

How can you compile and load a module into
the `erl` shell?

```
c(module.erl) .
```

Now that module's functions can be accessed with:

```
module:func
```

What do Erlang programs begin with?

```
-module(filename) .
```

... for a program defined in filename.erl.

How can you use methods in other modules
without module qualification?

```
-import(filename, [func1/arity1, ..., funcN/arityN]).
```

How can you make your module's functions available to other modules?

`-export(filename, [func1/arity1, ..., funcN/arityN]).`

or

`-compile(export_all).`

What are the Erlang case rules?

Similar to Prolog, and Haskell, functions and atoms begin with a lower case, variables begin with a capital letter or an underscore.

What is an atom?

What are its lexical rules?

An atom is a word that stands for itself.

It begins with a lowercase letter or is enclosed in single quotes.

What are Erlang's primitive data types?

Integers, floats, strings, atoms, lists, tuples, binaries.

How are lists, tuples, and binaries written?

Lists are enclosed in square brackets, tuples in curly brackets, and binaries in double angle brackets.