ML

- ML is a functional programming language
- the ML environment runs in an interactive mode

  ☐$ sml
  Standard ML of New Jersey v110.78 [built: Tue Sep 8 14:59:55 2015]-

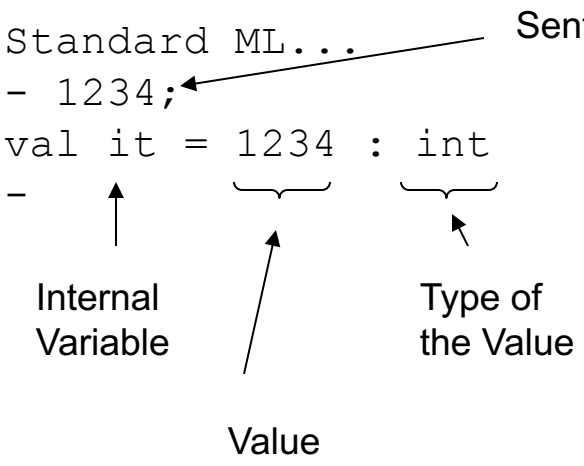
  - ← ML System Prompt

- At the prompt the system expects a valid sentence in ML

Read Chap 5

ML – Constant Exressions

The simplest sentence in the ML language is a constant expression



Other Constants:

real	123.4
bool	true/false
string	“Susan”
char	#”Q”

ML – Operators and Simple Expressions

Example:

```
- ~ 1 + 2 - 3;
val it = ~2 : int
```

~ is the unary -, here -2.

string concatenation; “abc” ^ “def”

Precedence	Operators
High	`not` `~`
	`*` `/` `div` `mod`
	`+` `-` `^`
	`<` `>` `<=` `>=` `=` `<>`
	`andalso` (logical and)
Low	`orelse` (logical or)

- (4 > 0) andalso (4 mod 2 = 0);
val it = true : bool

ML – Conditional Expressions

if – then – else

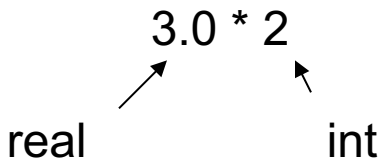
or

if – then

```
-if 1 < 2 then #”x” else #”y”;
val it = #”x” : char
```

ML – Type Conversions

Most programming languages we are used to allow for mixed-type expressions such as



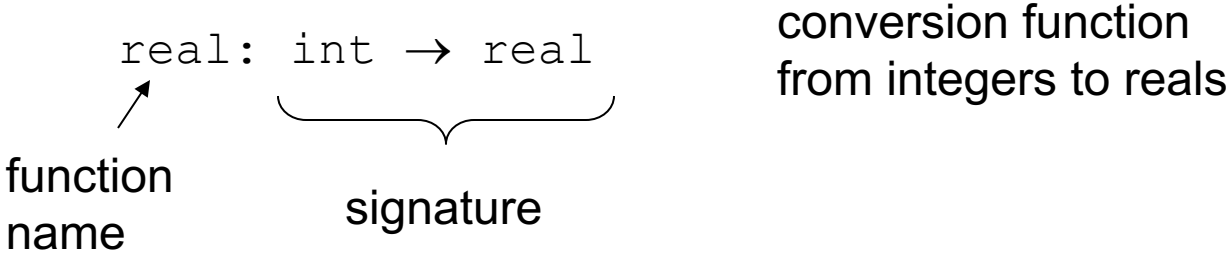
ML does not allow mixed-type expressions.

```
- 3.0 * 2;
Error: operator and operand don't agree
  operator domain: real * real
  operand:         real * int
  in expression:
    3.0 * 2
-
```

ML – Type Conversions

However, we can use type conversions to manipulate the types of an expression.

Example:



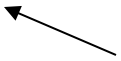
Other conversion functions:

```
floor: real → int   (round down)
ceil:  real → int   (round up)
round: real → int   (round to nearest int)
```

ML – Type Conversions

We can now rewrite our illegal expression from before:

```
- 3.0 * real(2);
```



Convert 2 into 2.0

```
val it = 6.0 : real
```

or

```
- floor(3.0) * 2;
val it = 6 : int
```

ML – Variable Definitions

Very simple syntax:

```
- val x = 1 + 2 * floor(3.0);
val x = 7 : int
```

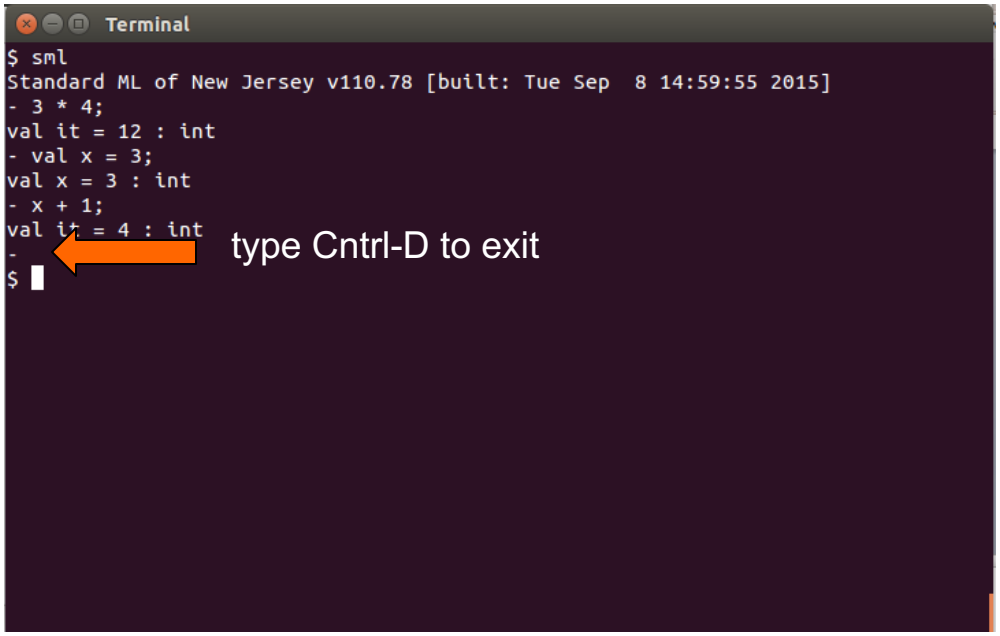
No longer the
default variable

In ML you do not need
to declare the type of a variable;
ML will determine its type through
type inference.

Of course we can use the values of variables:

```
- x + 1;
val it = 8 : int
```

Try it yourself



- Log into UbuntuBox
- account: csc301 password: csc301isfun
- start a terminal
- run sml (see above)