

Características de los Lenguajes Funcionales

1. Provee un conjunto de funciones primitivas

Son funciones básicas ya definidas en el lenguaje (como $+$, $-$, $*$, $/$, $==$, $<$, $>$, `length`, `head`, `tail`). Permiten construir programas más complejos combinándolas.

2. Provee un conjunto de formas funcionales

Son esquemas para construir funciones nuevas a partir de otras. Ejemplos: `map` (aplica una función a cada elemento de una lista), `filter` (filtra elementos según una condición), `foldr` (reduce una lista a un valor usando una función).

3. Semántica basada en valores

Cada expresión representa un valor. No hay instrucciones secuenciales como en lenguajes imperativos. Por ejemplo, la función cuadrado $x = x * x$ devuelve un valor directamente.

4. Transparencia referencial

Una expresión puede reemplazarse por su valor sin afectar el programa. Esto significa que no hay efectos colaterales. Ejemplo: si $x = 2 + 2$, entonces $x * x$ siempre será 16.

5. Regla de mapeo basada en combinación o composición

Permite construir funciones combinando otras. Ejemplo: si $f(x) = x+1$ y $g(x) = x^2$, entonces $h(x) = f(g(x))$ es una composición. También puede escribirse como $h = f \circ g$.

6. Las funciones de primer orden

Las funciones se tratan como valores: pueden pasarse como argumento, devolverse como resultado, y almacenarse en variables. Ejemplo: aplicar $f \ x = f \ x$ aplica una función a un valor.