prog_datasci_5_api_entrega

November 15, 2021

0.1 Unidad 5: Adquisición de datos en Python

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, puedéis 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

Hemos visto el uso de la librería Requests para realizar peticiones a la Web API de manera manual.

Mediante esta librería podemos realizar solicitudes como en el ejemplo que hemos visto de post-codes.io.

Hemos visto que, al hacer una petición a una web o API, recuperamos un objeto que contiene, entre otros, los siguientes atributos: `status.code`,`content`, `headers`.

Implementa una función de forma que:

1. acepte una *url* y, en caso de que el `status.code` sea correcto, imprima el nombre de cada uno de los `header` junto con su contenido.
2. En caso de que el `status.code` contenga un código de error, no debe imprimir los `header` sino el código de error recibido.
3. La función debe retornar el código de error a la salida.

Prueba la función con las siguientes URLs:

- http://wikipedia.org
- http://google.com/thehearthisflat

(1 punto) EI

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

1.0.2 Ejercicio 2

Para cada una de las siguientes direcciones:

1. https://cat-bounce.com
2. https://www.boredapi.com/api/

mediante los servicios de `requests`, muestra:

- La longitud del contenido de la dirección web.
- El contenido de la dirección web, limitando la salida hasta 1000 caracteres.

Accede a cada una de las direcciones con tu ordenador y observa el contenido en el navegador con el texto de la web. ¿Cuál es la diferencia en el contenido de la segunda dirección respecto a las primera?

(1 punto) NM

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

1.0.3 Ejercicio 3

En este ejercicio lucharemos un poco contra el aburrimiento. Hay una API para justamente esto en la dirección https://www.boredapi.com, podéis ver la documentación de la API en https://www.boredapi.com/documentation.

Por ejemplo, dada una llamada a esta dirección, obtenemos una actividad de forma aleatoria:

Call | Description |

|: ———: |: ———————————— ———————————————————— ———— | |
http://www.boredapi.com/api/activity/ | Random activity |

a) Usa requests para llamar a la dirección anterior 100 veces. ¿Cuántas actividades aparecen para dos o más participantes?

(1 punto) NM

[1]:
```
# Respuesta
```

b) La documentación de la API indica que es posible hacer búsquedas incluyendo parámetros, como por ejemplo via la ruta:

/api/activity?minprice=:minprice&maxprice=:maxprice

Escribe una función llamada `fun` que:

1. Tenga dos parámetros de entrada (`minprice`, `maxprice`).
2. Devuelva un diccionario solo con los elementos `price`, `activity` y `type`.

Llama a esta función otras 100 veces y comprueba que todos los precios se ajustan a los valores de la llamada.

(1 punto) NM

[1]:
```
# Respuesta
```

1.0.4 Ejercicio 4

Una del espacio.

SWAPI es la Star Wars API, quizás la primera fuente de datos accesible de forma programática sobre el universo Star Wars. Esta API no requiere autenticación y está ubicada en la URL: `https://swapi.dev/api/`.

Puedes consultar la documentación de la API en https://swapi.dev/documentation. Verás que la API permite consultar sobre los siguientes *atributes* (en inglés):

- **films** – The URL root for Film resources
- **people** – The URL root for People resources
- **planets** – The URL root for Planet resources
- **species** – The URL root for Species resources
- **starships** – The URL root for Starships resources
- **vehicles** – The URL root for Vehicles resources

Por una parte, usando la url `https://swapi.dev/api/<atribute>` podemos interrogar a la API sobre todos los elementos de un atributo. Si añadimos un `/<id>` a la url, podemos interrogar sobre un elemento en particular.

Así, por ejemplo, `https://swapi.dev/api/<atribute>/<id>/`, con el *slug* de *atribute* igual a `people`, y el *slug* de `id` igual a 1, devolverá la información relativa al personaje con identificador 1.

a) Obtén una variable con una cadena de texto o `string`, que contenga solo el nombre del personaje número 1 obtenido de forma programática mediante el acceso a la API via `requests`.

Nota: Se denomina usualmente un slug como una parte de una url, o un nombre corto de una url. El término proviene del mundo editorial y se usa extensivamente en internet. Puedes buscar más información en https://en.wikipedia.org/wiki/Slug_(publishing)

(0.5 puntos) EG

```
[ ]: # Respuesta
```

b) Construye una lista con la información de todos los personahes. Organiza tu código con funciones que te ayuden a realizar las solicitudes. Ten en cuenta que los resultados pueden estar paginados (solo se muestra una parte de resultados en cada solicitud).

¿Cuántos personajes has recuperado?

(1.5 puntos) EG

```
[ ]: # Respuesta
```

c) Construye un objeto dataframe de pandas que contenga, `people`, `species`,`classification` (de `species`),`designation` (de `species`) y `films` a partir de la información de la API. Ten en cuenta que :

- Crea tantas filas como necesites (e.g. si un personaje aparece en dos films).
- Usa el nombre de cada atributo (e.g. el nombre de propietario) para el contenido del dataframe, y no su identificador o url.
- Muestra por pantalla las primeras y las últimas 5 entradas del dataframe.

(1.5 puntos) NM

```
[ ]: # Respuesta
```

1.0.5 Ejercicio 5

En los ejercicios anteriores, hemos utilizado directamente una API para hacer solicitudes a servicios en línea, y nos encargamos directamente de la gestión de los datos de salida.

Sin embargo, también hemos visto en los materiales del curso el uso de librerías que facilitan el acceso a una API, como *tweepy*.

La mayoría de estas librerías (y la mayoría de APIs de proyectos populares) requieren un registro en una web de desarrolladores o API Key.

En este ejercicio os proponemos el uso de geopy https://geopy.readthedocs.io en conjunción con una API, Open Notify, que contiene la información tanto sobre los humanos residentes fuera de la tierra (es decir, en el espacio) como de la posición de la estación internacional Espacial. Open Notify.

Programa una función que obtenga la longitud y latitud de la posición sobre la que está volando la ISS. Calcula, de forma aproximada su velocidad (suponiendo que estuviera en la superficie de la

tierra), obteniendo dos localizaciones consecutivas dentro de un intervalo de tiempo, y calculando la distancia entre estas.

Puedes guiarte con estas dos funciones:

- la función `sleep` del módulo `time`, para pausar un número de segundos.
- la función `distance` del module `geopy.distance` para calcular la distancia entre dos localizaciones en el globo.

(1.5 puntos) EG

[ ]:
```
# Resultado
```

1.0.6 Ejercicio 6

El New Mexico Tech Seismological Observatory almacena un registro de eventos sísmicos recientes. Este registro está disponible, por ejemplo en esta web.

- https://geoinfo.nmt.edu/nmtso/events/home.cfml

Usa `scrappy` para mostrar la información del listado por pantalla de las "Date+Time (UTC)" de cada evento de la página.

Para ello:

- Utiliza el tutorial de Scrappy para encontrar un `xpath` que contenga la información requerida.
- Muestra la información requerida en forma de diccionario.

(1 punto) EI

[ ]:
```
# Respuesta
```

1.0.7 Ejercicio opcional

La NASA mediate su API publica cada día una imagen astronómica.

Implementa una función para descargar la imagen y visualizarla dentro del mismo notebook. Deberás obtener una clave para usar la API de la NASA.

EI

[ ]:
```
# Respuesta
```