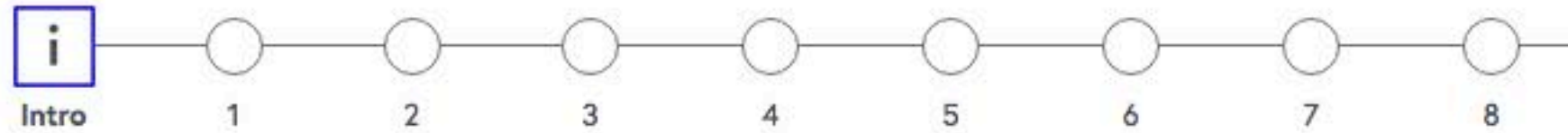


# Programs in Erlang - recap



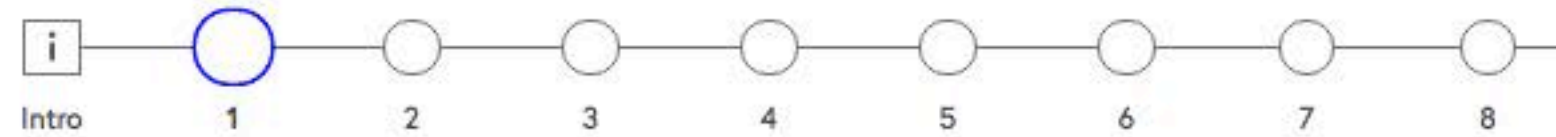
We'll finish this activity with a quiz. This will help you to check your understanding of the basics of writing a complete Erlang program, the principal differences between data in Erlang and in other languages, and the use of variables and patterns.

## QUIZ RULES

- Quizzes do not count towards your course score, they are just to help you learn
- You may take as many attempts as you wish to answer each question
- You can skip questions and come back to them later if you wish

[Begin quiz](#)[SUMMING UP  
VIDEO](#)[SKIP QUIZ  
GO TO STEP 1.18](#)

# Programs in Erlang - recap



## Question 1

Let's start with a fairly simple expression.

Assume that you have compiled the `first.erl` program in Erlang. What is the result of typing

```
2 + double(3)
```

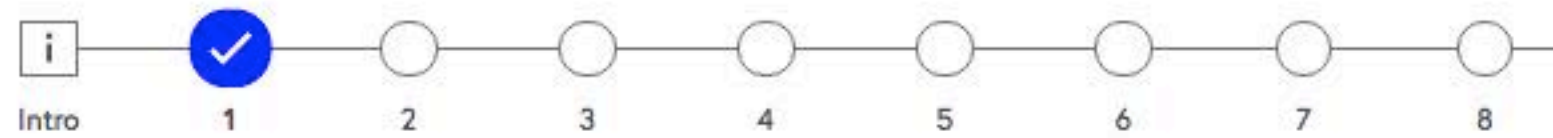
followed by a newline, to the prompt?

☐ an error message

☐ 8

☐ 6

☐ no answer



## Question 1

Let's start with a fairly simple expression.

Assume that you have compiled the `first.erl` program in Erlang. What is the result of typing

```
2 + double(3)
```

followed by a newline, to the prompt?

- ☐ an error message
- ☐ 8
- ☐ 6

☒ no answer

## Correct



Simon Thompson

LEAD EDUCATOR

## Question 2


OK, let's remember that full stop! Is there anything else wrong with the expression?


Assume that you have compiled the `first.erl` program in Erlang. What is the result of typing

```
2 + double(3).
```

followed by a newline, to the prompt?

- ☐ an error message
- ☐ 8
- ☐ 6
- ☐ no answer

 PREVIOUS QUESTION

SKIP QUESTION 



## Question 2

OK, let's remember that full stop! Is there anything else wrong with the expression?

Assume that you have compiled the `first.erl` program in Erlang. What is the result of typing

```
2 + double(3).
```

followed by a newline, to the prompt?

☒ an error message

☐ 8

☐ 6

☐ no answer

## Correct



Simon Thompson

LEAD EDUCATOR

That's right. You'll get an error message, because the function call should be fully qualified - that is, of the form

`first:double(3).`

## Question 3

What is wrong with the following definition of an Erlang function?

```
square(x) -> x*x
```

- ☐ square is a built-in function
- ☐ Variables should begin with a capital letter
- ☐ The function should begin with a capital letter
- ☐ The definition should end with a semi-colon

 PREVIOUS QUESTION

SKIP QUESTION 

### Categories

Courses grouped by subjects

### Courses

Browse all individual online courses

### Programs

Master a specific subject in depth

## Question 3

What is wrong with the following definition of an Erlang function?

```
square(x) -> x*x
```

- ☐ square is a built-in function
- ☒ Variables should begin with a capital letter
- ☐ The function should begin with a capital letter
- ☐ The definition should end with a semi-colon

## Correct



Simon Thompson LEAD EDUCATOR

That's right, the variable should be `X` not `x`. Note that semi-colons are used to separate different function clauses. We'll learn more about that in due course.



## Question 4

Suppose that when you compile an Erlang module you receive a warning that a function definition is not used. What could be the reason for this?

- ☐ The function is not called by any of the functions that are exported from the module.
- ☐ The function name is mis-spelled in the export list.
- ☐ The function is not called in any of the other definitions in the module.
- ☐ The function is not exported correctly, as a name/arity entry in the export list.

[PREVIOUS QUESTION](#)[SKIP QUESTION](#)



## Question 4

Suppose that when you compile an Erlang module you receive a warning that a function definition is not used. What could be the reason for this?

- ☐ The function is not called by any of the functions that are exported from the module.
- ☐ The function name is mis-spelled in the export list.
- ☐ The function is not called in any of the other definitions in the module.
- ☒ The function is not exported correctly, as a name/arity entry in the export list.

## Correct



Simon Thompson

LEAD EDUCATOR

That's right. The function is not exported correctly. Note that if the function name had been mis-spelled in the export list, you would get an error message saying that the mis-spelled function wasn't defined in the module.

## Question 5

Which of the following expressions will treble the value of `N`, when `N` is an integer, and you want the answer to be an integer too?

- ☐ `double(N) + N.`
- ☐ `first:mult(3.0,N).`
- ☐ `N + first:double(N).`
- ☐ `N = double(N) + N.`

[< PREVIOUS QUESTION](#)[SKIP QUESTION >](#)

### Categories

Courses grouped by subjects

### Courses

Browse all individual online courses

### Programs

Master a specific subject in depth

[Using FutureLearn](#)[Why it works](#)[About FutureLearn](#)[Our team](#)[Learning for business](#)[For healthcare](#)[Partners](#)[Become a partner](#)[FAQ](#)[Blog](#)[Support](#)

## Question 5

Which of the following expressions will treble the value of `N`, when `N` is an integer, and you want the answer to be an integer too?

- ☐ `double(N) + N.`
- ☐ `first:mult(3.0,N).`
- ☒ `N + first:double(N).`
- ☐ `N = double(N) + N.`

## Correct



Simon Thompson LEAD EDUCATOR

That's right. Note that `first:mult(3.0,N).` would first change integers to floats, and `N = double(N) + N.` will fail as a re-assignment. More about this later.



## Question 6

Which of these is *not* an Erlang atom

- ☐ erlang-atom
- ☐ erlang\_atom
- ☐ 'i am not an erlang atom'
- ☐ 'Erlang atom'

 PREVIOUS QUESTION

SKIP QUESTION 

### Categories

Courses grouped by subjects

### Courses

Browse all individual online courses

### Programs

Master a specific subject in depth

#### Using FutureLearn

Why it works  
Proving your learning

#### About FutureLearn

Our team  
Our principles

#### Learning for business

For healthcare

#### Partners

Become a partner

#### FAQ

#### Blog



## Question 6

Which of these is *not* an Erlang atom

- ☒ `erlang-atom`
- ☐ `erlang_atom`
- ☐ `'i am not an erlang atom'`
- ☐ `'Erlang atom'`

## Correct



Simon Thompson

LEAD EDUCATOR

That's right. The hyphen is interpreted as a minus sign.



PREVIOUS QUESTION

NEXT QUESTION



## Question 7

Which of these is *not* an Erlang string?

- ☐ `[$w,$e,$l,$l]`
- ☐ `"ac"`
- ☐ `'Erlang string'`
- ☐ `[97,99]`

[< PREVIOUS QUESTION](#)

[SKIP QUESTION >](#)

### Categories

Courses grouped by subjects

### Courses

Browse all individual online courses

### Programs

Master a specific subject in depth

#### Using FutureLearn

Why it works  
Proving your learning

#### About FutureLearn

Our team  
Our principles

#### Learning for business

For healthcare

#### Partners

Become a partner

#### FAQ

#### Blog

## Question 7

Which of these is *not* an Erlang string?

☐ `[$w,$e,$l,$l]`

☐ `"ac"`

☒ `'Erlang string'`

☐ `[97,99]`

## Correct



Simon Thompson

LEAD EDUCATOR

That's right. `'Erlang string'` is an *atom* - and that's *not* a string.



PREVIOUS QUESTION

NEXT QUESTION



## Question 8

Which is the correct statement about the difference between lists and tuples?

- ☐ Tuples must always have an atom as a first element; lists don't need to.
- ☐ There is an empty list, but not an empty tuple.
- ☐ Lists and tuples differ in the way that they can be analysed and pattern matched.
- ☐ The contents of a list are ordered, but those of a tuple are not.

 PREVIOUS QUESTION

FUNCTIONS IN ERLANG  
ACTIVITY INTRO 

### Categories

Courses grouped by subjects

### Courses

Browse all individual online courses

### Programs

Master a specific subject in depth



## Question 8

Which is the correct statement about the difference between lists and tuples?

- ☐ Tuples must always have an atom as a first element; lists don't need to.
- ☐ There is an empty list, but not an empty tuple.
- ☒ Lists and tuples differ in the way that they can be analysed and pattern matched.
- ☐ The contents of a list are ordered, but those of a tuple are not.

## Correct



Simon Thompson

LEAD EDUCATOR

That's right. For example, it's possible to write a pattern that will match *any* non-empty list, `[ X | Xs ]`, whereas it is not possible to write a pattern that will match any non-empty pattern: it is only possible to write patterns that match tuples of a given length, such as `{ A, B, C }`, which matches a tuple of size three.