dia2

May 28, 2024

1 DATOS BOOLEANOS

[1]:
```
bool_1=bool(1)
bool_2=bool(0)
bool_3=bool(None)
print(bool_1,bool_2,bool_3)
```

```
True False False
```

2 CASTING

[ ]:
```
# int()
# str()
# bool()
# float()
# list()
# tuple()
# set()
```

3 DATOS TIPO STRING (CADENA DE CARACTERES)

[2]:
```
str_="Hello World"
print(type(str))
isinstance(str_,str)
```

```
<class 'type'>
```

[2]:
```
True
```

[3]:
```
str_[0]
```

[3]:
```
'H'
```

[4]:
```
str_[-1]
```

[4]:
```
'd'
```

[5]:
```
# str slicing
print(str_[2:5])  #forward
print(str_[-5:]) #backward
str_[::-1] #reverse
```

```
llo
World
```

[5]: 'dlroW olleH'

4 CONCATENACION

[6]:
```
str_1="first"
str_2="second"
print(str_1+" "+str_2)
print(str_*3) #muliple
print(str_1,str_2,sep=":") #sep
print(str_1,str_2,end=" ,") #end
```

```
first second
Hello WorldHello WorldHello World
first:second
first second ,
```

[7]:
```
# delete string
del str_
print(str_)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-38151ea13c07> in <cell line: 3>()
      1 # delete string
      2 del str_
----> 3 print(str_)

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

5 PARTICION

[8]:
```
str_part="Hello, how are your?, are they?"
print(str_part.partition("are")) #particion de la sentencia
```

```
('Hello, how ', 'are', ' your?, are they?')
```

[9]:
```
print(str_part.rpartition("are")) #ultima
```

```
('Hello, how are your?, ', 'are', ' they?')
```

6 FUNCIONES

[10]:
```
str_strip="**********     hi    *******"
print(str_strip.strip("*"),end="")
```

```
     hi
```

[11]:
```
print(str_strip.rstrip("*"),end="")
```

```
**********     hi
```

[12]:
```
print(str_strip.lstrip("*"),end="")
```

```
     hi    *******
```

[13]:
```
print(str_part.count("are"))  #Cuenta
```

```
2
```

[14]:
```
str_ex="welcome everyone"
print(str_ex.split())
```

```
['welcome', 'everyone']
```

[15]:
```
str_ex.find("everyone")
```

[15]: 8

[16]:
```
str_ex.replace("everyone",",hi")
```

[16]: 'welcome ,hi'

[17]:
```
str_ex.index("welcome")
```

[17]: 0

[18]:
```
num_="10"
print(num_.isnumeric(),
num_.isalnum(),
num_.isdecimal(),
num_.isdigit(),
num_.islower(),
num_.isupper(),
num_.isspace(),
num_.isascii())
```

```
True True True True False False False True
```

7 LISTAS

[19]:
```
list_1=[1,2,3,4,5]
list_type=["hi",5.8,7,[5,7,8,9],(7,8,2)]
print(list_type[2])
type(list_type)
print(list_1[:5])
print(list_1[1:4])
print(list_1[-2:])
print(list_1[:-3])
```

```
7
[1, 2, 3, 4, 5]
[2, 3, 4]
[4, 5]
[1, 2]
```

8 LISTAS FUNCIONALES

[20]:
```
print(list_1.append(10))
print(list_1.insert(0,"hi"))
print(list_1.pop())   #remove last element
print(list_1.pop(5))
del list_1 #.clear()
print(list_1)
```

```
None
None
10
5
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-20-748c9a1ce3cb> in <cell line: 6>()
      4 print(list_1.pop(5))
      5 del list_1 #.clear()
----> 6 print(list_1)

NameError: name 'list_1' is not defined
```

9 BUCLE Y AFILIACIÓN

[21]:
```
for i in list_type:
    print(i)
```

```
hi
5.8
7
[5, 7, 8, 9]
(7, 8, 2)
```

[22]:
```
for i in enumerate(list_type):
    print(i)
```

```
(0, 'hi')
(1, 5.8)
(2, 7)
(3, [5, 7, 8, 9])
(4, (7, 8, 2))
```

[23]:
```
# revertir & ordenar & ordenado
sort_reverse=[3,6,4,57,8,2]
sort_reverse.sort()
print(sort_reverse)
print(sorted(sort_reverse))
sort_reverse.sort(reverse=True)
print(sort_reverse)
```

```
[2, 3, 4, 6, 8, 57]
[2, 3, 4, 6, 8, 57]
[57, 8, 6, 4, 3, 2]
```

[24]:
```
any(sort_reverse)
```

[24]: True

[25]:
```
all(sort_reverse)
```

[25]: True

10 TUPLA

[27]:
```
(1,) #tupla
```

[27]: (1,)

[28]:
```
tupla_=(1,2.3,"hi",1)
print(tupla_.index(2.3))
tupla_.count(1)
```

```
1
```

[28]: 2

[30]:
```
tupla_a=(1,4,6)
tupla_b=(7,4,8)
print(tupla_a+tupla_b)
tupla_x=tupla_a+tupla_b
```

```
(1, 4, 6, 7, 4, 8)
```

[31]:
```
tupla_a*3
```

[31]: (1, 4, 6, 1, 4, 6, 1, 4, 6)

[32]:
```
for i in tupla_a:
    print(i)
```

```
1
4
6
```

[33]:
```
# Asterisco
ex_tupla=(1,2,3,4,5,4,8,0)
(x,*y,z)=ex_tupla
print(x)
print(y)
print(z)
```

```
1
[2, 3, 4, 5, 4, 8]
0
```

[34]:
```
tupla_x[:]
```

[34]: (1, 4, 6, 7, 4, 8)

[35]:
```
print(tupla_x[-5:-2])
print(tupla_x[1:3])
tupla_x[::-1]
```

```
(4, 6, 7)
(4, 6)
```

[35]: (8, 4, 7, 6, 4, 1)

[36]:
```
# tupla es immutable (no cambiable)
# uno del camino se actualiza
tupla_1=(88,)
tupla_x+=tupla_1
print(tupla_x)
```

```
(1, 4, 6, 7, 4, 8, 88)
```

11 SET

12 set es un tipo de datos mutable no duplicado

[37]:
```
set_={1,5,8,4,5,4} #Se eliminan los duplicados
set_
```

[37]: {1, 4, 5, 8}

[38]:
```
#funciones
set_a={1,8,5,7,2,6}
set_={1,5,8,4,5,4}
```

[39]:
```
print(set_.add(5),
set_.difference(set_a),
set_.intersection(set_a),
set_.union(set_a),
set_.symmetric_difference(set_a),
set_.pop(),
set_.update([5,8,70]),
set_)
```

```
None {4} {8, 1, 5} {1, 2, 4, 5, 6, 7, 8} {2, 4, 6, 7} 8 None {1, 4, 5, 70, 8}
```

[40]:
```
list(enumerate(set_))
```

[40]: [(0, 1), (1, 4), (2, 5), (3, 70), (4, 8)]

13 DICT

[41]:
```
dict_={"A":1,"B":2,"C":3}
dict_
```

[41]: {'A': 1, 'B': 2, 'C': 3}

[42]:
```
dict_.items()
```

[42]:
```
dict_items([('A', 1), ('B', 2), ('C', 3)])
```

[43]:
```
dict_.values()
```

[43]:
```
dict_values([1, 2, 3])
```

[44]:
```
dict_.keys()
```

[44]:
```
dict_keys(['A', 'B', 'C'])
```

[45]:
```
a=[1,4,7,3]
b={2,5,3,7}
dict_.fromkeys(a,b)
```

[45]:
```
{1: {2, 3, 5, 7}, 4: {2, 3, 5, 7}, 7: {2, 3, 5, 7}, 3: {2, 3, 5, 7}}
```

[ ]:
```
# dict_.[]
# dict_.get()           #Accesando a los valores
```

[46]:
```
dict_.pop("A")
```

[46]:
```
1
```

[ ]:
```
# (*)args & (**)kwargs
# *args -> longitud variable Argumentos sin palabra clave (pasados como una␣
 ↪tupla)
# **kwargs -> longitud variable Palabra clave Argumentos (pasados como␣
 ↪diccionario)
```