



Haskell

- Haskell was an attempt to create "the" standard functional language to achieve a consistent base for research.
- *Haskell* is a functional programming language.
- The implementation of Haskell we will be using is *Hugs* (the name comes from Haskell User's Gofer System, where 'Gofer' is Good for equational reasoning.)
- Be clear about the distinction: Haskell is the *programming language* and Hugs is the *system* that runs Haskell programs.

Sessions

- When you start up Hugs, you will get a window called a *session*. It will say something like this:

```
Type :?  for help
Prelude>
```

- The `>` is a *prompt*. You can type in an *expression* at the prompt. Hugs will print its value and then prompt for another expression. For example:

```
Prelude> 3+4-2
5
Prelude>
```

- The language in which you write the expression is Haskell.



Scripts

- Hugs knows about lots of functions and operators that can be included in expressions. They are defined in the *Prelude* which is just a Haskell program (`Prelude.hs`).
- A Hugs session is like a powerful calculator, but using it is not really *programming*.
- Programming in Haskell centres around defining your own functions and this is done in a *script*, which is a file containing *definitions*, *declarations* and *comments*.
- By *loading* a script into Hugs, you can use the session to evaluate expressions containing functions and operators defined in the script, as well as those in the Prelude.



Script Contents

- Comments

```
-- square the value given by x
```

is a comment. Comments start with `--` and the remainder of the line is ignored.

- Declarations

```
square :: Int -> Int
```

is a type declaration. It tells Hugs (and human readers) the type of `square`.

- Definitions

```
square x = x * x
```

is a definition. It defines the function `square`.