

Expresiones: son usadas sólo para **denotar valores**. Los valores son un subconjunto de las expresiones. Es importante distinguir entre las expresiones y el valor que denotan (ej: la expresión $6*7$ denota el número abstracto cuarenta y dos, al igual que "42").

Definiciones: permiten introducir nuevos valores y definir operaciones. Consisten en la **asociación de una expresión a un nombre**.

Patrones: expresiones formadas por variables, constructores y constantes. Para definir una función sólo puedo usar patrones para analizar los parámetros.

Pattern-matching: análisis por casos donde las guardas se usan para discernir sobre la forma del argumento y hacer referencia a sus componentes.

Constructores: **símbolos elementales** con los que construimos expresiones de cierto tipo. Los constructores del tipo A definen la forma de los valores de las expresiones de tipo A.

Tipos: toda expresión tiene un tipo asociado. Estos preceden a las expresiones, y a cada expresión correcta se le puede asignar un tipo, ya sea básico o compuesto obtenido a partir de tipos básicos. Cuando el tipo de una expresión incluye variables se lo llama polimórfico, y la variable se denomina variable de tipo.

- *bool*: true y false son sus constructores.
- *num* (int, double, float): infinitos valores, no accedemos a constructores,
- *char*: todo carácter es su constructor
- *[a]*: [] y : son sus constructores (*definiciones de f sobre listas de p. 123 en adelante*)

Predicado: función que retorna un valor de verdad. Algo que o vale, o no vale: booleano.

Un programa funcional es un conjunto de definiciones. Su ejecución consiste en evaluar una expresión y luego reducirla a su forma normal.