

mcpp_taller1_john_caro

August 12, 2016

1 Taller 1

Métodos Computacionales para Políticas Públicas - URSario

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

[John Alexander Caro Becerra] [jhonalexbc@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_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.

(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.

1.2.1 Multiple Choice

1. B
2. D
3. D
4. A
5. B
6. B
7. C
8. B
9. A
10. D

1.2.2 Programming Exercises

```
In [1]: print("Hello, world!")
```

```
Hello, world!
```

Hello, world! es un tipo str y sale la coma que se encuentra en la frase

```
In [2]: print("Hello", "world!")
```

```
Hello world!
```

Hello world! es un tipo str pero no sale la coma, ya que no esta en comillas y de paso separa las palabra hello y world!

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

```
3
```

3 es un tipo int

```
In [4]: print(3.0)
```

```
3.0
```

3.0 es un tipo float

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

```
5
```

5 es un tipo int, ademas se imprime la suma entre 2+3, dos número enteros

```
In [6]: print(2.0+3.0)
```

5.0

5.0 es un tipo float, además se imprime la suma de entre 2.0+3.0, dos números decimales

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

23

23 es un tipo str ya que al estar el 2 y 3 entre comillas, se imprime la unión de caracteres o sea 23

```
In [8]: print("2 + 3 =", 2+3)
```

2 + 3 = 5

se imprime la combinación entre tipos str y int, ya que se va imprimir 2+3= ya que es un carácter por estar entre comillas, pero el 5 es el resultado de la suma de los enteros

```
In [9]: print(2*3)
```

6

se imprime el resultado de la multiplicación entre 2 y 3

```
In [10]: print(2**3)
```

8

se imprime el operador de elevación o de potencia, es decir, se imprime el resultado de 2 elevado a la 3

```
In [11]: print(2/3)
```

0.6666666666666666

se imprime la división entre 2 y 3

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 $z = 10.0$ $w = x$ El valor de w es 11.75, ya que la operación es: $x\%2$ es igual a 1 ya que $7/2$ es igual a 6 y % saca el excedente de la división, por eso es 1. Luego se suma la división entre 5 sobre 10 que es igual a $\frac{1}{2}$; luego se suma 10. Luego se suma la división entre 5 sobre 20 dando como resultado $\frac{1}{4}$. Es decir $1 + \frac{1}{2} + 10 + \frac{1}{4}$ es igual a 11.75.

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

$c = \text{True}$ $d = \text{False}$ $c = c \text{ and } d$ $c = \text{not } c \text{ or } d$ El valor de c es True, ya que $c = c \text{ and } d$ convierte a c en False porque verdadero y falso es igual a falso, pero en la línea $c = \text{not } c \text{ or } d$ lo convierte en True ya que falso o falso es falso pero al negarlo con not se convierte en 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"
```

```
In [12]: 1 == 1
```

```
Out[12]: True
```

```
In [13]: "1" == "1"
```

```
Out[13]: True
```

```
In [14]: 1=="1"
```

```
Out[14]: False
```

```
In [15]: type("1")
```

```
Out[15]: str
```

```
In [16]: type(1)
```

```
Out[16]: int
```

La razón principal es porque en la primera prueba lógica se está probando si 1 es igual a 1, siendo 1 un número entero en este caso, es decir, se comparan dos números enteros (int) que son iguales. En la segunda prueba lógica se está comparando es dos cadenas (str) ya que al colocar entre comillas los 1, estos se convierten en cadena automáticamente, siendo así que al compararse 1 entre comillas contra otro uno entre comillas, el resultado es verdadero ya que los dos son iguales. Pero en la tercera prueba lógica, la principal razón por la que es falsa es que se están comprobando si un número entero es igual a un carácter o tipo cadena y un número jamás será igual que un carácter o cadena en este caso.

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 [17]: print("Por favor ingrese su nombre.")
         nombre=input()
```

```
Por favor ingrese su nombre.
John Alexander Caro Becerra
```

```
In [18]: print("Hola, ",nombre,". ¡Veo que aprendes Python rápidamente! ¡Felicitaciones!")
```

```
Hola,  John Alexander Caro Becerra . ¡Veo que aprendes Python rápidamente! ¡Felicitaciones!
```