

Práctica 1 - Introducción a Python

Ingeniería de Software 2024-1

Erick Martínez Piza
ermarp@ciencias.unam.mx

Rogelio Alcantar Arenas
rogelio-aa@ciencias.unam.mx

Luis Fernando Yang Fong Baeza
fernandofong@ciencias.unam.mx

Valeria García Landa
vale.garcia.landa@gmail.com

Francisco Valdés Souto
fvaldes@ciencias.unam.mx

23 de Agosto 2023

1. Introducción

El motivo de esta práctica es para que el alumno aprenda el uso correcto de Python en un entorno muy parecido a desarrollo profesional, así como distintas bibliotecas y el uso de entornos virtuales, así como ventajas y desventajas.

La práctica consta de 4 puntos con un valor de 2.5 puntos cada script, cada script debe de ser generado por el alumno siguiendo las convenciones que en este PDF se muestran.

2. Información

Python es un lenguaje de programación interpretado y muy versátil, puesto que cuenta con muchos componentes de distintos paradigmas como el paradigma orientado a objetos, el paradigma imperativo o paradigma declarativo. Por este motivo, su popularidad ha crecido bastante y hoy en día es uno de los lenguajes más usados en la industria.

Para este curso (incluyendo el desarrollo de los casos de uso) hemos decidido usar este lenguaje ya que permite el uso de entornos virtuales, haciendo los desarrollos mucho más sencillos sin necesidad de tener una gran instalación y/o arquitectura.

Un entorno virtual es la creación de un intérprete en una versión específica pero que no cuenta con alguna biblioteca específica, es decir, un intérprete sin instalaciones de bibliotecas externas, todo esto sin afectar el intérprete que viene instalado por default en la memoria de la computadora, sin importar la versión tampoco, por ejemplo, es posible que en la versión de la computadora se cuente con la versión 3.8.5 del intérprete de Python y en un entorno virtual con la versión 3.9.0.

Esto facilita los desarrollos puesto que de esta manera el entorno virtual ahora forma parte del proyecto también y puede ser ejecutado en cualquier computadora.

Python consta de una sintaxis bastante intuitiva que para cualquier persona que cuente con un conocimiento relativamente básico del idioma inglés y conceptos básicos de programación, le sea muy fácil aprender a programar en este lenguaje, sin mencionar que cuenta con todo tipo de bibliotecas para distintos propósitos como puede ser, cómputo científico, cómputo para servidores, cómputo para aplicaciones o hasta incluso ejecutar código en una Raspberry Pi, además de que Python cuenta con el instalador de paquetes conocido como `pip`

que facilita la búsqueda e instalación de paquetes, así como el saber qué paquetes ya se encuentran instalados.

En esta práctica se aprenderá el uso del lenguaje Python desde la sintaxis, estructuras de control, clases, objetos y bibliotecas externas, en esta práctica NO se espera el uso de cómputo concurrente ni es acreedor a puntos extras el implementarlo de esta manera.

3. Desarrollo

El alumno deberá de seguir los siguientes puntos para la creación de su entorno virtual así como la creación de sus scripts para cada uno de los puntos. Los 4 incisos constan del mismo valor respecto a la calificación, de manera que se recomienda hacer primero los que se consideren más fáciles para dejarle mayor tiempo a los scripts más difíciles.

Todos los scripts deben de ser ejecutados con el entorno virtual, sin excepción alguna.

3.1. Creación de entornos virtuales

Como primer paso, el alumno debe de crear un entorno virtual, para esto basta con usar el comando:

```
python3 -m venv {nombre_entorno}
```

Esto creará una carpeta con el nombre dado en el comando mencionado que hace referencia al entorno virtual. Para acceder a él, basta con movernos a dicho directorio (Utilizando el comando `cd`) y ahí dentro, ejecutar:

```
source ./bin/activate
```

Esto modificará la terminal, agregando el nombre del ambiente virtual al inicio de la línea de comandos. Para verificar que se cuenta con un entorno nuevo, se recomienda que se ejecute el comando: `pip list`. Para salirse del entorno, basta con ejecutar el comando: `deactivate`

Punto extra: El alumno que encuentre la manera de ejecutar sus scripts sin entrar al entorno virtual, será acreedor a un punto extra para esta práctica.

3.2. Componentes básicos del lenguaje

El alumno deberá de crear un script que pueda llevar el marcador de un partido de tenis (es obligación del alumno investigar el conteo de este deporte), en donde, el script debe de tener el siguiente comportamiento:

1. El script debe de leer el nombre de los dos jugadores.
2. El script debe de avisar cuando el juego haya terminado y quien es el ganador, el juego está esperado a ser el mejor de 3 sets.
3. El script debe de esperar hasta que el usuario ingrese quién ha hecho un punto.
4. El script debe de avisar cuando el juego haya concluido y que jugador se lleva dicho juego.
5. El script también debe de avisar cuando los jugadores tengan que hacer cambio de cancha.
6. El programa debe de avisar cuando algún jugador haya ganado un set.

La entrada del programa es a libre formato pero debe de ser especificado dicho formato en un archivo `readme_script1`, la salida debe de ser la salida estándar. (Terminal)

El nombre de este script deberá de ser `Script1.py`

3.3. Estructuras de datos y objetos

Python como cualquier lenguaje cuenta con estructuras de datos integradas y además permite la creación de las mismas, de esta manera para el segundo script debe de contener lo siguiente:

1. Una función que resuelva el siguiente problema; dado un arreglo de números como parámetro y un número **target** se debe de determinar si existen dos números s_i, s_j con $i \neq j$ tales que $s_i + s_j = target$, en caso de existir, se debe de regresar como respuesta $[i, j]$, si no existe dicha pareja, entonces regresar $[-1, -1]$.
2. Una implementación de un árbol binario ordenado utilizando la palabra reservada **class**. La implementación es a libre albedrío, de manera que no necesariamente se debe de hacer una clase nodo, cada alumno es libre de implementar su árbol como sea, sin embargo, este árbol debe de cumplir con poder ejecutar los 3 recorridos de árboles (pre, in y post order), obviamente, también se debe de incluir el método **agregar** puesto que si no, esta clase es inservible, el método de **eliminar**, no es necesario pero tampoco es de punto extra.

Punto extra: Al alumno que entregue la solución del primer punto con una complejidad de $O(n \log n)$ o más rápido.

El nombre de este script deberá de ser **Script2.py**

3.4. Importación de bibliotecas externas

Utilizando el instalador de paquetes **pip**, deberán de instalar la biblioteca de **matplotlib** y crear un script que utilice esta biblioteca para graficar una función de la elección del alumno, puede ser cualquier tipo de función pero es responsabilidad del alumno, aprender como es que funciona la sintaxis de **matplotlib**.

Para la instalación, bastaría con utilizar el comando **pip install matplotlib**. El nombre de este script deberá de ser **Script3.py**.

4. Formato de entrega

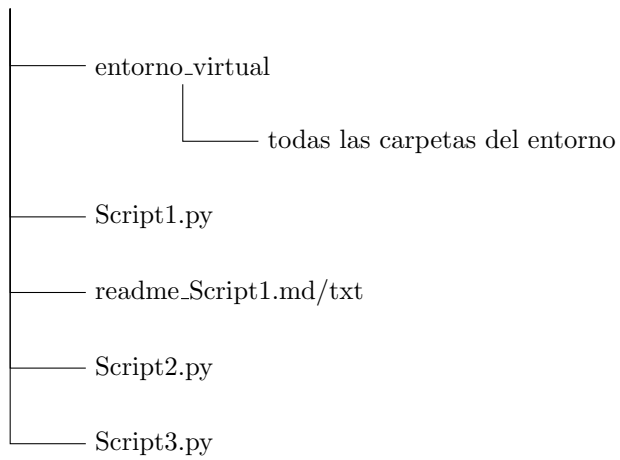
La entrega es individual por cada alumno, no es válido entregar las prácticas en parejas, tercias ni cualquier cantidad mayor de integrantes.

Se deberá de entregar un único archivo comprimido (**.zip**) a través de la plataforma Moodle, cuyo nombre debe de corresponder al del alumno empezando por apellido paterno seguido del apellido materno y el número 01, ejemplo:

GarciaLanda01.zip
FongBaeza01.zip

El contenido de la carpeta que se comprime debería de ser:

GarciaLanda01



La entrega es para el día Miércoles 30 de Agosto, antes de las 23:59:59.