Notas de clase – Fundamentos de Programación

Autor: Kimi (Kimi-ai)
Fecha: 22 de septiembre de 2025

---

1. Compilador, transpilador e intérprete

表格 复制

Herramienta	Entrada	Salida	Mecanismo
Compilador	Código fuente (alto nivel)	Código máquina o byte-code **diferente**	Traduce **to**
Transpilador	Código fuente A	Código fuente B (mismo nivel de abstracción)	Traducción
Intérprete	Código fuente	Ejecución directa	Traduce **lín**

Flowgorithm **no compila** ni **interpreta**: **transpila** a varios lenguajes.

---

2. ¿Qué herramienta traduce `.fprg` a Python?

Es un **transpilador interno** integrado en Flowgorithm:

- Lee el XML `.fprg` (estructura de bloques).
- Recorre el árbol de funciones y genera **código Python legible** (mismo nivel de abstracción).
- **No produce byte-code**; el resultado es un archivo `*.py` que puedes editar.

Menú: *Program → Export → Python*.

---

3. Análisis comparativo línea por línea

A continuación se muestra:

- **Bloque Flowgorithm** (vista gráfica)
- **Código Python exportado**
- **Explicación transpiladora** paso a paso

Ejemplo: "Imprimir Hola y sumar dos números"

a) Archivo `demo.fprg` (fragmento XML)

xml 复制

```xml
<?xml version="1.0"?>
<flowgorithm fileversion="4.5">
  <algorithm name="demo">
    <attributes><attribute key="isMain" value="true"/></attributes>
    <function name="Main" type="None">
      <parameters/>
      <body>
        <output expression="&amp;quot;Hola&amp;quot;" newline="True"/>
        <declare name="a" type="Integer" array="False"/>
        <declare name="b" type="Integer" array="False"/>
        <declare name="s" type="Integer" array="False"/>
        <assign variable="a" expression="5"/>
        <assign variable="b" expression="7"/>
        <assign variable="s" expression="a + b"/>
        <output expression="&amp;quot;Suma = &amp;quot; &amp;amp; s" newline="True"/>
      </body>
    </function>
  </algorithm>
</flowgorithm>
```

b) Código Python generado (completo)

Python 复制

```python
#!/usr/bin/env python3
# Generated by Flowgorithm 4.5

def main():
    # Bloque Output
    print("Hola")

    # Bloque Declare
    a = 0
    b = 0
    s = 0

    # Bloque Assign
    a = 5
    b = 7
    s = a + b

    # Bloque Output
    print("Suma =", s)

if __name__ == "__main__":
    main()
```

c) Comparativa línea a línea

Paso en .fprg	Línea Python	¿Qué hace el transpilador?
<output expression=""Hola"" newline="True"/>	print("Hola")	Convierte la cadena y añade `print()` con `newline=True` → `print()` por defecto incluye `\n`
<declare name="a" type="Integer" .../>	a = 0	Crea variable con **valor inicial del tipo**: 0 para Integer, 0.0 para Real, "" para String, False para Boolean
<assign variable="a" expression="5"/>	a = 5	Traduce la expresión literal; mantiene operadores (+, -, *, /, &&, ``, etc.)
<output expression=""Suma = " & s" newline="True"/>	print("Suma =", s)	Convierte & (concatenación Flowgorithm) en **coma** de `print()` para evitar `str()` explícito

Reglas generales del transpilador

1. **Una función Main** → bloque `def main():` + `if __name__ == "__main__":`.
2. **Declaraciones** → inicialización con valor neutro.
3. **Asignaciones** → expresión tal cual; precedencia de operadores respetada.
4. **Salida** → función `print()`; concatenación con coma.
5. **Comentarios XML** → comentarios Python `# Bloque ...` para facilitar la lectura.

Conclusión para el alumno

- Flowgorithm **no compila** ni interpreta: **transpila** a código fuente Python.
- El archivo `.fprg` es XML; el exportado es **Python legible**.
- Comprender la traducción línea a línea ayuda a **depurar** y a **aprender Python** desde el diagrama de flujo.