An interactive evaluator for a language.

One bit less than the size of a machine word.

- Using the `float` operators, `+.`, `*.`, etc.
- Using the `float_of_int` conversion.

```
Char.code

Char.chr
```

```
str1 ^ str2
```

... mutable.

```
# let seuss = "cat";;
val seuss : string = "cat"
# seuss.[0] <- 'h';;
- : unit = ()
# seuss;;
- : string = "hat"
```

Type `bool` has values `true` and `false`.

Negation is performed with unary `not`.

Equality:

```
x = y
x <> y
```

Identity:

```
x == y
x != y
```

&&, ||

```
if cond then expr1 else expr2
```

```
null
```

The same type.

- `ocaml` is the toploop
- `ocamlc` is the bytecode compiler
- `ocamlopt` is the machine code compiler
- `ocamldebug` is the debugger