

October 7, 2020

## 1 Fundamentos de Programación

En este Notebook se encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podréis sacar la máxima nota de la PEC sin necesidad de hacer este ejercicio. El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Veréis que todas las actividades de la PEC tienen una etiqueta, que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- **NM Sólo materiales:** las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- **EG Consulta externa guiada:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- **EI Consulta externa independiente:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, podéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

---

### 1.0.1 Ejercicio 1

Dada la lista:

```
[1]: lst = [1, 5, 7, 13, 22, 15, 26, 64, 34, 72, 52, 14]
```

a) Recorre la lista y muestra el índice y el valor de cada posición. NM (0.5 puntos)

```
[2]: # Respuesta:
```

b) Recorre la lista y muestra el índice y valor de cada posición, solo si el valor es múltiplo de 3. NM (0.5 puntos)

```
[3]: # Respuesta:
```

c) Recorre las posiciones pares de la lista, iniciando en el índice 0, y muestra su valor si este es menor a 25. NM (0.5 puntos)

```
[4]: # Respuesta:
```

d) Podemos ver que la lista no está ordenada, podríamos crear una función que la ordenara... pero ya existe una función que nos permite hacer esto. Utiliza la función `sort()` y muestra el resultado tras ordenar la lista de menor a mayor. EG (0.5 puntos)

**Nota:** Podemos consultar como utilizar `sort()` en la documentación de Python, o en la siguiente referencia <https://stackoverflow.com/questions/19199984/sort-a-list-in-python>

```
[5]: # Respuesta:
```

e) ¿Podrías mostrar la lista ordenada ahora de mayor a menor? EG (0.5 puntos)

**Nota:** Podemos consultar como utilizar `sort()` en la documentación de Python, o en siguiente referencia <https://stackoverflow.com/questions/19199984/sort-a-list-in-python>

```
[6]: # Respuesta:
```

## 1.0.2 Ejercicio 2

Python dispone de un **idiom** muy útil conocido como **list comprehension**. Utilizando este idiom, proporcionad una expresión que devuelva las listas siguientes.

**Nota:** Para realizar esta actividad necesitaréis investigar qué son las list comprehension y qué sintaxis utilizan. Para ello, se recomienda en primer lugar que utilicéis un buscador para encontrar información genérica sobre esta construcción. Después, os recomendamos que consultéis [stackoverflow](https://stackoverflow.com) para ver algunos ejemplos de problemas que se pueden resolver con esta construcción. [stackoverflow](https://stackoverflow.com) es un sitio de preguntas-y-respuestas muy popular entre programadores.

Veréis que para la gran mayoría de las dudas que tengáis, habrá alguien que ya les habrá tenido (y consultado) anteriormente! Así pues, más allá de preguntar vosotros mismos las dudas allí (nosotros ya tenemos el foro del aula para ello!), consultar esta web os permitirá ver qué soluciones proponen otros programadores a estas dudas. A menudo habrá más de una solución a un mismo problema, y podréis valorar cuál es la más adecuada para vuestro problema.

Para ver ejemplos de problemas que son adecuados para resolver con list comprehensions, os recomendamos las siguientes referencias:

- <https://stackoverflow.com/questions/12555443/squaring-all-elements-in-a-list>
- <https://stackoverflow.com/questions/57166908/using-list-comprehension-i-want-to-print-odd-even-with-string-indicating-even/57167016>

- a) Obtén con una sola expresión, una lista con el cuadrado más uno ( $x^2+1$ ) de cada valor de la lista. EG (0.5 puntos)

[7]: `lst = [1, 5, 7, 13, 22, 15, 26, 64, 34, 72, 52, 14]`

*# Respuesta:*

- b) Obtén ahora, también con una expresión, una lista con los strings “Odd” y “Even” en función de si el número de la lista proporcionada es par o impar. EG (0.5 puntos)

[8]: `lst = [1, 5, 7, 13, 22, 15, 26, 64, 34, 72, 52, 14]`

*# Respuesta:*

### 1.0.3 Ejercicio 3

Hace poco más de 10 años, la guipuzcoana Edurne Pasaban se convertía en la primera alpinista en coronar los catorce picos con una altura superior a 8000 metros del planeta. Elizabeth Hawley, conocida como la notaria del Himalaya, así lo reconoció tras no considerar clara una ascensión de la coreana Miss Oh que afirmaba haberlo logrado con anterioridad.

**Nota:** Podemos ver los picos de los que hablamos en la siguiente referencia <https://en.wikipedia.org/wiki/Eight-thousander>

- a) Genera un diccionario, usando el nombre del pico como clave y la altitud (en metros) como valor. EG (0.5 puntos)

[9]: *# Respuesta:*

- b) Recorre el diccionario y muestra el nombre de los picos con una altura superior a los 8200 metros. NM (0.5 puntos)

[10]: *# Respuesta:*

Superar los 8000 metros parece un reto muy importante, pero muchos países anglófonos como Estados Unidos, Canadá o el Reino Unido, utilizan el pie como unidad de medida.

Si sabemos que un pie equivale a 0.3048 metros

- c) Crea una función que tenga como entrada un valor numérico en metros y retorne la misma distancia en pies. Muestra como ejemplo la equivalencia en pies a 8848 metros. NM (1 punto)

[11]: *# Respuesta:*

- d) Utilizando la función definida anteriormente muestra el valor de altitud de todos los picos en pies (recorriendo el diccionario, pero sin modificar el valor en metros). NM (0.5 puntos)

[12]: # Respuesta:

- e) Es posible que tras la conversión, se haya mostrado la altura con muchos decimales. Modifica la función implementada anteriormente para redondear el valor con un decimal, y vuelve a mostrar el resultado. EG (1 punto)

**Nota:** Podemos consultar como utilizar `round()` en la documentación de Python, o en la siguiente referencia: <https://stackoverflow.com/questions/13479163/round-float-to-x-decimals/22155830>

[13]: # Respuesta:

#### 1.0.4 Ejercicio 4

Alguna noche habremos mirado atentamente al cielo, observando las estrellas. Quizás también hayamos visto, sin saberlo, alguno de los satélites que orbitan alrededor de la Tierra. En Enero del 2020 existían más de 2500. En el fichero ‘UCS-Satellite-Database.csv’, podemos encontrar alguna información básica sobre ellos. Esta información ha sido obtenida de la siguiente web <https://www.ucsusa.org/resources/satellite-database>.

El fichero con el formato Comma Separated File (CSV) tiene tres columnas de información, separadas entre ellas con un punto y coma, como podéis ver si visualizáis el fichero en un editor de texto. La primera fila contiene la cabecera, que define cada una de las columnas. La primera columna define el tipo de uso del satélite, la segunda su nombre, y la última, el país del propietario.

- a) Abre el fichero, recórrelo fila a fila, y obtén el número total de satélites, y cuántos de ellos son de uso exclusivamente Civil. NM (1.5 puntos)

[14]: # Respuesta:

- b) Escribe esta vez en un segundo fichero, que se llame ‘Civil-Satellite-Database.csv’, la segunda y tercera columna (Nombre y País) de los satélites de uso exclusivamente Civil. Visualiza el fichero creado tras ejecutar el código, para asegurarte que contiene la información y formato esperado. NM (1.5 puntos)

[15]: # Respuesta:

- c) Realiza de nuevo el apartado a) pero esta vez utilizando alguna librería de python que automatice la lectura del fichero csv. EI (OPCIONAL)

**NOTA:** Puedes consultar en <https://docs.python.org/3.7/library/csv.html>

[16]: # Respuesta: