

Taller 1

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 7-feb-2020 11:59 PM

****[Monica Alejandra Robayo Gonzalez]****

[monica.robayo@urosario.edu.co]

Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso. Sugiero una estructura similar a la del repositorio del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: `mcpp_taller1_santiago_matallana`
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
 1. Descárguelo en PDF. Esto puede implicar instalar LaTeX en su computador. Resuélvalo por su cuenta, por favor. Recuerde: Google es su amigo.
 2. Suba los dos archivos (.pdf y .ipynb) a su repositorio en GitHub antes de la fecha y hora límites. Asegúrese de que Daniel sea "colaborador" de su repositorio y de que los dos archivos queden en su repositorio, en la nube (no solo en su computador). No lo deje para última hora. Talleres subidos después de la fecha y hora límites no serán valorados, como tampoco lo serán si son remitidos vía e-mail.

(Todos los ejercicios tienen el mismo valor.)

1. Zelle, sección 1.10 (p. 17):

- "Multiple Choice", Ejercicios # 1-10.
- "Programming Exercises", Ejercicio # 1.

1. Multiple Choice , Ejercicios 1-10

In []:

```
1= b
2= d
3= d
4= a
5= b
6= b
7= c
8= b
9= a
10= d
```

2. Programming Exercises", Ejercicio # 1.

a) ("hello, world!")

In [1]:

```
print ("hello, world!")
```

hello, world!

b) ("hello", "world!")

In [2]:

```
print ("hello", "world!")
```

hello world!

c) (3)

In [3]:

```
print (3)
```

3

d) (3.0)

In [4]:

```
print (3.0)
```

3.0

e) (2+3)

In [5]:

```
print (2+3)
```

5

f) (2.0+3.0)

In [7]:

```
print (2.0+3.0)
```

5.0

g) ("2" + "3")

In [8]:

```
print ("2" + "3")
```

23

h) ("2 + 3) =", 2 + 3)

In [9]:

```
print ("2 + 3) =", 2 + 3)
```

2 + 3) = 5

i) (2 * 3)

In [10]:

```
print (2 * 3)
```

6

j) (2 ** 3)

In [11]:

```
print (2 ** 3)
```

8

k) (2/3)

In [14]:

```
print (2/3)
```

0.6666666666666666

En *computer science* son comunes los ejercicios denominados "pensar como un computador". Con estos usted evalúa si está comprendiendo el material, siempre y cuando no utilice un computador para correr el código del enunciado. Siempre que vea un ejercicio marcado con la etiqueta "pensar como un computador", use papel y lápiz o incluso una calculadora si es necesario para descifrar la respuesta, pero nunca ejecute el código en computador.

2. [Pensar como un computador] ¿Cuál es el valor de w después de ejecutar el siguiente código?

$x = 7$ $y = 5.0$ $z = 10.0$ $w = x \% 2 + y / z + z + y / (z + z)$

In [15]:

```
x=7  
y=5.0  
z=10.0
```

In [17]:

```
x % 2 + y / z + z + y / (z + z)
```

Out[17]:

11.75

primero se realizan las divisiones, así:

In [22]:

```
y/(z+z)
```

Out[22]:

0.25

In [23]:

```
y/z
```

Out[23]:

0.5

en la segunda parte operamos %

In [24]:

```
x%2
```

Out[24]:

1

finalmente operamos las sumas, así:

In [28]:

```
1+0.5+10+0.25
```

Out[28]:

11.75

el resultado de la operacion es:

In [31]:

```
w=11.75
```

3. [Pensar como un computador] ¿Cuál es el valor de *c* después de ejecutar el siguiente código?

```
c = True d = False c = c and d c = not c or d
```

In [97]:

```
c = True
```

In [98]:

```
c
```

Out[98]:

True

In [99]:

```
d = False
```

In [100]:

```
d
```

Out[100]:

False

And

In [101]:

```
c = c and d
```

In [102]:

```
c
```

Out[102]:

False

Or

In [105]:

```
c= c or d
```

In [106]:

```
c
```

Out[106]:

True

Not

In [103]:

```
c = not c or d
```

In [104]:

```
c
```

Out[104]:

True

4. Ejecute el siguiente código y responda: ¿Por qué es falsa la tercera línea, mientras que las primeras dos son verdaderas?

```
1 == 1 "1" == "1" 1 == "1"
```

In [57]:

```
1 == 1
```

Out[57]:

True

In [58]:

```
"1" == "1"
```

Out[58]:

True

In [59]:

```
1 == "1"
```

Out[59]:

False

R: La tercera es falsa ya que los operadores son diferentes, es decir 1 es diferente que "1"

5. Escriba un programa que le pida al usuario ingresar su nombre y que arroje un texto saludando de vuelta al usuario, así: "Hola, <nombre>. ¡Veo que aprendes Python rápidamente! ¡Felicitaciones!".

In [60]:

```
nombre = input("Digame su nombre: ")
```

Digame su nombre: Monica

In [62]:

```
nombre
```

Out[62]:

```
'Monica'
```

In [74]:

```
print("¡Hola", nombre, "!")
```

```
¡Hola Monica !
```

In [75]:

```
print("¡Hola", nombre, "¡Veo que aprendes Python rápidamente! ¡Felicitaciones!")
```

```
¡Hola Monica ¡Veo que aprendes Python rápidamente! ¡Felicitaciones!
```
