

# Taller 1

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 21-ago-2020 11:59 PM

**\*\*[Nicolás Garcés Rodríguez]\*\***

[nicolas.garces@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\_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. Zelle, sección 1.10 (p. 17):

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

## Multiple Choice

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?

11. 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 languages

6. A statement is

b) a complete computer command

7. One difference between a compiler and an interpreter is

c) a compiler is no longer needed after a program is translated

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) parameters

## Programming Exercises

1. 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!")`**

```
In [1]:
```

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

```
hello,world!
```

Al imprimir un elemento entre comillas, el resultado es el contenido literal del elemnto.

**(b) `print("Hello", "world!")`**

```
In [2]:
```

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

```
Hello world!
```

La , sirve para separar elementos str al imprimirlos.

**(c) `print(3)`**

In [3]:

```
print(3)
```

3

Al imprimir un elemento integer el resultado es el mismo integer.

**(d) `print(3.0)`**

In [5]:

```
print(3.0)
```

3.0

Al imprimir un floating point number el resultado es el mismo floating point number.

**(e) `print(2 + 3)`**

In [5]:

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

5

Al imprimir una operación matemática el resultado es la solución de la operación.

**(f) `print(2.0 + 3.0)`**

In [6]:

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

5.0

Al imprimir una operación matemática entre floating point numbers el resultado es la solución de la operación, que es un floating point number.

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

In [7]:

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

23

Al imprimir la suma de dos elementos str el resultado es la unión de los dos elementos en un elemento str.

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

In [8]:

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

2 + 3 = 5

Esta impresión tiene dos partes. Una parte de una cadena que se imprime tal cual está entre "" y una operación entre enteros en la que se imprime el resultado. las dos partes se separan con la coma.

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

In [9]:

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

6

Al imprimir una operación matemática, el resultado es la solución de la operación.

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

In [10]:

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

8

El simbolo \*\* es de exponenciación, por lo que esta expresión es 2 elevado a la 3.

**(k) `print(2 / 3)`**

In [11]:

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

0.6666666666666666

Al imprimir una operación matemática el resultado es la solución de la operación, en este caso un número irracional.

---

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?

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

1.  $w = x \% 2 + y / z + z + y / (20.0)$
2.  $w = x \% 2 + 0.5 + z + 0.25$
3.  $w = 1 + 0.5 + z + 0.25$
4.  $w = 1 + 0.5 + 10.0 + 0.25$
5.  $w = 11.75$

In [30]:

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

### 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
```

1. `c = c` y `d = false`
2. `c = no c` o `d = True`
3. `c = True`

In [6]:

```
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"
```

`==` significa igual, por ende:

1. `1==1` es verdadero porque 1 es igual a 1 y son de la misma clase (int).
2. `"1"=="1"` es verdadero porque "1" es igual a "1" y son de la misma clase (str).
3. `1 == "1"` es falso porque son elementos de diferentes clases.

### 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 [37]:

```
nombre = input("Escriba su nombre:", )
print("Hola, ", nombre, ". ¡Veo que aprendes Python rápidamente! ¡Felicitaciones!.")
```

Escriba su nombre:Nicolás

Hola, Nicolás . ¡Veo que aprendes Python rápidamente! ¡Felicitaciones!.