

Q&A - tutorial VO with Python

Preguntas de la anterior escuela

Q: ¿Debería preocuparnos el “Deprecation warning”?

A: En nuestro caso sí que lo podemos ignorar. Este tipo de error se utiliza para marcar funcionalidades, dentro de una librería de python, que pueden sufrir cambios o ser reemplazadas en futuras versiones (suele ser más para informar a futuros desarrolladores de las librerías). En nuestro caso, con las versiones que se utilizan para los paquetes, no tiene importancia.

Q: ¿Es el orden importante en la intersección? (en la sección “working with MOCs in Python”)

A: El orden en el que se aplique la intersección es irrelevante, siempre te calcula la intersección entre los dos MOCs.

Posibles preguntas

Q: ¿Por qué utilizamos la versión 0.8.5 del paquete MOCPy (en el archivo requirements.txt)?

A: A partir de esta versión, la funcionalidad de importar un MOC desde un fits da problemas para algunos fits.

Q: ¿Qué es un “schema” en una base de datos?

A: El término “schema” es ampliamente utilizado cuando hablamos de bases de datos relacionales. En nuestro caso, cada “schema” representa una estructura dentro de la base de datos que incluye entidades que se encuentran en un mismo contexto (por ejemplo, un data release concreto), y cómo se relacionan entre sí.

Q: DataFrames en Python

A: Un DataFrame es una estructura de datos de dos dimensiones, en la que podemos guardar datos de distintos tipos, sobre la que podemos realizar una gran variedad de operaciones analíticas. Para trabajar con DataFrame en Python utilizamos la librería Pandas.

Q: Diferencia entre clases, métodos y atributos en Python

A: En Python, las clases son las que nos permiten crear un nuevo tipo de objeto (Python es un lenguaje orientado a objetos), y las podemos entender como moldes a partir de los

cuales podemos crear un tipo determinado de objeto. Las clases definen los atributos y métodos de los objetos que se crean a partir de ella.

Un atributo es una variable que únicamente existe dentro de un objeto. Se puede consultar de la siguiente forma: `gaia_data.fieldnames`. Aquí, consultamos el atributo `fieldnames` del objeto `gaia_data`

Un método es una función que forma parte de un objeto. Se puede aplicar de la siguiente forma: `gaia_data.getdesc()`. Aquí, aplicamos la función (método) `getdesc()` del objeto `gaia_data`.

Q: Diferencia entre comillas simples, dobles y triples en Python

A: En Python, no existe ninguna diferencia entre usar comillas simples y dobles a la hora de definir una *string*. Sin embargo, podemos utilizar las comillas dobles para definir una *string* que contiene una comilla simple, y viceversa.

Las comillas triples se utilizan para definir una *string* que contiene tanto comillas simples como dobles, o para definir una *string* en varias líneas.

Q: ¿Se pueden realizar más operaciones lógicas entre MOCs con el paquete MOCPy, aparte de unión e intersección?

A: Sí, también podemos calcular la diferencia entre dos MOCs (`moc1.difference(moc2)`), u obtener la cobertura de cielo complementaria de un MOC (`moc.complement()`). En este link está la documentación de la clase MOC (se puede dirigir aquí al alumno para que lo consulte él mismo):

<https://cds-astro.github.io/mocpy/stubs/mocpy.MOC.html#mocpy.MOC>

Q: ¿Dónde podríamos consultar todos los servicios TAP disponibles?

A: En la página del Euro-VO Registry se pueden ver todos los servicios TAP, así como las URLs de acceso correspondientes. Para consultarlo, ir a

<https://registry.euro-vo.org/evor/#adql> y seleccionar la opción "Find all TAP services; return their accessURLs"