

Taller 1

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 8-feb-2019 11:59 PM

****Francisco Monsalve****

francisco.monsalve@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 Juan David 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.

Respuestas de selección múltiple:

- 1) B. What can be computed?
- 2) D. Recipe
- 3) D. It is not practical to solve
- 4) A. RAM
- 5) B. High-level computer language
- 6) B. A complete computer command
- 7) B. A compiler is used to translate high-level language into machine language
- 8) B. Main
- 9) A. They make the program more efficient
- 10) D. Parameters

Respuestas de "Programming Exercises"

```
In [3]: print("Hello, World!")
```

Hello, World!

```
In [4]: print("Hello", "world!")
```

Hello world!

```
In [5]: print(3)
```

3

```
In [6]: print(3.0)
```

3.0

```
In [7]: print(2+3)
```

5

```
In [8]: print(2.0+3.0)
```

5.0

```
In [9]: print("2"+"3")
```

23

```
In [11]: print("2"+"3=", 2+3)
```

23= 5

```
In [12]: print(2*3)
```

6

```
In [13]: print(2**3)
```

8

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?

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

Valor de W $W = 1 + 1/2 + 10 + 5/20$ $W = 1 + 0,5 + 10 + 0,25$ $W = 11,75$

In [2]: `x = 7
y = 5.0
z = 10.0
w = x % 2 + y / z + z + y / (z + z)
print (w)`

11.75

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

$c = \text{True}$ $d = \text{False}$ $c = c \text{ and } d$ $c = \text{not } c \text{ or } d$

---Vedadero

In [15]: `c = True`

```
d = False
c = c and d
c = not c or d
print (c)
```

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"
```

1 == 1 es verdadero porque está diciendo que el valor 1 es igual a 1. Ambos son type integers
"1" == "1" es verdadero porque está diciendo que el texto "1" es igual a "1". Ambos son type string
1 == "1" es falso porque el valor de 1 no es igual al texto "1". Uno es type integer y el otro string

```
In [17]: 1 == 1
```

```
Out[17]: True
```

```
In [18]: "1" == "1"
```

```
Out[18]: True
```

```
In [19]: 1 == "1"
```

```
Out[19]: False
```

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 [20]: nombre_user = input ("por favor ingrese su nombre")  
         "Hola, " + nombre_user + ". " + "¡Veo que aprendes Python rápidamente!  
         ¡Felicitaciones!"
```

por favor ingrese su nombreFrancisco

```
Out[20]: 'Hola, Francisco. ¡Veo que aprendes Python rápidamente! ¡Felicitaciones!'
```