

# Python 101

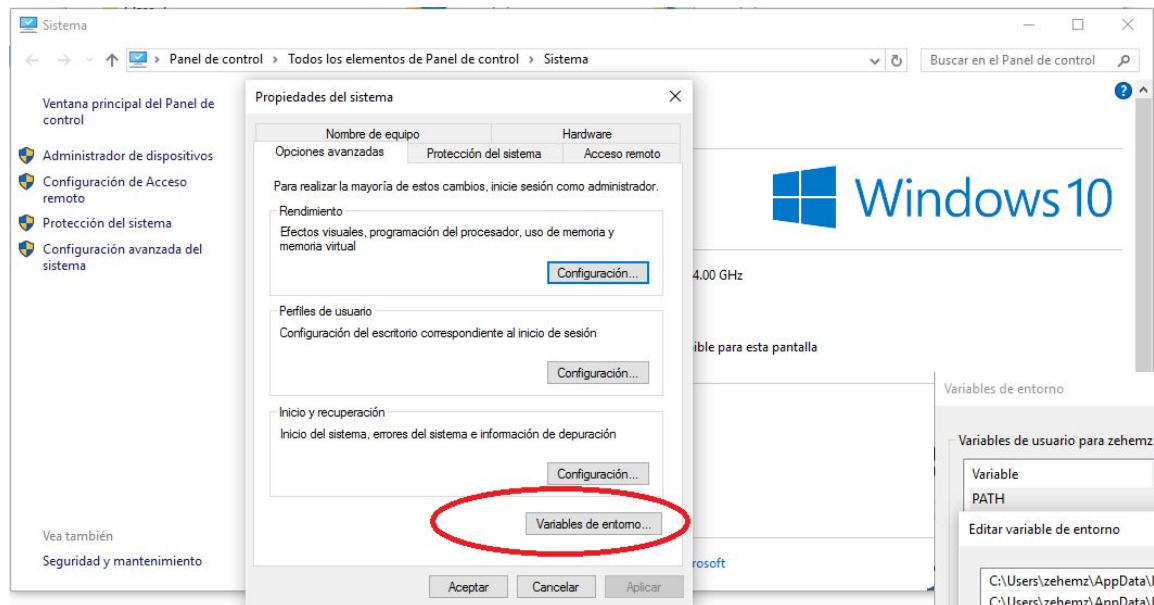
- + Entorno de desarrollo, Tipos de datos, `__main__`, funciones

# Instalar python en windows

Ingresa a python.org -> <https://www.python.org/>

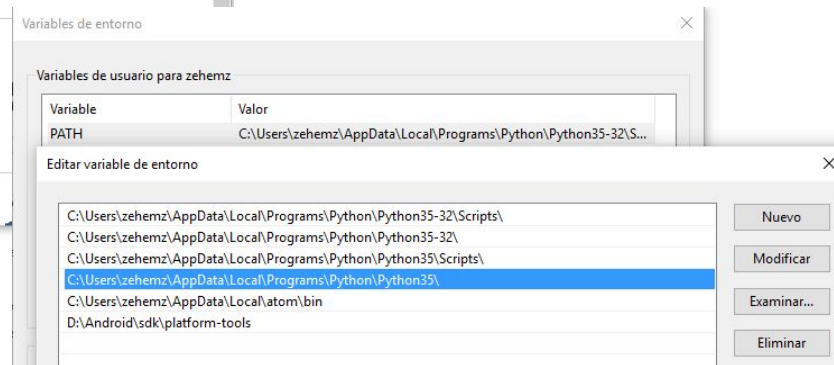


# Variables de entorno



The screenshot shows the Windows 10 'Sistema' (System) window in the Control Panel. The 'Propiedades del sistema' (System Properties) dialog box is open, displaying the 'Opciones avanzadas' (Advanced) tab. The 'Variables de entorno...' (Environment Variables) button at the bottom is circled in red.

Variables de entorno para usuario



The 'Variables de entorno' (Environment Variables) dialog box is shown, displaying the 'Variables de usuario para zehemz' (User variables for zehemz) tab. The 'PATH' variable is selected, and its value is displayed in the 'Valor' (Value) field. The 'Editar variable de entorno' (Edit environment variable) dialog box is also open, showing the list of paths in the 'Variable' field.

Variable	Valor
PATH	C:\Users\zehemz\AppData\Local\Programs\Python\Python35-32\Scripts\

Variable	Valor
C:\Users\zehemz\AppData\Local\Programs\Python\Python35-32\Scripts\	
C:\Users\zehemz\AppData\Local\Programs\Python\Python35-32\	
C:\Users\zehemz\AppData\Local\Programs\Python\Python35\Scripts\	
C:\Users\zehemz\AppData\Local\Programs\Python\Python35\	
C:\Users\zehemz\AppData\Local\atom\bin	
D:\Android\sdk\platform-tools	

# Ide - Eclipse

Requiere Java client (eclipse utiliza motor java para funcionar), pueden bajar el JDK de java aquí:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Bajar eclipse installer de: <https://eclipse.org/downloads/> (eclipse neon) y seleccionar “Eclipse IDE for Java Developers”



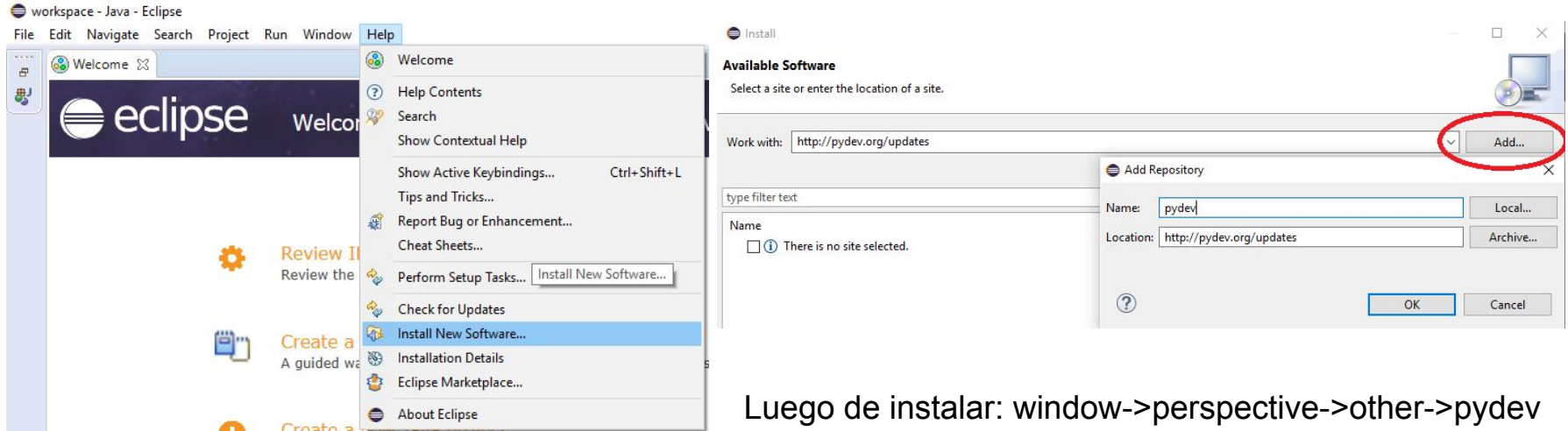
## Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

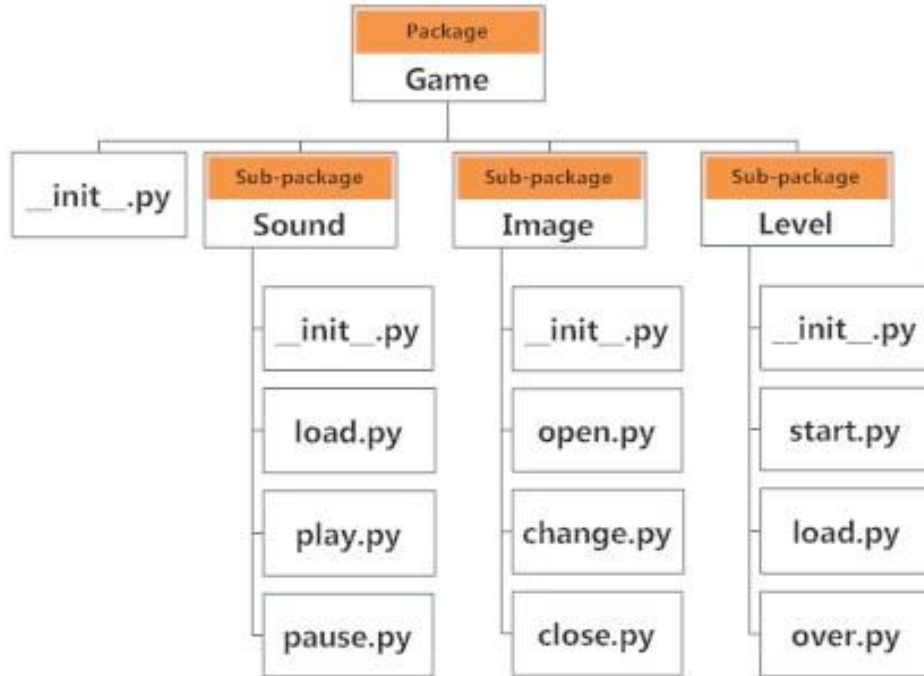
# PyDev Plugin para Eclipse

Pydev: <http://www.pydev.org/download.html>

Pydev url para eclipse plug-in: <http://pydev.org/updates>



# Módulos - **from** e **import**



# Concepto de `__main__`

El main es un espacio de escritura en el cual cada archivo .py tiene la posibilidad de ejecutar instrucciones al ser ejecutado, previo a cualquier línea anterior.

```
if __name__ == "__main__":
```

```
    ... #Código
```

Vamos a la consola!

# Funciones

```
def nombre(param1="def", param2, *argL, **argD):
```

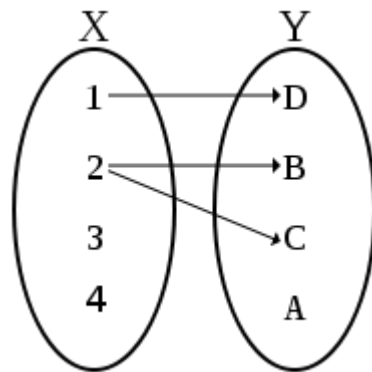
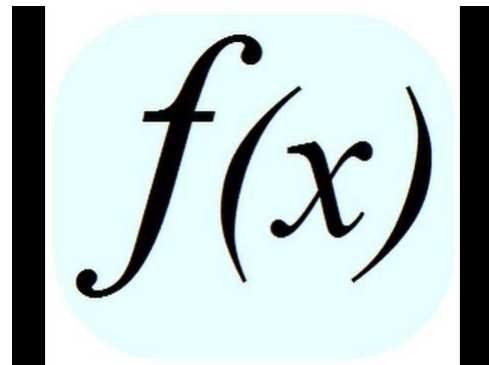
```
....code
```

```
....code
```

( **return** - **yield**?)

```
nombre(param2=10)
```

¿cómo se llaman las funciones que no retornan valores? p...





# Tuplas

Una tupla consiste de un número de valores separados por comas, por ejemplo:

```
t = 12345, 54321, 'hola!'
```

# Las tuplas pueden anidarse:

```
... u = t, (1, 2, 3, 4, 5)
```

# Listas

```
>>> cuadrados = [1, 4, 9, 16, 25]
```

```
>>> cuadrados
```

```
[1, 4, 9, 16, 25]
```

# Diccionarios

```
>>> tel = {'jack': 4098, 'sape': 4139}
```

```
>>> tel['guido'] = 4127
```

```
>>> tel
```

```
{'sape': 4139, 'jack': 4098, 'guido': 4127}
```

```
>>> tel['jack']
```

```
4098
```

```
>>> del tel['sape']
```

# Funciones como variables

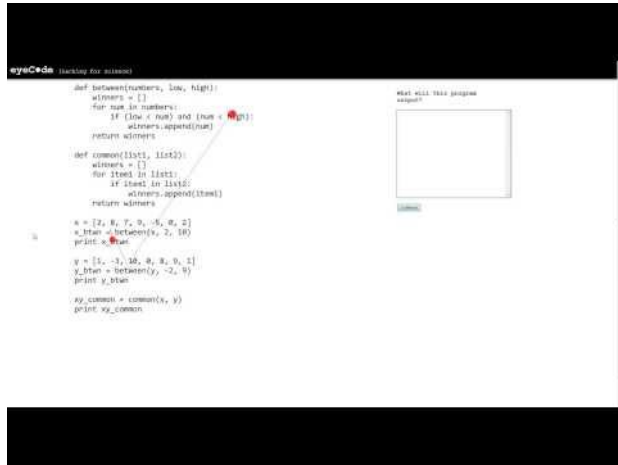
nuevo = nombre

nuevo() -> es la función nombre.

## Funciones de orden superior

```
def saludar(lang):  
    def saludar_es():  
        print ("hola")  
    def saludar_en():  
        print ("Hi")  
    def saludar_fr():  
        print ("Salut")  
  
    lang_func = {"es": saludar_es, "en": saludar_en, "fr": saludar_fr}  
    return lang_func[lang]
```

**Python style rules:** <https://www.python.org/dev/peps/pep-0008/>



**Utf-8 en python2:** # -\*- coding: utf-8 -\*-

**Entorno de interprete (para ejecutable, unix):** #!/usr/bin/env python



# Archivos y OS

¡Vamos a la consola!



*That's all Folks!*