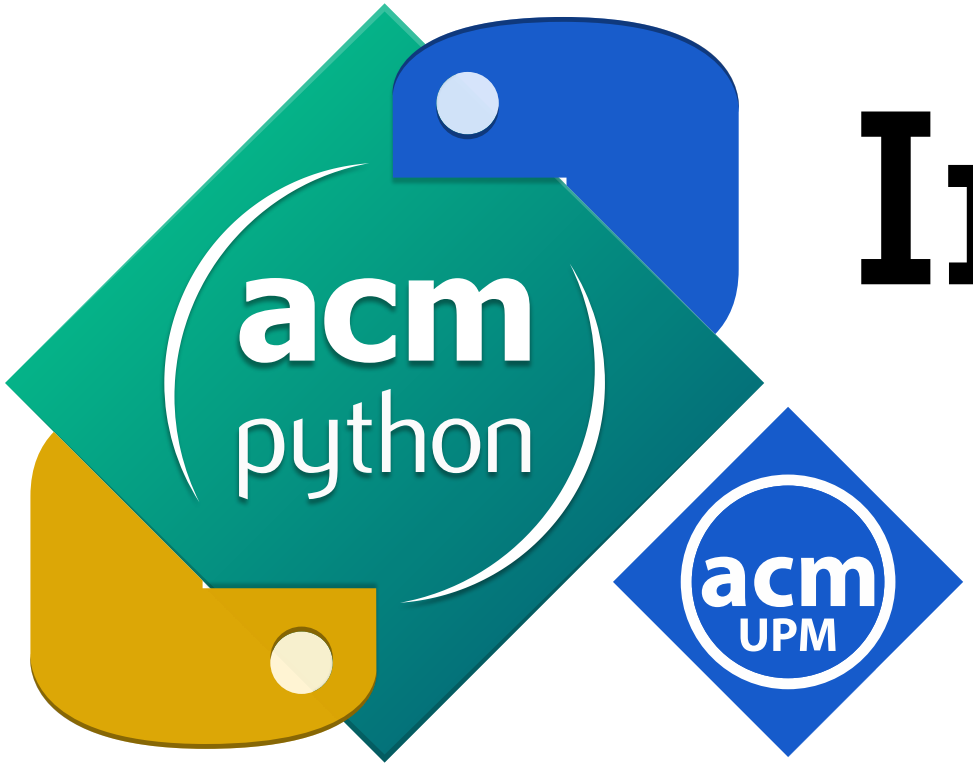
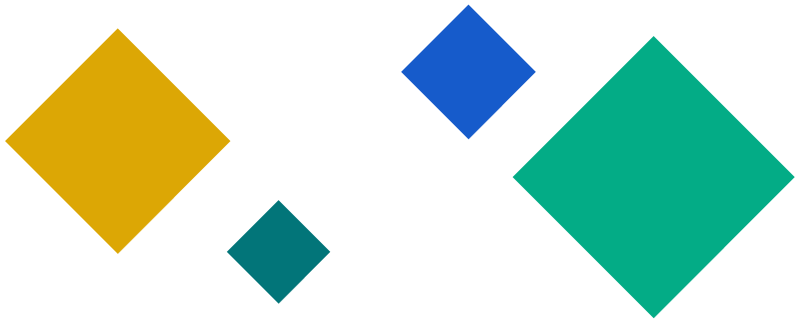


Intro a Python



Introducción

Este taller tiene un repositorio en Github.

Aunque probablemente no hayáis usado Git nunca, ¡os recomendamos que os familiaricéis con esta herramienta!

Si no queréis podéis descargar el repositorio en un zip.

https://github.com/aafrecct/python-sig-intro

Instalación

La forma de instalar Python, varía dependiendo del sistema operativo. Tenéis las instrucciones en el README.

A veces en Windows, Python no se añade al PATH. Una vez instalado, intentad ejecutar "python" desde la consola de comandos o PowerShell. Si os da un error, o se os abre la tienda de Windows, ¡decidlo!

Interprete

"A los profesores de programación no les va a compilar el código en Python" -- S.Eibe

Entre otras cosas, porque la implementación principal de Python es interpretada, no compilada (Aunque depende de a quien le preguntes).

Esto significa que lee el código "línea a línea" y lo va ejecutando, en vez de compilarlo a código máquina una vez y luego ejecutar ese código máquina.

Tipos

```
number = 42
floating_point = 1.4
string = "Yellow"

print(number, " is a ", type(number))
print(string, " is a ", type(string))

# Type hinting
number_2: int = 5
string_2: str = 'Brian'

number = 3.14
print(number, " is a ", type(number))
```

Hello World

```
my_name = "borja"
my_enemys_name = "anto"
user_name = input("What's your name? ")

if (un := user_name.lower()) == my_name:
    print("Hello {}!".format(user_name))
elif un == my_enemys_name:
    print("Get the fuck out of here {}!!!".format(user_name))
else:
    print("I don't really know who you are...")
```

Listas

```
the_acm_gang = ['Anto', 'Paula', 'Jorge', 'Neku', 'Diego',
                'Carlos', 'Younes', 'Ferrero', 'Gaspar',
                'Roberto', 'Jas', 'Samu', 'Borja']
user_name = input("What's your name? ")

user_in_gang = False
for member in the_acm_gang:
    user_in_gang |= user_name.lower() == member.lower()

print("Have a good day" if user_in_gang
      else "Join ACM or die.")
```

Listas por comprehension

```
good_students = [input('Name: ') for i in range(2)]
bad_students = [input('Name: ') for i in range(6)]

sad_students = good_students + bad_students
acm_students = [student for student in sad_students
                if student[0].lower() in ['a', 'c', 'm']]

print(acm_students)
```

Representación de matrices

```
matrix = [[1, 0, 0, 1],
          [0, 0, 0, 1],
          [1, 1, 0, 0]]

print(matrix[2][2])
print(matrix[1][3])
```

Diccionarios

```
fav_games = {'Anto' : 'Pokemon Red',
             'Gaspar' : 'MegaMan 2',
             'Borja' : 'Minecraft'}

your_game = input("What's your favorite game? ")

if your_game in fav_games.values():
    print("Your favorite game is good.")
else:
    print("Your favorite game is bad.")
```

Tuplas

```
coords = (10, 15)
x, y = coords

coords_2 = (3 * i for i in coords)

# ¿Que imprimiran las siguientes lineas?
print(x, y)
print(coords_2)
print(type(coords), type(coords_2))
```

Sets

```
the_acm_gang = {'Anto', 'Paula', 'Jorge', 'Neku', 'Diego',
                'Carlos', 'Younes', 'Ferrero', 'Gaspar',
                'Roberto', 'Jas', 'Samu', 'Borja'}
traitors = set()
afirmative_answers = {'yes', 'y', 'yeah', 'yep', 'si'}

user_name = input("What's your name? ")
user_wants_in = input("Wanna join ACM? ").lower() \
                in afirmative_answers

group_to_add = the_acm_gang if user_wants_in else traitors
group_to_add.add(user_name)

print("ACM:", the_acm_gang, "\nTraitors:", traitors)
```

Funciones

```
def list_adder(ls):
    for i in range(len(ls) - 1):
        ls[i + 1] += ls[i]

list_1 = [0, 1, 2, 3, 0]
list_2 = [-1, 1, -2, 1, 4, 8, -5, -3, 0]

list_adder(list_1)
list_adder(list_2)

print(list_1)
print(list_2)
```

PIP pip

```
pip3 install pygame
# or
python3 -m pip install pygame
```

El juego de la vida

Es un juego de simulación, que consiste de una cuadricula, con células "vivas" y células "muertas".
Se establecen unas normas para que nazcan, vivan y mueran según pasan "turnos" y luego podemos ver como reaccionan distintas formas y estructuras dentro del juego.

Las reglas que tenéis que implementar son las siguientes:

https://en.wikipedia.org/wiki/Conway's_Game_of_Life#Rules

El juego de la vida

En la carpeta 'code' tenéis tres archivos, queremos que escribais la función 'next_step' que hay dentro del archivo 'game.py'. Podéis definir otras funciones, variables, lo que queráis.

Necesitamos que next_step reciba como parámetro o una matriz o un set de coordenadas, aplique las reglas del juego y luego lo devuelva.

Las reglas que tenéis que implementar son las siguientes:

https://en.wikipedia.org/wiki/Conway's_Game_of_Life#Rules

Contacto

Si no habéis terminado con el juego y quereis que alguien lo mire, podéis mandármelo por:

Telegram: @bmcaos
Email: bmartinenac+intropy@gmail.com

¡¡¡Y uníos todos a ACM!!!