

Post Destacado

Configurando la Impresión Perfecta de un Libro de Excel

escrito por Federico Garay

Aunque vivimos en una era digital, a veces todavía es necesario imprimir datos, tablas, gráficos, o información de cualquier tipo que hayamos procesado en Excel. Especialmente si estamos hablando de informes financieros, gráficos estadísticos, ...

Encapsulamiento - Pilares de la Programación Orientada a Objetos en Python

escrito por Federico Garay



El **encapsulamiento** es el pilar de la programación orientada a objetos que se relaciona con **ocultar al exterior determinados estados internos de un objeto**, tal que sea **el mismo objeto quien acceda o los modifique**, pero que dicha acción **no se pueda llevar a cabo desde el exterior**, llamando a los métodos o atributos directamente.

Si bien en algunos lenguajes (como *Java*), puede resultar un procedimiento habitual, **Python** no lo implementa por defecto, pero nos propone una vía alternativa para lograrlo. Esto se hace **anteponiendo dos guiones bajos (__)** al nombrar un método o atributo. De esa manera, los mismos quedarán definidos como **“privados”**, y únicamente el mismo objeto podrá acceder a ellos.

Escuela Directa | Blog

```
# No accesible desde el exterior
def __metodo_oculto(self):
    print("Este método está oculto")
    self.__variable = 0
# Accesible desde el exterior
def metodo_normal(self):
    # El método si es accesible desde el interior
    self.__metodo_oculto()
alumno = Persona()
# alumno.__metodo_oculto() # Este método no es accesible desde el exterior
alumno.metodo_normal()    # Este método es accesible
```

Existe un pequeño truco (*no recomendado*) para acceder a los atributos y métodos ocultos. Dichos métodos están presentes con un nombre algo distinto:

instancia + _ + NombreClase + método/atributo oculto

```
alumno._Persona__metodo_oculto()
print(alumno._Persona__atributo_privado)
```



FEDERICO GARAY



Posts favoritos

Cohesión - Pilares de la Programación Orientada a Objetos en Python

La cohesión se refiere al grado de relación entre los elementos de un módulo . Cuando diseñamos una función, debemos identificar de un modo bien específico qué tarea va a realizar, reduciendo su finalidad a un objetivo único y bien definido. En resumen: para que una función sea cohesiva del ...

Acoplamiento - Pilares de la Programación Orientada a Objetos en Python

El acoplamiento es un concepto que mide la dependencia entre dos módulos distintos (como por ejemplo, clases). Podemos hablar de dos tipos: Acoplamiento débil , que implica que no hay dependencia entre un módulo y otros. Esta es la situación ideal. Acoplamiento fuerte , que es la situación ...

Con la tecnología de Blogger

Todos los derechos reservados. Prohibida su reproducción total o parcial