



Instalando un paquete

Existen dos escenarios básicos a la hora de instalar un paquete, el primero es que se use un equipo de cómputo con un usuario que no tenga permisos de administrador y, el segundo escenario es que el usuario tenga cuenta con perfil de administrador.

Si se desea instalar algo para solo el usuario actual o no se cuentan con permisos de administrador, se debe utilizar la opción `--user` (doble guión), caso contrario, se omite dicha opción, esto instalará el paquete para todo el S.O.

También es posible con el comando `install` actualizar los paquetes, ya sea a la última versión registrada o a una versión específica.

Unset

```
//global
```

```
> pip install pygame
```

```
//para el usuario
```

```
> pip install --user pygame
```

```
//Actualizar el paquete a la versión más nueva
```

```
> pip install -U pygame
```

```
//Actualizar el paquete a la versión específica
```

```
> pip install -U pygame==1.9.2
```



```
C:\Users\jolgo> pip install pygame
Collecting pygame
  Downloading pygame-2.5.2-cp310-cp310-win_a
md64.whl.metadata (13 kB)
  Downloading pygame-2.5.2-cp310-cp310-win_amd
64.whl (10.8 MB)
    ━━━━━━━━━━━ 10.8/10.8 3.5 MB/s eta 0:00:00
    MB
Installing collected packages: pygame
Successfully installed pygame-2.5.2
```

Fuente: Elaboración propia

Para desinstalar un paquete se hace con el comando
uninstall.

Unset

```
> pip uninstall nombre_paquete
```

Ahora probando el paquete instalado, documentación del
paquete Pygame Front Page – pygame v2.6.0
documentation

Python

```
import pygame
```

```
run = True
```

```
width = 400
```

```
height = 100
```

```
pygame.init()
```

```
screen = pygame.display.set_mode((width, height))
```

```
font = pygame.font.SysFont(None, 48)
```

```
text = font.render("Welcome to pygame", True, (255, 255, 255))
```

```
screen.blit(text, ((width - text.get_width()) // 2, (height - text.get_height()) // 2))
```

```
pygame.display.flip()
```

```
while run:
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT\
```

```
        or event.type == pygame.MOUSEBUTTONDOWN\
```

```
        or event.type == pygame.KEYUP:
```

```
            run = False
```

Fuente: Elaboración propia



- La línea 1: importa pygame y nos permite usarlo.
- La línea 3: el programa se ejecutará mientras la variable run sea True.
- Las líneas 4 y 5: determinan el tamaño de la ventana.
- La línea 6: inicializa el entorno pygame.
- La línea 7: prepara la ventana de la aplicación y establece su tamaño.
- La línea 8: crea un objeto que represente la fuente predeterminada de 48 puntos.
- La línea 9: crea un objeto que represente un texto dado, el texto será suavizado (True) y blanco (255,255,255).
- La línea 10: inserta el texto en el búfer de pantalla(actualmente invisible).
- La línea 11: invierte los búferes de la pantalla para que el texto sea visible.
- La línea 12: el bucle principal de pygame comienza aquí.
- La línea 13: obtiene una lista de todos los eventos pendientes de pygame.
- Las líneas 14 a la 16: revisan si el usuario ha cerrado la ventana o ha hecho clic en algún lugar dentro de ella o ha pulsado alguna tecla.
- La línea 15: si es así, se deja de ejecutar el código.

Fuente: Elaboración propia