

# mcpp\_taller1\_monica\_gasca

August 11, 2016

## 1 Taller 1

Métodos Computacionales para Políticas Públicas - UROSARIO

**Entrega: viernes 12-ago-2016 11:59 PM**

[ monica gasca] [monicagascarojas@gmail.com]

### 1.1 Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material 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.

(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

1. b

2. c
3. d
4. a
5. b
6. b
7. c
8. b
9. a
10. d

### Programming Exercise

```
In [2]: print ("Hello, world")
        print ("Hello", "world!")
        print (3)
        print (3.0)
        print (2+3)
        print (2.0+3.0)
        print ("2"+"3")
        print ("2+3=", 2+3)
        print (2*3)
        print (2**3)
        print (2/3)
```

```
Hello, world
Hello world!
3
3.0
5
5.0
23
2+3= 5
6
8
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.

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

$x = 7$   $y = 5.0$   $z = 10.0$   $w = x$

Respuestas :

$x \% 2 = 1$  residuo de cuantas veces esta 2 en 7

```
y/z= 0.5
y/(z+z)= 0.25
w=11.75
```

**1.4 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
```

El valor de  $c$  después de ejecutar es falso, pues al final el `not c`, elimina  $c$  (verdadero y falso) y queda  $d$  que siempre ha sido falso

**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"
```

```
In [3]: 1 == 1
```

```
Out[3]: True
```

```
In [4]: "1" == "1"
```

```
Out[4]: True
```

```
In [5]: 1 == "1"
```

```
Out[5]: False
```

La tercer línea es falsa porque el tipo de  $1$  es distinto,  $1$  es un entero y `"1"` es string, mientras que las dos primeras validaciones se hacen entre elementos del mismo tipo.

**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!"**

```
In [7]: name= input ("enter your name: ")
        print ("Hola", name, ".¡Veo que aprendes Python rapidamente! ¡Felicitaciones!")
```

```
enter your name: monica
```

```
Hola monica .¡Veo que aprendes Python rapidamente! ¡Felicitaciones!
```

---