

Introducción a Standard ML

5to Año - IPS



AÑO 2016

- Lenguaje funcional, es decir, basado en la definición y aplicación de funciones.
- Libre de efectos colaterales. El cálculo procede a través de la evaluación de expresiones y no de la asignación de variables.
- Soporta funciones de orden superior.
- Soporta el polimorfismo.
- Recomienda la recursión por sobre las iteraciones con bucles.
- Es un lenguaje fuertemente tipado. Todo valor y variable tiene un tipo que puede ser decidido en tiempo de compilación.



Interactuando con ML

La mayoría de las implementaciones son interactivas:

“read-eval-print”

ML analiza, compila y ejecuta las expresiones y el resultado es mostrado en la terminal

```
- 3+2;  
val it = 5 : int
```



Pueden presentarse distintos tipos de **errores** cuando interactuamos con ML.

La mayoría corresponde a alguna de las siguientes **tres categorías**:

- **Error de sintaxis**
- **Error de tipeo**
- **Error en tiempo de ejecución**



Pueden presentarse distintos tipos de **errores** cuando interactuamos con ML.

La mayoría corresponde a alguna de las siguientes **tres categorías**:

- **Error de sintaxis**

```
- let x=3 in x end;  
stdIn:9.1-9.7 Error: syntax error: deleting LET ID EQUAL OP  
stdIn:9.9 Error: syntax error found at IN
```

- **Error de tipeo**

- **Error en tiempo de ejecución**



Pueden presentarse distintos tipos de **errores** cuando interactuamos con ML.

La mayoría corresponde a alguna de las siguientes **tres categorías**:

- **Error de sintaxis**

- **Error de tipeo**

```
- 3+true;  
stdIn:1.2-10.2 Error: operator and operand don't agree [literal]  
operator domain: int * int  
operand:         int * bool  
in expression:  
  3 + true
```

- **Error en tiempo de ejecución**



Pueden presentarse distintos tipos de **errores** cuando interactuamos con ML.

La mayoría corresponde a alguna de las siguientes **tres categorías**:

- **Error de sintaxis**
- **Error de tipo**
- **Error en tiempo de ejecución**

```
- 3 div 0;
```

```
uncaught exception Div [divide by zero]  
  raised at: stdIn:14.3-14.6
```



Expresiones básicas, valores y tipos

En ML un **TIPO** es una colección de valores.

Ejemplos: los enteros forman un tipo, las cadenas de caracteres otro tipo de datos y los booleanos otro.

Dados dos tipos de datos ω y τ , el conjunto de pares ordenados cuya componente izquierda es de tipo ω y su componente derecha pertenece a τ es un tipo de datos.

El conjunto de funciones que mapean un tipo de dato a otro, también forman un tipo.

ML permite al usuario definir sus propios tipos de datos.



Tipos y valores básicos

Nombre del tipo	Descripción	Valores
unit	Denota vacío	()
bool	Valores booleanos	true, false
int	Números enteros	3, 78, ~4
char	Caracteres	#"A", #"\n"
string	Cadena de caracteres	"HOLA", "", "HO LA"
real	Números reales	3.0, ~2.3, 1e~3

