

Examen rápido No. 1

Inteligencia Artificial

Julio Waissman Vilanova

16 de agosto de 2017

1. Tipos mutables

1. Mira el siguiente código y ejecutalo:

```
# Vamos a generar 3 listas que parecerían iguales
lista_1 = [1, 2, 3, "toto", ["a", "b", "c"]]
lista_2 = lista_1
lista_3 = lista_1[:]
```

```
lista_2[0] = "XXX"
print(lista_2)
```

```
lista_3[-1][0] = 1000
print(lista_3)
```

Ahora responde (sin revisar) lo que crees que debería salir con

```
print(lista_1)
print(lista_2)
```

```
# Ejemplo:
# print(lista_3) --> [1, 2, 3, 'toto', [1000, 'b', 'c']]
```

Una vez que lo contestaste, ejecuta las operaciones y explica el porqué de los resultados.

2. Revisa el siguiente código

```
class coordenadas:
    def __init__(self, x=0, y=0):
        "Inicializa un objet coordenadas"
        self.x = x
        self.y = y

    def __str__(self):
        return "\n\tCoordenada x = {}\n\tCoordenada y = {}".format(self.x, self.y)
```

```
a = coordenadas(3, 4)
b = a
b.x = 100
print("b = " + str(b))
```

Y ahora escribe que crees que debería salir (sin revisar) con

```
print("a = " + str(a))
```

Una vez que lo contestaste, revisa el resultado y explícalo brevemente.

2. Listas y diccionarios

1. Escribe, en una sola línea, una expresión que genere todos los números enteros que se encuentran entre 1 y 1000 que sean divisibles por 2, 3, 5 y 7 al mismo tiempo. Aprovecha de las ventajas para crear listas de:

```
[expresión for elemento in lista],  
[expresión for elemento in lista if condición]
```

2. Escribe una función que reciba una lista de elementos (letras, números, lo que sea), cuente la ocurrencia de cada elemento en la lista y la devuelva en forma de diccionario e imprima un histograma de ocurrencias, por ejemplo:

```
>>> d = FuncionEjemplo( [1,'a',1, 13, 'hola', 'a', 1, 1, 'a', 1], Imprime = True)
```

```
1          *****   (5 -> 50%)  
'a'        ***       (3 -> 30%)  
13         *         (1 -> 10%)  
'hola'     *         (1 -> 10%)
```

```
>>> print d  
{1:5, 'a':3, 13:1, 'hola':1}
```

3. Escribe una función que modifique un diccionario y regrese el diccionario modificado y una copia del original, donde cada entrada del diccionario sea una lista de valores, ejemplo de la función:

```
>>> dic1 = {'Pepe':[12, 'enero', 1980], 'Carolina':[15,'mayo',1975], 'Paco':[10,'nov',1970]}  
>>> dic2 = fundicos(dic1, 'Pepe', 1, 'febrero')  
>>> print dic1  
{'Pepe':[12, 'enero', 1980], 'Carolina':[15,'mayo',1975], 'Paco':[10,'nov',1970]}  
>>> print dic2  
{'Pepe':[12, 'febrero', 1980], 'Carolina':[15,'mayo',1975], 'Paco':[10,'nov',1970]}
```

3. Funciones y clases

1. Escribe una función `fun1` que reciba un número n y calcule el número primo inmediatamente superior. Escribe una función `fun2` que reciba como argumento un número y una función, y devuelva una lista con la evaluación de la función desde 1 hasta n . Prueba la función con `fun1` y con `math.sqrt`.
2. Escribe una función, lo más compacta posible, que escoja entre los 3 patrones ascii a continuación, e imprima en pantalla el deseado, pero de la dimensión n deseada ($n \geq 4$, toma en cuenta que para algunos valores de n habrá algún(os) patrones que no se puedan hacer).

*	++++	00000000
**	++++	000 000
***	++++	00 00
****	++++	0 0
*****	++++	0 0
*****	++++	00 00
*****	++++	000 000
*****	++++	00000000

3. Diseña una clase Matriz e implementa con sobrecarga la suma de matrices, la multiplicación de matrices y la multiplicación por un escalar, eliminar columna y eliminar fila. Como inicialización de un objeto es necesario conocer n y m (en caso de no proporcionarlos la matriz tendrá una dimensión de 1×1). Igualmente, de no especificarse todos los elementos se inicializan a 0, a menos que exista un tipo espacial ('unos' o 'diag' por el momento). Programa la representación visual de la matriz. Ten en cuenta también el manejo de errores. Por ejemplo, para su uso:

```
>>> A = Matriz(n=3, m=4)
>>> print A
0 0 0 0
0 0 0 0
0 0 0 0
>>> A = A.quitafile(2)
>>> print A
0 0 0 0
0 0 0 0
>>> B = Matriz(4,4,'diag')
>>> print B
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
>>> C = Matriz(4,1,'unos')
>>> print C
1
1
1
1
>>> D = 3*B*C
>>> print D
3
3
3
3
>>> E = 3*B + C
error "Si no son de la misma dimensión las matrices no se pueden sumar"
```