not support item assignment
```

You can see a error in the code, because the tuples don't can change the values

7 Manipulation vars in python

We have a lot of function to manipulate vars in phython, here we shown some of its

In [65]:
```
a=1.2345
b="red"
type(a) #  this    function   return   the    type   of   var   a
print type(a),type(b)
```
```
<type 'float'> <type 'str'>
```

We also can change the type of an one var

In [67]:
```
a=1.23456
print type(a)
b=str(a) # str() function   change   float   and   integer    vars   to   string
print type(b)
```

```
<type 'float'>
<type 'str'>
```

Also we can change string to number

In [73]:
```
a="10.23456"
b=float(a)     #  float()   function   change   string  to   float
print b,type(b)
c=int(b)     # int() function change  float to  integer
print c,type(c)
```
```
10.23456 <type 'float'>
10 <type 'int'>
```

In [73]:

In []: