

# mcpp\_taller1\_juan\_munoz

August 9, 2019

## 1 Taller 1

Métodos Computacionales para Políticas Públicas - UROSARIO

**Entrega: viernes 9-ago-2019 11:59 PM**

**Juan Sebastian Muñoz Vargas** jsebastianmvargas@gmail.com

### 1.1 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\_mataallana
- 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.)

---

### 1.2 1. Zelle, sección 1.10 (p. 17):

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

“Multiple Choice” - Ejercicios

1. **What is the fundamental question of computer science?** b) What can be computed?
2. **An algorithm is like a** d) recipe
3. **A problem is intractable when** d) it is not practical to solve
4. **Which of the following is not an example of secondary memory?** a) RAM
5. **Computer languages designed to be used and understood by humans are** b) high-level computer
6. **A statement is** b) a complete computer command
7. **One difference between a compiler and an interpreter is** b) a compiler is used to translate high-level language into machine language
8. **By convention, the statements of a program are often placed in a function called** b) main
9. **Which of the following is not true of comments?** a) They make a program more efficient.
10. **The items listed in the parentheses of a function definition are called** d) both b) and c) are correct

“Programming Exercises” Start up an interactive Python session and try typing in each of the following commands. Write down the results you see. a) `print (“Hello, world!”)` b) `print(“Hello”, “world!”)` c) `print (3)` d) `print(3.0)` e) `print (2 + 3)` f) `print(2. 0 + 3. 0)` g) `print ( II 2 II + II 3 II )` h) `print ( “2 + 3 =”, 2 + 3)` i) `print(2 * 3)` j) `print (2 ** 3)` k) `print (7 I 3)` l) `print(? // 3)`

[1]: `print (“Hello, world! ”)`

Hello, world!

[2]: `print(“Hello”, “world!”)`

Hello world!

[3]: `print(3)`

3

[4]: `print(3.0)`

3.0

[5]: `print(2 + 3)`

5

[9]: `print(2.0 + 3.0)`

5.0

[10]: `print(“2” + “3”)`

23

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

2 + 3 = 5

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

6

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

8

```
[15]: print(7 / 3)
```

2.3333333333333335

```
[16]: print(7 // 3)
```

2

---

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.

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

$x = 7$   $y = 5.0$   $z = 10.0$   $w = x$  La respuesta es: 11.75

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

$c = \text{True}$   $d = \text{False}$   $c = c \text{ and } d$   $c = \text{not } c \text{ or } d$  La respuesta es: True

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

$1 == 1$   $"1" == "1"$   $1 == "1"$

```
[18]: 1 == 1
```

```
[18]: True
```

```
[20]: "1" == "1"
```

```
[20]: True
```

```
[21]: 1 == "1"
```

```
[21]: False
```

La tercera línea es falsa porque el 1 está siendo tomado un entero, mientras que el "1" por estar entre comillas es tomado como un carácter. Así, en el primer caso se comparan dos números enteros, en el segundo dos caracteres iguales; más en el tercero un carácter y un número entero. Lo anterior se ve a continuación:

```
[22]: type(1)
```

```
[22]: int
```

```
[23]: type("1")
```

```
[23]: str
```

---

**1.6 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!".**

```
[ ]: print("Bienvenido/a, ¿cuál es tu nombre?")
nombre = input ()
print("Hola, " + nombre + ". ¿Veo que aprendes Python rápidamente!
→¿Felicitaciones!")
```