

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

What do Erlang programs begin with?

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

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

What are the Erlang case rules?

What is an atom?

What are its lexical rules?

What are Erlang's primitive data types?

How are lists, tuples, and binaries written?

What's odd about Erlang's Boolean operators?

How are "greater or equal to" and "less than or equal to" expressed?

What are the equality operators?

How are they used?

What is = in Erlang?

What is the syntax of `case` expressions?

What is an expression sequence?

What is its value?

Give the syntax of `if` expressions.

What happens if no guard in an `if` expression is true?

How can this be avoided.

What is the huge restriction on guards?

How do you access the length of a list?

Of a tuple?

Give the syntax of named functions.

Give the syntax of anonymous functions.

What are they often used for?

Give the syntax of list comprehensions.

Give a generator that doubles the members of
a list.

Which list functions do not need to be imported or qualified?

Create a range of integers.

Get a line from stdin.

Put a line to stdout.

How do you write to a file?

What are the reserved sequences in format strings?

What is the comment syntax?

`filter` and `map` take which arguments?

How can you get truncating division?

How can you use previously defined functions as parameters to other functions?

What are strings, really?

What do `lists:all` and `lists:any` do?

How is Erlang's `zip` more restrictive than other languages'?

Append two lists.

What do `takewhile` and `dropwhile` do?

What does `partition` do?

How can you run code in a new process?

How can you generate a "return address" to send with your message?

Give the syntax of `receive`.

How is Erlang's `loop()` different from
Scala's?

What's an idiomatic way to set up an Erlang remote procedure call server?

Describe Pid registration.

What's the difference between `spawn` and
`spawn_link`?

How can you introduce/remove links between two running processes?

How can you terminate the current process?

How can you send exit signals *without
actually exiting?*

How can you link to a process while reserving the right to handle the partner process's exit however you like?

Why isn't `try/catch` much used in Erlang?

What is a common strategy for making functions tail recursive?