



ProfesorenVideo.com

Terminología básica en Python

Diferencias entre Módulo, Paquete y Sub-paquete:

Módulo: Un módulo en Python es un archivo individual que contiene código Python. Puede contener **variables**, **funciones**, **clases** y **otros elementos** de código relacionados. Los módulos se utilizan para organizar y reutilizar el código de manera eficiente. Se importan con **import**.

Paquete: Un paquete en Python es un **directorio** que contiene uno o más módulos.

Sub-paquete: Un sub-paquete en Python es un paquete dentro de otro paquete. Es simplemente un paquete que está ubicado dentro de otro paquete. Los sub-paquetes se utilizan para organizar aún más el código en jerarquías más profundas.



ProfesorenVideo.com

Terminología básica en Python

Cuántos tipos de paquetes existen ?: Existen tres(3) tipos de paquetes en Python, a saber **Paquetes estándar**, **paquetes de terceros** y **paquetes de usuario**.

Paquetes estándar: Vienen incluidos con la instalación estándar de Python. Pueden requerir , o no, el uso de **import**.

Paquetes de terceros: Estos son paquetes desarrollados por la comunidad de Python y otros desarrolladores externos. No vienen incluidos con la instalación estándar de Python y deben ser instalados por separado utilizando administradores de paquetes como "**pip**".

Paquetes de usuario: Estos son paquetes que son creados por los propios usuarios de Python para su uso personal o para compartir con otros.



ProfesorenVideo.com

Terminología básica en Python

Bueno, ya se lo que es un **paquete** , un **sub-paquete** y un **módulo** , pero **qué será entonces una Biblioteca?**: Realmente el término Biblioteca es más amplio, ya que contiene a los demás. Ejemplo:

Biblioteca:

- * Paquetes:
 - + Sub-paquetes:
 - Módulos.
 - + Módulos.
- * Módulos

Nota: Hay la biblioteca estándar y de terceros.



ProfesorenVideo.com

Terminología básica en Python

Y cuándo debo usar el **import** y cuándo **NO** ?

- Las **funcionalidades integradas**(o **incorporadas**) **NO** requieren el uso del **import**. A estas funcionalidades se les llama comúnmente “funciones integradas” o “funciones incorporadas”. Ejemplos de estas funciones son: **print()**, **len()**, **input()**, **range()** .
- Los **módulos estándar**, los **módulos de usuario** y los **módulos de terceros** **SÍ** requieren el uso del **import**. Ejemplos ...

Sistema de Importación en Python

Ejemplos de módulos y funciones que requieren ser **importados**:

- `import numpy as np` # de terceros
- `Import os` # incorporada
- `import matplotlib.pyplot as plt` # de terceros
- `from math import sqrt` # incorporada

Sistema de Importación en Python

Explicando la instrucción de Importar: **import**

Sintaxis general: Hay dos formas básicas para importar

- **import** modulo as alias
- **from** modulo **import** identificador as alias

✓ **import** numpy as np

✓ **Import** os

✓ **import** matplotlib.pyplot as plt # paquete.modulo

✓ **from** math **import** sqrt

Sistema de Importación en Python

Explicando la instrucción de Importar: **import**

Cuándo usar **from** en lugar del **import** directo?: Haremos uso del **from** para evitar escribir la ruta completa del elemento a utilizar

```
paquete.modulo.funcion()  
paquete.modulo.variable  
paquete.modulo.Clase()
```

```
from math import sqrt
```

```
resultado = sqrt(25)
```

Vs

```
import math
```

```
resultado = math.sqrt(25)
```



ProfesorenVideo.com

Muchas gracias por su atención!