

Taller de Python

Clase I

Comisión de Talleres

Centro de Estudiantes Tecnológicos

Table of contents

1. Introducción
2. Funciones
3. Variables
4. Controladores de flujo
5. Repaso

Introducción

Para llevar a cabo las actividades propuestas, recomendamos descargar **Anaconda** de Continuum siguiendo el enlace que ofrecemos a continuación.

`https://www.continuum.io/downloads`

¿Qué es Python?

Python es un **lenguaje de programación** con las siguientes características:

- Es un lenguaje interpretado,
- multiparadigma,
- fuertemente tipado.

¡También es muy sencillo!

Funciones

Una función es un fragmento de código con un nombre asociado que realiza una serie de tareas y devuelve un valor.

El interprete de Python tiene un número de funciones y tipos siempre disponibles.

`https://docs.python.org/3/library/functions.html`

Funciones

Para definir nuestras propias funciones, usamos el comando **def**.

```
def F(x,y):  
    return x + y**2  
  
def G():  
    print('Hello world!')
```

- La línea que contiene al comando **def** termina siempre con ':'.
• Todas las sentencias que formen parte de la definición de la función deben estar correctamente indentadas.

¿Qué diferencia existe entre **return** y **print**?

Variables

Python tiene cinco tipos de datos estándares:

1. Números
2. *Strings*
3. Listas
4. Tuplas
5. Diccionarios

```
'''A las variables a, b y c se le asignan distintos valores, como se muestra a continuación.'''
```

```
a = 10
```

```
b = 10.0
```

```
c = 10 + 10j
```

```
# Utilice la función type() para saber de qué tipo de  
# dato se trata
```

¿Qué diferencias existen entre los distintos tipos? ¿es posible convertir de un tipo a otro?

Strings

Los *Strings* en Python se identifican como un conjunto de caracteres contiguos encerrados entre comillas.

```
'Mi nombre es...'
```

El texto entre comillas es un tipo de dato y es por lo tanto posible operar con él.

```
a = 'nombre'  
type(a)  
len(a)  
b = a[0]  
2 * a
```

¿Qué resultados se obtienen al ejecutar las acciones sugeridas?

Listas Contienen elementos separados por comas escritos entre corchetes.

Tuplas Son semejantes a las listas pero sus elementos están escritos entre paréntesis.

Diccionarios Sus elementos consisten en pares *key-value* separados por comas y escritos entre llaves.

¿Qué otras diferencias existen entre estos tipos de datos?

Listas, tuplas y diccionarios

```
# Lista
```

```
list = ['a', 'b', 3, 5, 7]
```

```
# Tupla
```

```
tuple = ('a', 'b', 3, 5, 7)
```

```
# Diccionario
```

```
dict = {1:'Ana', 2:'Bruno', 3:'Carlos'}
```

```
list.append(9) # Intente hacer lo mismo con la tupla
```

```
list
```

```
list[0]
```

```
list[2:3]
```

Asignación de variables

Softcoding es un término que en programación hace referencia al hecho de obtener un valor o función desde una fuente externa. Es lo opuesto de **hardcoding**, término que hace referencia a programar valores y funciones en el código fuente.

Evitar el hard-coding de valores comúnmente modificados es una buena práctica en programación.

`input()` es una función que permite que el usuario ingrese datos y asignarlos a variables definidas en el código del programa.

```
Te = input('Ingrese la temperatura inicial en Kelvin: ')\nTe = float(Te)
```

¿Por qué necesitamos la segunda línea?

Controladores de flujo

Condicionales

```
number = 23
guess = int(input('Enter an integer : '))

if guess == number:
    print('Congratulations, you guessed it.')
    print('(but you do not win any prizes!)')
elif guess < number:
    print('No, it is a little higher than that')
else:
    print('No, it is a little lower than that')

print('Done')
```

Analice el código presentado prestando especial atención a la *indentación*.

Condicionales

```
number = 23
running = True # boolean

while running:
    guess = int(input('Enter an integer : '))
    if guess == number:
        print('Congratulations, you guessed it.')
        running = False
    elif guess < number:
        print('No, it is higher than that.')
    else:
        print('No, it is lower than that.')
print('Done')
```

¿Qué diferencia(s) hay entre los dos códigos presentados?

```
for i in range(1, 5):  
    print(i)  
else:  
    print('The for loop is over')
```

- This is important

- This is important
- Now this

- This is important
- Now this
- And now this

- This is really important
- Now this
- And now this

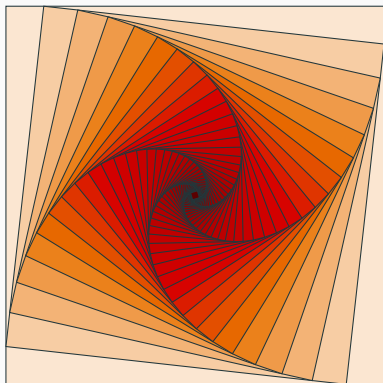


Figura 1: Rotated square from texample.net.

Tabla 1: Largest cities in the world (source: Wikipedia)

City	Population
Mexico City	20,116,842
Shanghai	19,210,000
Peking	15,796,450
Istanbul	14,160,467

Three different block environments are pre-defined and may be styled with an optional background color.

Default

Block content.

Alert

Block content.

Example

Block content.

Default

Block content.

Alert

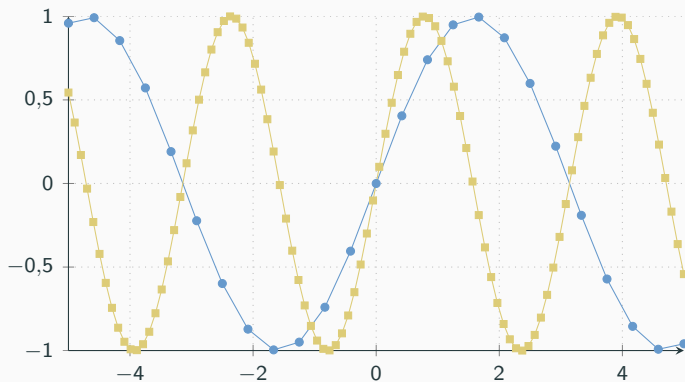
Block content.

Example

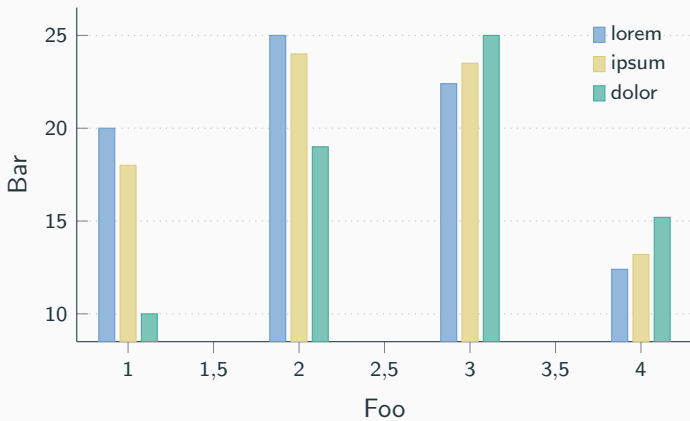
Block content.

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

Line plots



Bar charts



Veni, Vidi, Vici

metropolis defines a custom beamer template to add a text to the footer. It can be set via

```
\setbeamertemplate{frame footer}{My custom footer}
```

Repaso

¿Preguntas?

Enlaces de interés

- [Comunidad Python Argentina](http://www.python.org.ar)
`http://www.python.org.ar`
- [AeroPython](https://github.com/AeroPython)
`https://github.com/AeroPython`
- [Tutorialspoint](https://www.tutorialspoint.com/index.htm)
`https://www.tutorialspoint.com/index.htm`
- <https://www.gitbook.com/book/swaroopch/byte-of-python/details>
`https://python.swaroopch.com/control_flow.html`
- [Python programming](https://pythonprogramming.net)
`https://pythonprogramming.net`