

*Creado por:*

*Isabel Maniega*

```
In [2]: from IPython.display import Image
```

## Jython

Jython y Python son dos versiones del mismo lenguaje, que se utilizan en contextos diferentes. Jython es una implementación de Python en Java, lo que, en pocas palabras, significa que es Python ejecutándose en un entorno de máquina virtual Java (JVM). Se escribe como Python, pero puede acceder a todo el potencial de las bibliotecas de Java.

Veremos las diferencias entre Jython y Python y por qué Jython se está volviendo cada vez más popular entre los desarrolladores de Java y Python, quienes están aprendiendo a usar esta implementación en el contexto de JVM para mejorar la productividad y lograr resultados más rápidos.

En lugar de debatir cuál es mejor, ya que comparten las mismas características principales, mostraremos cómo conectar Python a Java a través de Jython abre un mundo completamente nuevo de posibilidades.

A continuación se muestran las diferencias entre Python y Jython:

- La implementación de referencia de Python, llamada CPython, está escrita en lenguaje C. Jython, por otro lado, está completamente escrito en Java y es una implementación de JVM.
- Python estándar está disponible en varias plataformas. Jython está disponible para cualquier plataforma que tenga una JVM instalada.
- El código Python estándar se compila en un archivo .pyc , mientras que el programa Jython se compila en un archivo .class.
- Las extensiones de Python se pueden escribir en lenguaje C. Las extensiones para Jython se escriben en Java.
- Jython es verdaderamente multiproceso por naturaleza. Sin embargo, Python utiliza el mecanismo de bloqueo de intérprete global (GIL) para este propósito.

Ambas implementaciones tienen diferentes mecanismos de recolección de basura.

### Beneficios de Jython

Ahora que hemos establecido qué es Jython, qué no es y para qué sirve, veamos por qué es una opción tan atractiva para los desarrolladores de todo el mundo. Jython no solo cierra la brecha entre las posibilidades que ofrecen Python y Java, sino que también crea otras nuevas.

Es un lenguaje de programación fácil de aprender e implementar que tiene mucha potencia gracias a la enorme cantidad de librerías Java que trae. Se puede utilizar para crear Interfaces Gráficas de Usuario rápidas, llamar a una base de datos, crear frames o probar el código rápidamente en busca de errores.

Visualmente, incluso luce mejor. Al igual que Python, Jython se basa en sangrías y espacios en lugar de corchetes para dividir bloques de código y definir la estructura, lo que limita la presencia de símbolos innecesarios en el código.

### **Python + Java= Jython**

Por lo tanto, no se trata de una situación de Python contra Jython, sino de una combinación ganadora de características que convierte a Jython en una poderosa combinación de Python y Java. Jython combina la ligereza y versatilidad de Python con el potencial del universo Java, lo que lo convierte en una opción seria para los desarrolladores que buscan introducirse en las aguas de los lenguajes de alto nivel utilizando una sintaxis más simple.

Ya sea para desarrolladores de Java o Python, puede valer la pena considerar Jython para ampliar las capacidades de programación y crear aplicaciones potentes con una sintaxis elegante.

<https://www.jython.org/download.html>

<https://repo1.maven.org/maven2/org/python/jython-installer/2.7.4/>

## **Instalación**

Descargamos el instalador:

```
In [2]: Image(filename='./images/instalador.png')
```

Out[2]:

## org/python/jython-installer/2.7.4

./			
<a href="#">jython-installer-2.7.4-javadoc.jar</a>	2024-08-18 19:45	3659818	
<a href="#">jython-installer-2.7.4-javadoc.jar.asc</a>	2024-08-18 19:45	499	
<a href="#">jython-installer-2.7.4-javadoc.jar.asc.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4-javadoc.jar.asc.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4-javadoc.jar.asc.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4-javadoc.jar.asc.sha512</a>	2024-08-18 19:45	128	
<a href="#">jython-installer-2.7.4-javadoc.jar.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4-javadoc.jar.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4-javadoc.jar.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4-javadoc.jar.sha512</a>	2024-08-18 19:45	128	
<a href="#">jython-installer-2.7.4-sources.jar</a>	2024-08-18 19:45	17779360	
<a href="#">jython-installer-2.7.4-sources.jar.asc</a>	2024-08-18 19:45	499	
<a href="#">jython-installer-2.7.4-sources.jar.asc.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4-sources.jar.asc.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4-sources.jar.asc.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4-sources.jar.asc.sha512</a>	2024-08-18 19:45	128	
<a href="#">jython-installer-2.7.4-sources.jar.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4-sources.jar.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4-sources.jar.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4-sources.jar.sha512</a>	2024-08-18 19:45	128	
<a href="#">jython-installer-2.7.4.jar</a> ←	2024-08-18 19:45	95341227	
<a href="#">jython-installer-2.7.4.jar.asc</a>	2024-08-18 19:45	499	
<a href="#">jython-installer-2.7.4.jar.asc.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4.jar.asc.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4.jar.asc.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4.jar.asc.sha512</a>	2024-08-18 19:45	128	
<a href="#">jython-installer-2.7.4.jar.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4.jar.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4.jar.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4.jar.sha512</a>	2024-08-18 19:45	128	
<a href="#">jython-installer-2.7.4.pom</a>	2024-08-18 19:45	2120	
<a href="#">jython-installer-2.7.4.pom.asc</a>	2024-08-18 19:45	499	
<a href="#">jython-installer-2.7.4.pom.asc.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4.pom.asc.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4.pom.asc.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4.pom.asc.sha512</a>	2024-08-18 19:45	128	
<a href="#">jython-installer-2.7.4.pom.md5</a>	2024-08-18 19:45	32	
<a href="#">jython-installer-2.7.4.pom.sha1</a>	2024-08-18 19:45	40	
<a href="#">jython-installer-2.7.4.pom.sha256</a>	2024-08-18 19:45	64	
<a href="#">jython-installer-2.7.4.pom.sha512</a>	2024-08-18 19:45	128	

Acontinuación ejecutamos el archivo con:

```
java -jar jython-installer-2.7.4.jar
```

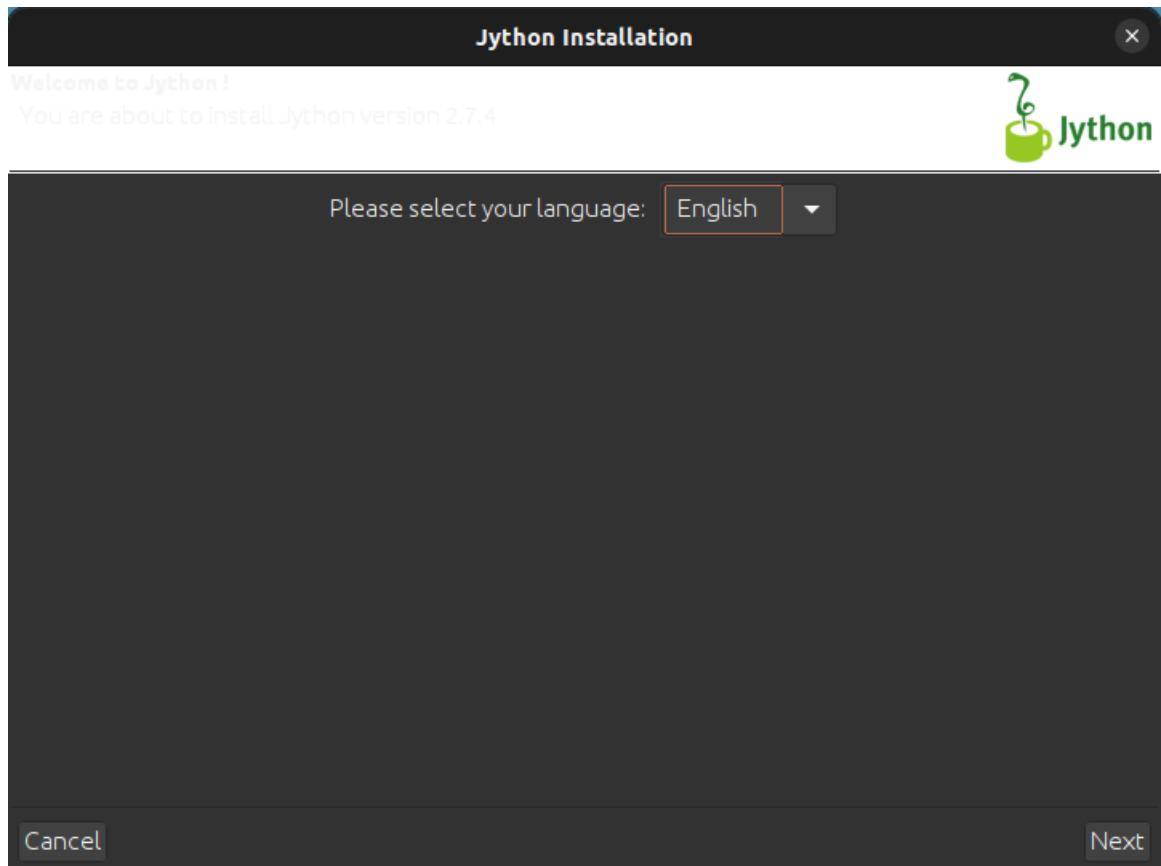
In [3]: `Image(filename='./images/command.png')`

Out[3]: `isabelmaniega@isabelmantega:~/Descargas$ java -jar jython-installer-2.7.4.jar`

Nos abre una aplicación, donde escogemos el idioma:

In [5]: `Image(filename='./images/Jython Installation_001.png')`

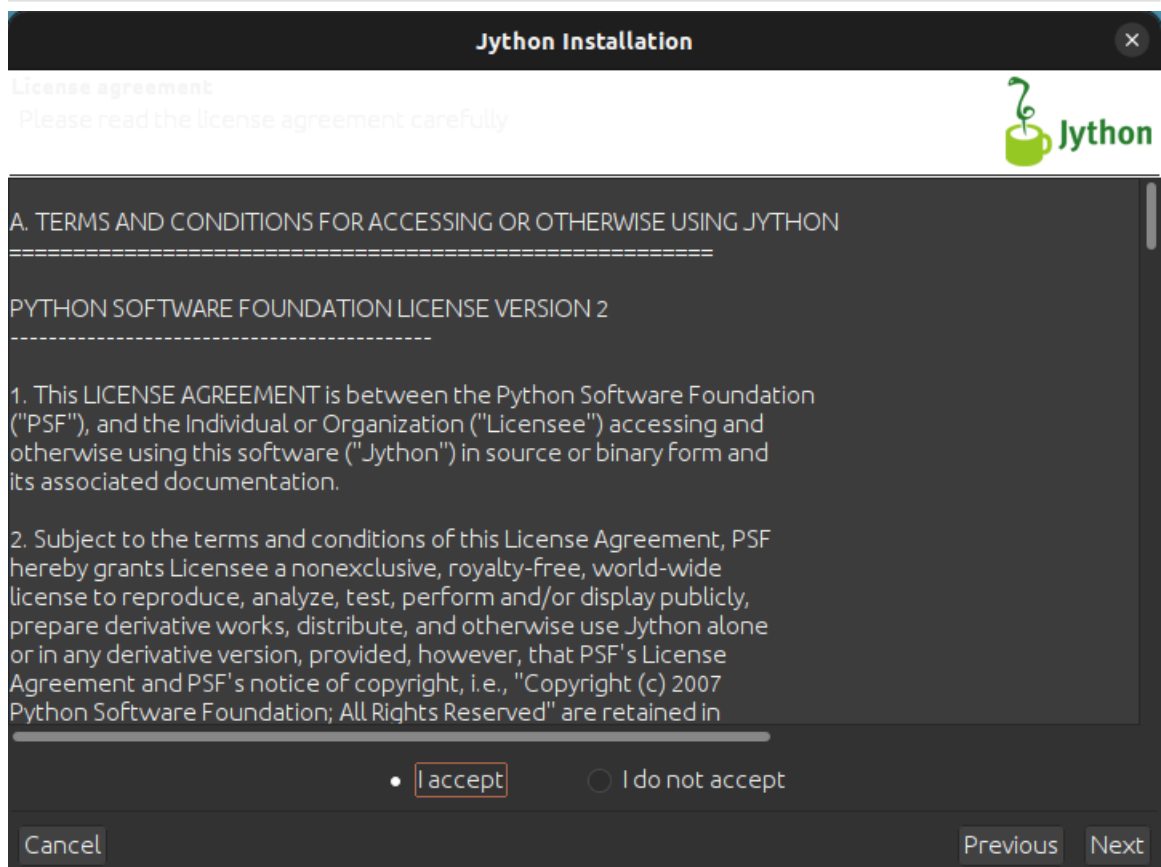
Out[5]:



Aceptamos la licencia y pulsamos Next:

```
In [6]: Image(filename='./images/Jython Installation_002.png')
```

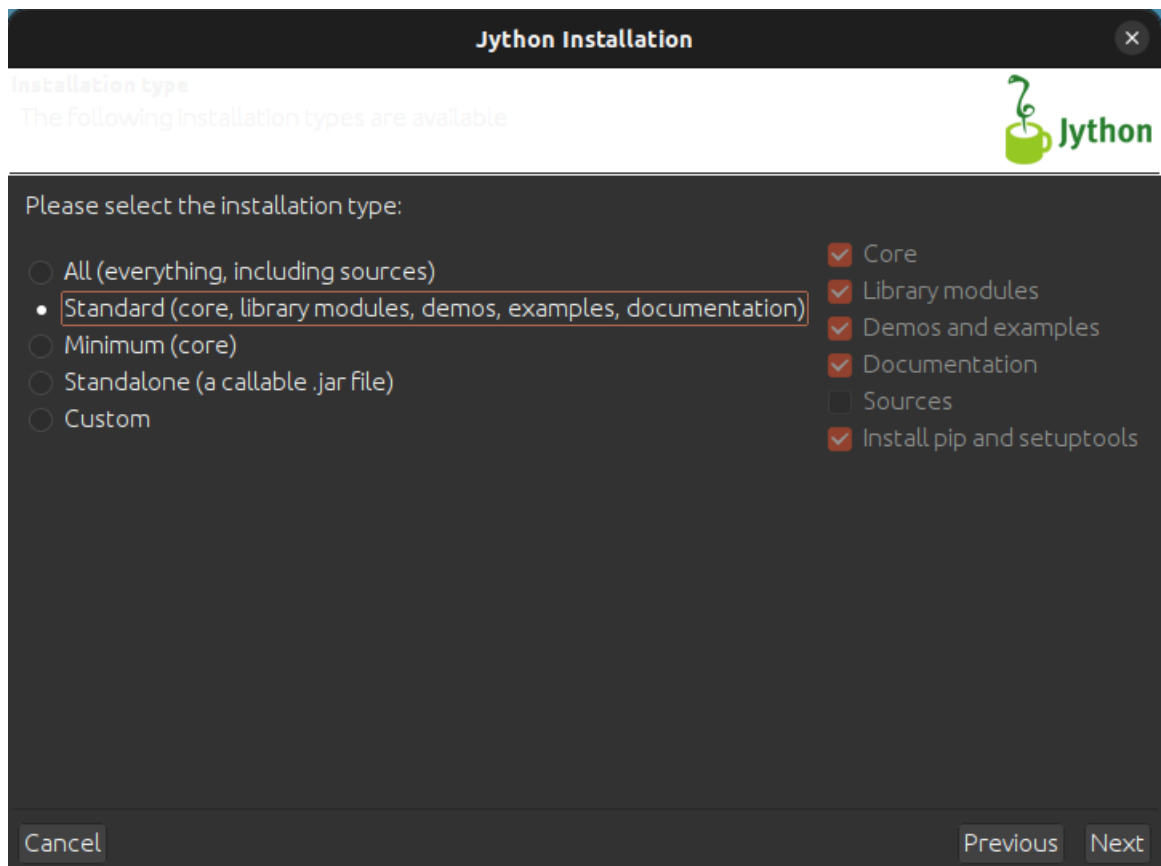
Out[6]:



Instalamos la versión estándar y pulsar Next:

```
In [7]: Image(filename='./images/Jython Installation_003.png')
```

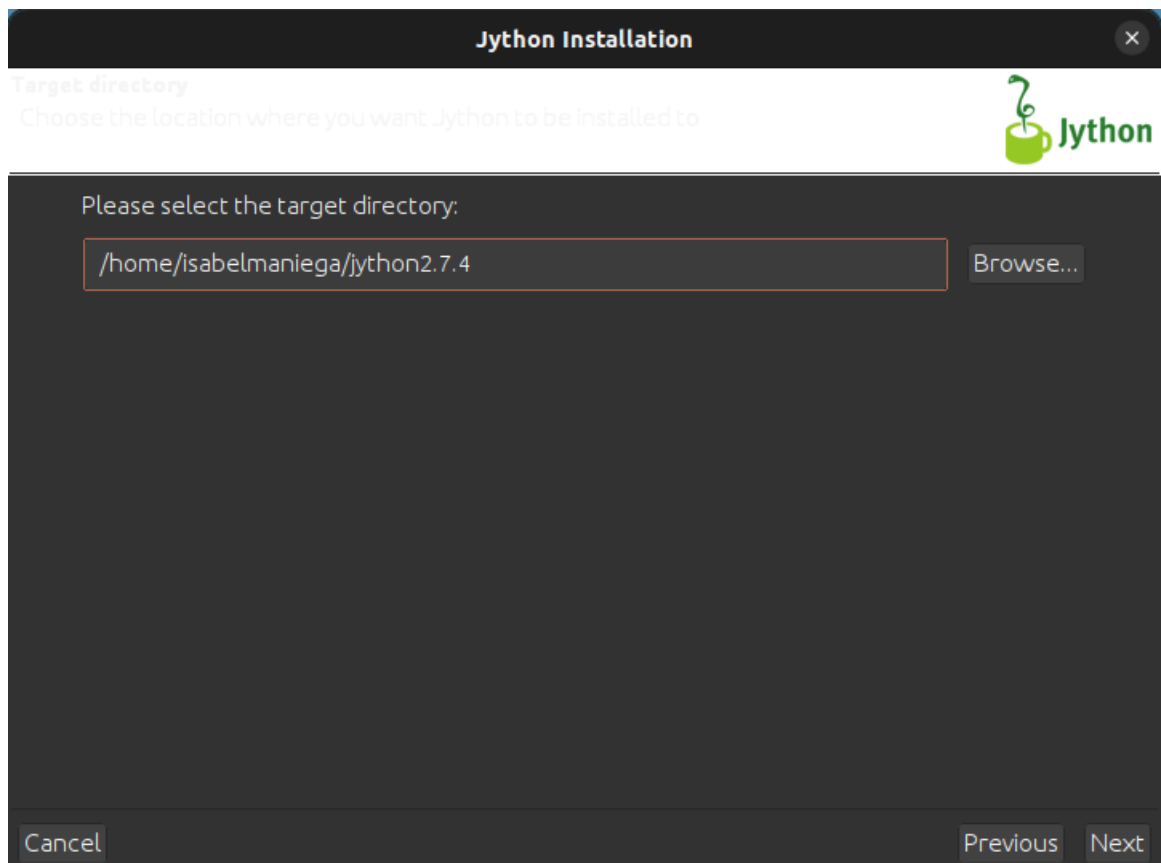
```
Out[7]:
```



Nos indica la ubicación de la instalación y pulsar Next:

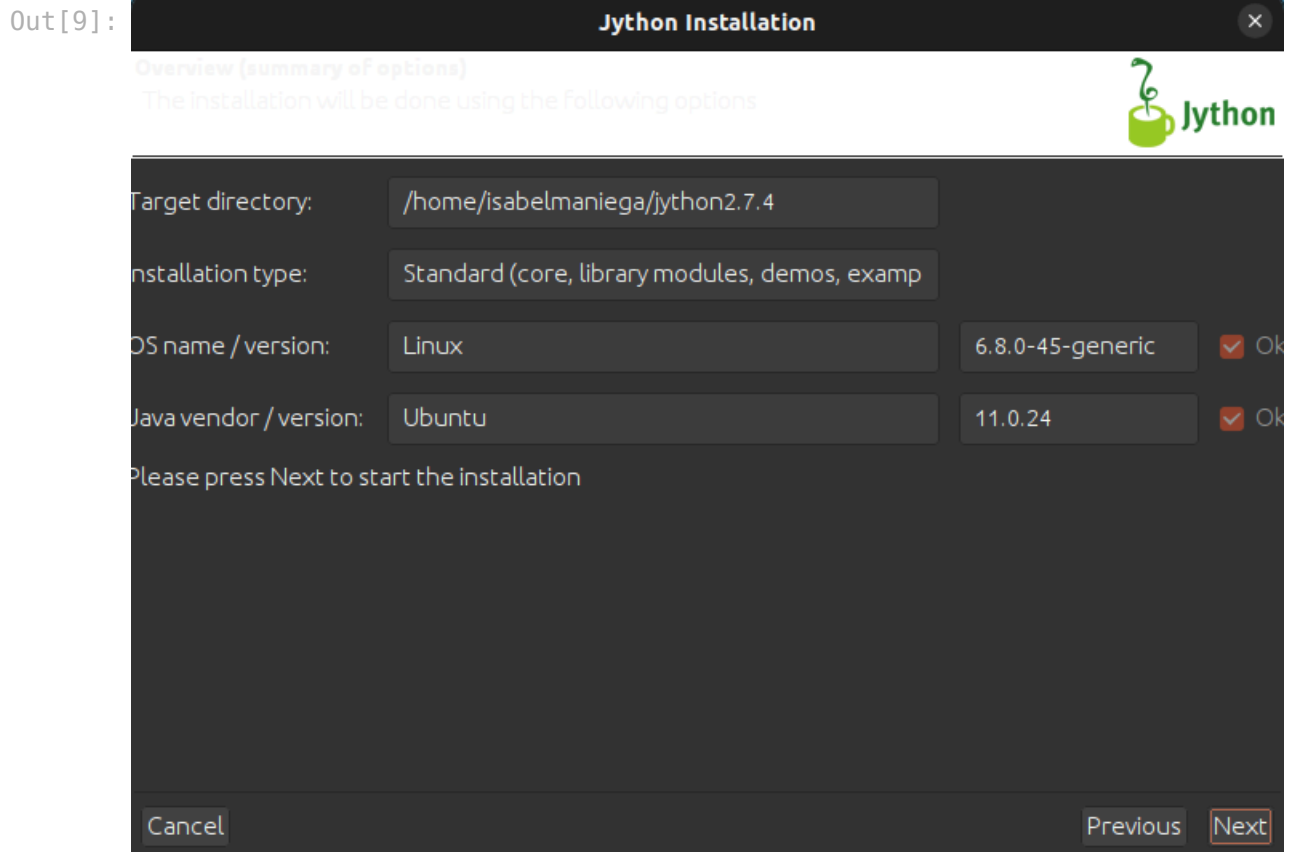
```
In [8]: Image(filename='./images/Jython Installation_004.png')
```

```
Out[8]:
```



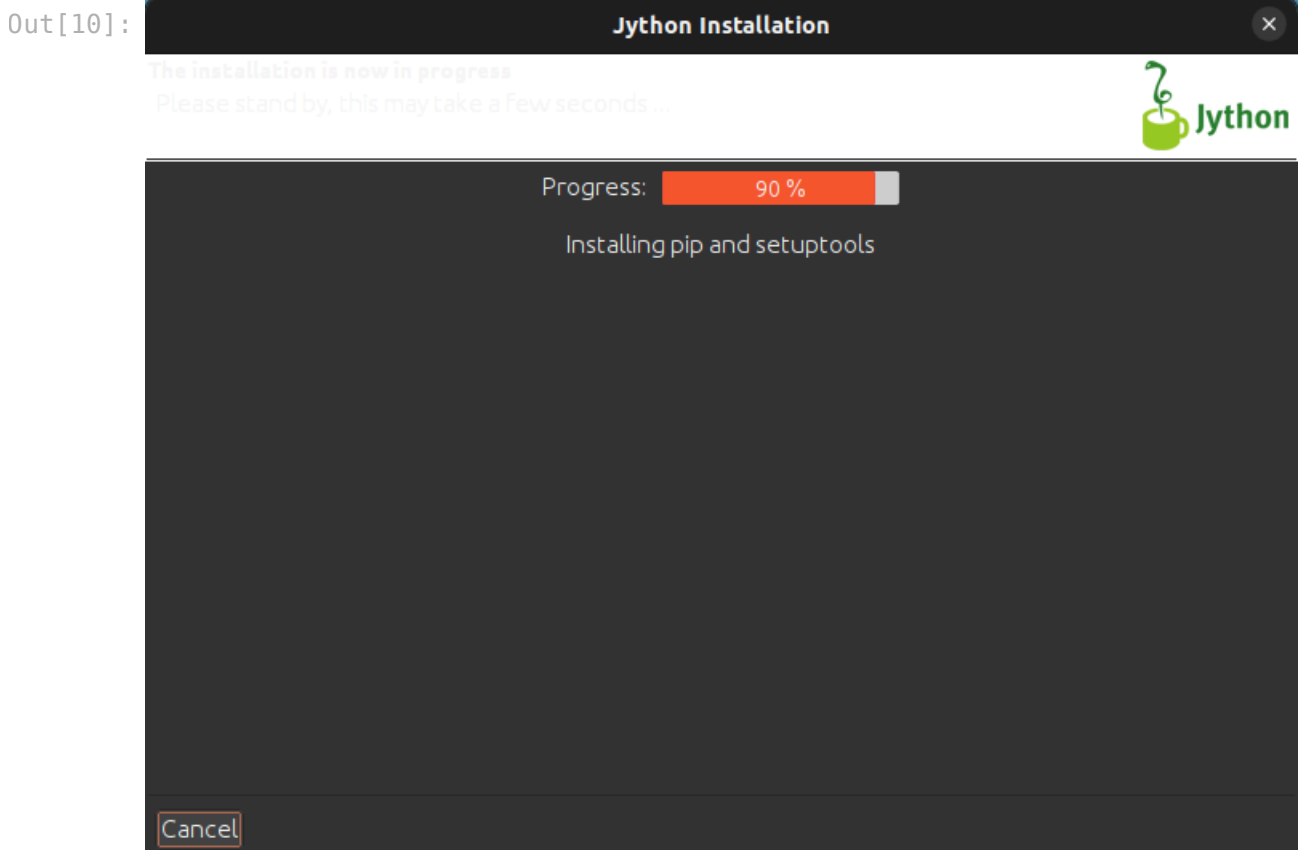
Nos indica la configuración final y pulsar Next:

In [9]: `Image(filename='./images/Jython Installation_005.png')`



Realiza la instalación...

In [10]: `Image(filename='./images/Jython Installation_006.png')`



No muestra información sobre la licencia y pulsar Next:

```
In [11]: Image(filename='./images/Jython Installation_007.png')
```

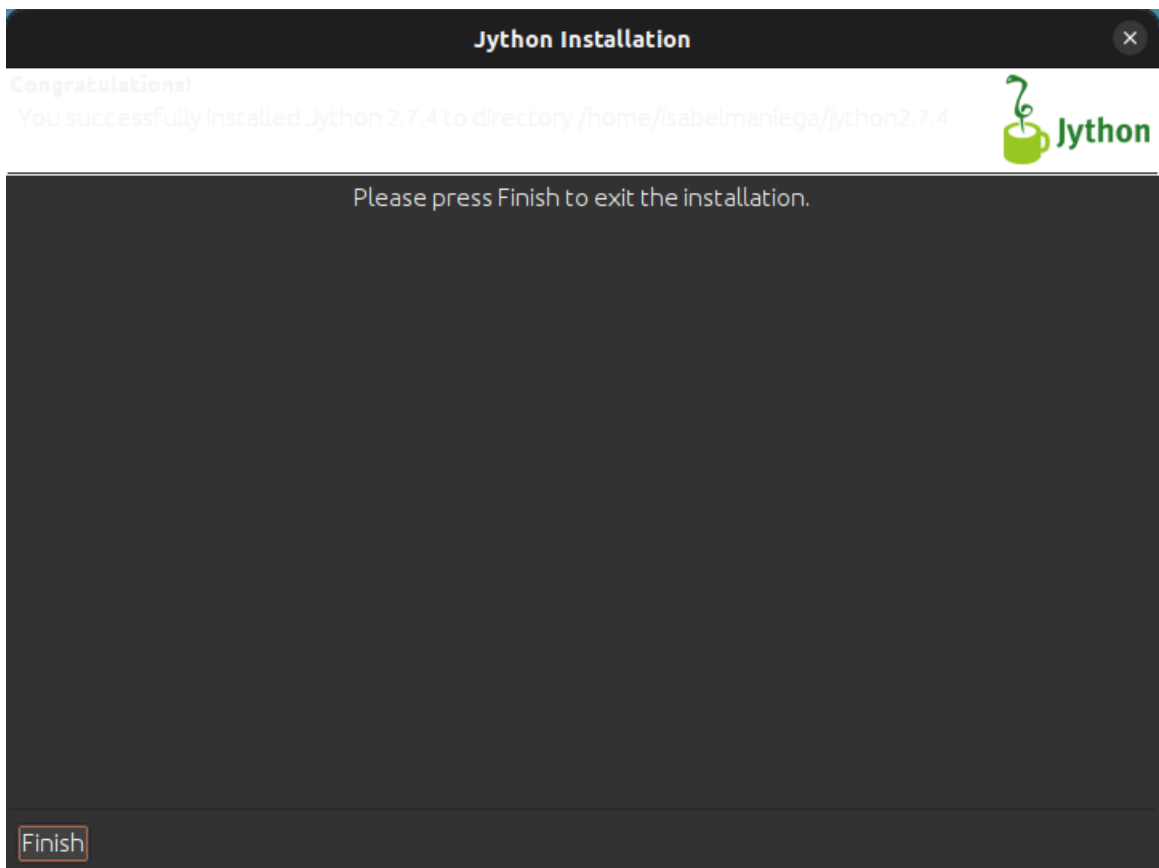
Out[11]:



Si es correcto pulsaremos finish:

```
In [12]: Image(filename='./images/Jython Installation_008.png')
```

Out[12]:



En el caso de **ubuntu** la instalación la realizamos con el comando:

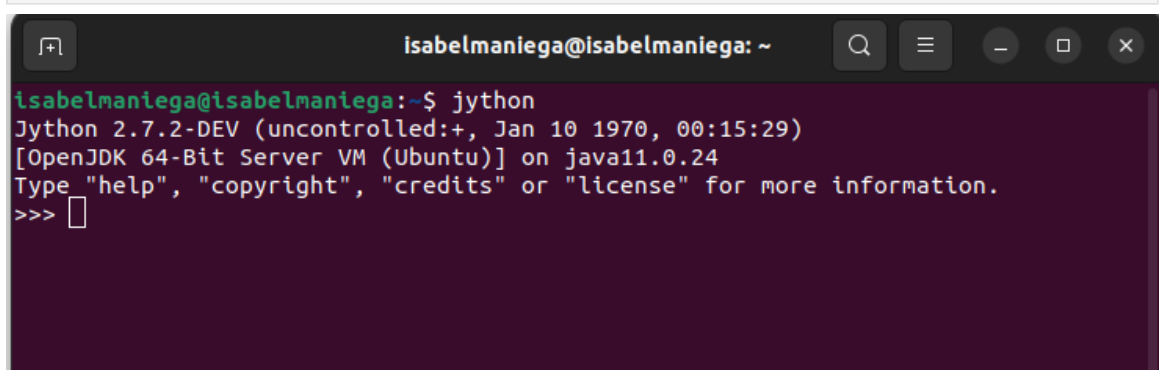
```
sudo apt-get install jython
```

Para comprobar que se instalado correctamente ponemos en el terminal:

```
jython
```

In [16]: `Image(filename='./images/jython.png')`

Out[16]:



## Importar librerías desde Java

Una de las características más importantes de Jython es su capacidad de importar clases Java en un programa Python. Podemos importar cualquier paquete o clase Java en Jython, tal como lo hacemos en un programa Java. El siguiente ejemplo muestra cómo se importan los paquetes `java.util` en un script Python (Jython) para declarar un objeto de la clase `Date`.



Creamos un archivo llamado utilDate.py

```
from java.util import Date

d = Date()
print(d)
```

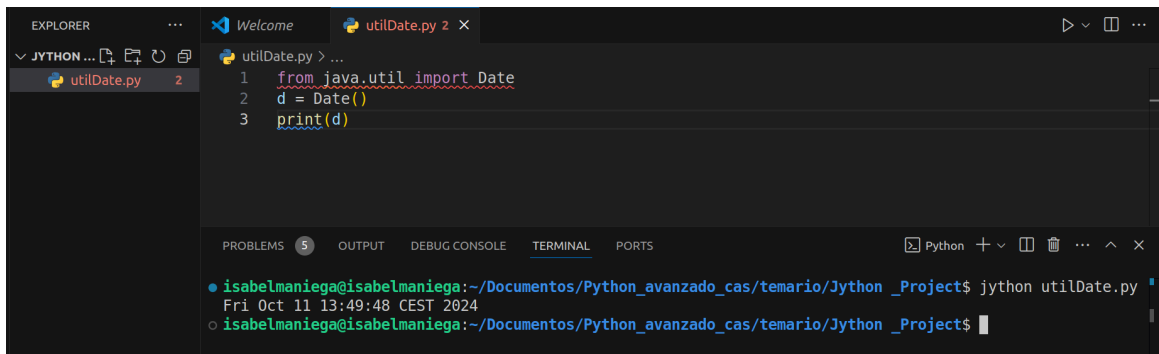
Guarda y ejecuta el código anterior como UtilDate.py desde la línea de comandos:

```
jython utilDate.py
```

Se mostrará una instancia de la fecha y la hora actuales:

In [3]: `Image(filename='./images/jython_1.png')`

Out[3]:



Los siguientes paquetes de la biblioteca Java se importan con mayor frecuencia en un programa Jython principalmente porque la biblioteca Python estándar no tiene sus equivalentes o no es tan buena. Como son:

- Servlets
- JMS
- J2EE
- Javadoc
- Swing se considera superior a otros kits de herramientas GUI

Cualquier paquete Java puede importarse en un script Jython. Aquí, el siguiente programa Java se almacena y compila en una carpeta (paquete) llamado foo:

```
# HelloWorld.java

package foo;
public class HelloWorld {
    public void hello() {
        System.out.println("Hello World!");
    }
    public void hello(String name) {
        System.out.printf("Hello %s!", name);
    }
}
```

In [4]: `Image(filename='./images/jython_2.png')`

Out[4]:

```

1 package foo;
2
3 public class HelloWorld {
4     public void hello() {
5         System.out.println("Hello World!");
6     }
7     public void hello(String name) {
8         System.out.printf("Hello %s!", name);
9     }
10 }

```

```

# importx.py

from foo import HelloWorld

h = HelloWorld()
h.hello()
h.hello("Everyone")

```

In [10]: `Image(filename='./images/jython_3.png')`

Out[10]:

```

1 from foo import HelloWorld
2
3 h = HelloWorld()
4 h.hello()
5 h.hello("Everyone")

```

Nos posicionamos en la carpeta foo y creamos el compilado de HelloWorld.java:

```
javac HelloWorld.java
```

In [11]: `Image(filename='./images/jython_4.png')`

Out[11]:

```

1 from foo import HelloWorld
2
3 h = HelloWorld()
4 h.hello()
5 h.hello("Everyone")

```

```

isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado_cas/temario/Jython_Project$ cd foo/
isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado_cas/temario/Jython_Project/foo$ javac HelloWorld.java
isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado_cas/temario/Jython_Project/foo$

```

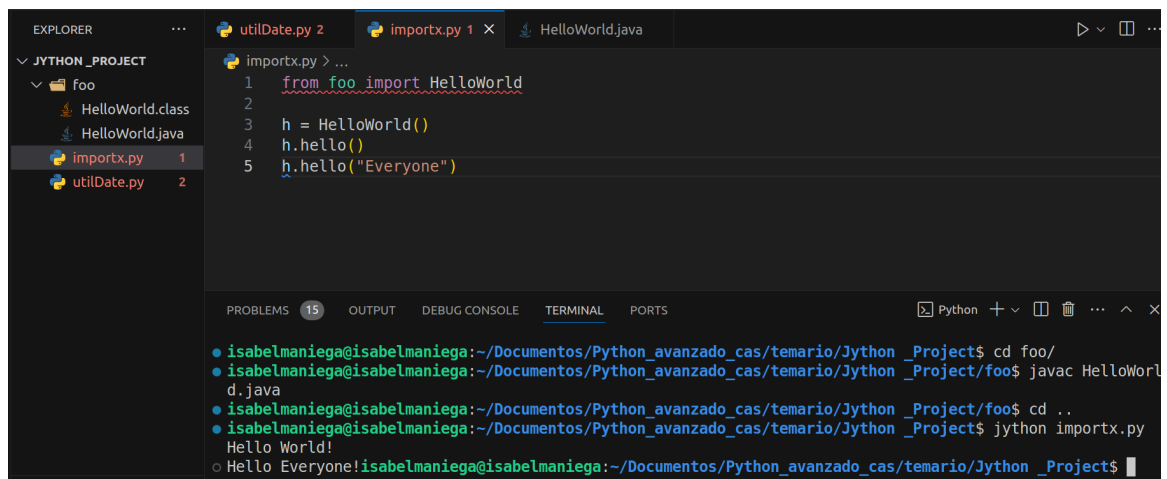
Nos ha creado un compilado llamado **HelloWorld.class**

Ahora volvemos a la carpeta de Jython\_project y ejecutamos el script importx.py:

```
jython importx.py
```

```
In [12]: Image(filename='./images/jython_5.png')
```

```
Out[12]:
```



Vemos como es capaz de ejecutar el código realizado en Java con Jython, devolviendo en primer lugar Hello World! y después recogiendo el valor de Everyone devolviendo: Hello Everyone!

## Tipos de variables en Jython

Las variables son ubicaciones con nombre en la memoria de la computadora. Cada variable puede contener un dato. A diferencia de Java, Python es un lenguaje de tipado dinámico. Por lo tanto, al utilizar Jython, no se realiza una declaración previa del tipo de datos de la variable. En lugar de que el tipo de variable decida qué datos se pueden almacenar en ella, los datos deciden el tipo de variable.

En el siguiente ejemplo, se asigna un valor entero a una variable. Con la función incorporada `type()`, podemos verificar que el tipo de variable sea un entero. Pero, si a la misma variable se le asigna una cadena, la función `type()` asignará la cadena como el tipo de la misma variable.

```
In [14]: Image(filename='./images/jython_6.png')
```

