

```
str1 ++ str2
```

Lists of characters.

\neq and unary not

```
if bool then expr1 else expr2
```

The `else` clause is required, to ensure an interesting result for the whole `if` expression.

... strict.

`:t expr` shows a value's type in `ghci`.

`:set +t` makes it permanent for that session.

The optional type declaration and the function declaration.

```
module MyMath where
```

```
  my_max :: Integer -> Integer -> Integer
```

```
  my_max x y = if x > y then x else y
```


- Multiple function definitions.
- Guards.

Delay the equals sign, splitting up the argument lists.

```
func args  
    | boolean1 = ...  
    | boolean2 = ...
```

`otherwise` is an alias for `true`, often used as the last guard.

Tuples use round parens, lists use brackets.