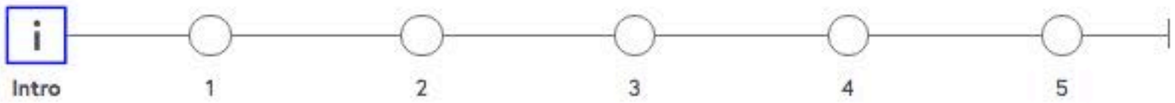
Messages, mailboxes and timing



This quiz aims to consolidate your understanding of timing and message interleaving in the context of multiple Erlang processes.

QUIZ RULES

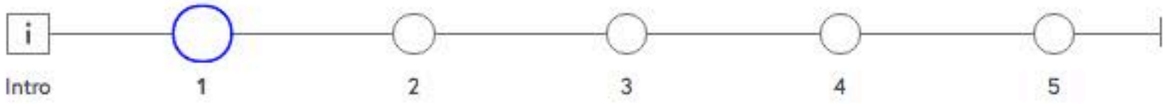
- Quizzes do not count towards your course score, they are just to help you learn
- You may take as many attempts as you wish to answer each question
- You can skip questions and come back to them later if you wish

Begin quiz

TIMING AND MESSAGE ORDERING
VIDEO

SKIP QUIZ
GO TO STEP 1.11

Messages, mailboxes and timing



Question 1

For the following expression, which of the following statements is true?

```
receive
  a -> a;
  b -> b
end
```

- ◯ It will receive message a and then message b.
- ◯ If only the message b is in the mailbox, then execution will wait for a message a to be delivered.
- ◯ It will receive a or b and then the 'other' message, i.e. b or a.
- ◯ It will receive one of the messages a and b.

Question 2

Suppose that the following function is spawned as a process. Which of the following statements is true?

```
loop() ->
  receive
    a -> loop();
    b -> loop()
  end.
```

- ◯ This process will consume all as and bs in the mailbox, in the order that they occur.
- ◯ This process will first process all the as in the mailbox, and then all the bs in the mailbox.
- ◯ The process will fail if there are any messages in the mailbox other than as and bs.
- ◯ The process will terminate when all as and bs are removed from the mailbox.

PREVIOUS QUESTION

SKIP QUESTION

Question 3

Three processes `A`, `B` and `C` are running concurrently. `A` and `B` both send messages to process `C`. If `A` sends the message `A1` then `A2`, and `B` sends the message `B1`, how many different orders can the messages be delivered into `C`'s mailbox?

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ 6

PREVIOUS QUESTION

SKIP QUESTION

Question 4

Suppose that the `loop` function below is spawned with argument `0`, and that after receiving some messages it prints out a number: what will that number represent?

```
loop(N) ->
  receive
    stop -> io:format("~w~n",[N]);
    _Msg -> loop(N)
  after 1000 ->
    loop(N+1)
  end.
```

- ◯ It will print the number of times that there is at least a second between messages.
- ◯ It will return the number of milliseconds the process runs until the first stop.
- ◯ It will return the maximum number of seconds between messages.
- ◯ It will return the total number of complete seconds in which no message is received.

PREVIOUS QUESTION

SKIP QUESTION

Support

Question 5

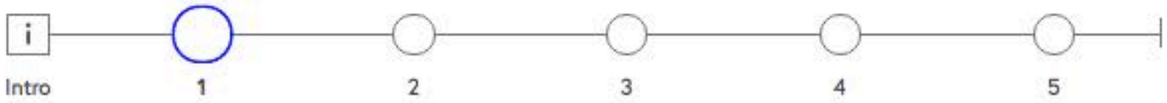
Executing `loop()` emits a `tick` every second. Suppose a reset is received 400 milliseconds after the last `tick`: how long is it from this point until the next `tick`?

```
loop() ->
  receive
    reset -> loop()
  after 1000 ->
    io:format("tick~n"),
    loop()
  end.
```

- ◯ 0 milliseconds
- ◯ 400 milliseconds
- ◯ 600 milliseconds
- ◯ 1 second

PREVIOUS QUESTION

ESTABLISHING COMMUNICATION: PIDS AND NAMED PROCESSES VIDEO



Question 1

For the following expression, which of the following statements is true?

```
receive
  a -> a;
  b -> b
end
```

- ( ) It will receive message a and then message b.
- ( ) If only the message b is in the mailbox, then execution will wait for a message a to be delivered.
- (•) It will receive a or b and then the 'other' message, i.e. b or a.
- ( ) It will receive one of the messages a and b.

Submit answer

Support

Question 1

For the following expression, which of the following statements is true?

```
receive
  a -> a;
  b -> b
end
```

- ◯ It will receive message a and then message b.
- ◯ If only the message b is in the mailbox, then execution will wait for a message a to be delivered.
- ◯ It will receive a or b and then the 'other' message, i.e. b or a.
- ◉ It will receive one of the messages a and b.

Submit answer

INTRO

SKIP QUESTION

For the following expression, which of the following statements is true?

```
receive
  a -> a;
  b -> b
end
```

- ◯ It will receive message a and then message b.
- ◯ If only the message b is in the mailbox, then execution will wait for a message a to be delivered.
- ◯ It will receive a or b and then the 'other' message, i.e. b or a.
- ✓ **It will receive one of the messages a and b.**

Correct

Simon Thompson LEAD EDUCATOR

Yes, that's right. It will receive the first message in the mailbox that matches either a or b.

Suppose that the following function is spawned as a process. Which of the following statements is true?

```
loop() ->
  receive
    a -> loop();
    b -> loop()
  end.
```

- [x] **This process will consume all as and bs in the mailbox, in the order that they occur.**
- [ ] This process will first process all the as in the mailbox, and then all the bs in the mailbox.
- [ ] The process will fail if there are any messages in the mailbox other than as and bs.
- [ ] The process will terminate when all as and bs are removed from the mailbox.

Correct

Simon Thompson LEAD EDUCATOR

Yes! That's right.

Question 3

Three processes `A`, `B` and `C` are running concurrently. `A` and `B` both send messages to process `C`. If `A` sends the message `A1` then `A2`, and `B` sends the message `B1`, how many different orders can the messages be delivered into `C`'s mailbox?

- ○ 1
- ○ 2
- ● 3
- ○ 6

Correct

Simon Thompson LEAD EDUCATOR

That's right. `A1` **must** come before `A2`, but `B1` can come before `A1`, between `A1` and `A2`, or after `A2`.

Question 4

Suppose that the `loop` function below is spawned with argument `0`, and that after receiving some messages it prints out a number: what will that number represent?

```
loop(N) ->
  receive
    stop -> io:format("~w~n",[N]);
    _Msg -> loop(N)
  after 1000 ->
    loop(N+1)
  end.
```

- ○ It will print the number of times that there is at least a second between messages.
- ○ It will return the number of milliseconds the process runs until the first stop.
- ○ It will return the maximum number of seconds between messages.
- ● **It will return the total number of complete seconds in which no message is received.**

Correct

Simon Thompson LEAD EDUCATOR

That's right. The variable is only incremented when there is a one second period in which no message is received. It will reset and start counting to one second again.

Question 5

Executing `loop()` emits a `tick` every second. Suppose a reset is received 400 milliseconds after the last `tick`: how long is it from this point until the next `tick`?

```
loop() ->
  receive
    reset -> loop()
  after 1000 ->
    io:format("tick-n"),
    loop()
  end.
```

- ○ 0 milliseconds
- ○ 400 milliseconds
- ○ 600 milliseconds
- ● **1 second**

Correct

Simon Thompson LEAD EDUCATOR

That's right. Once `loop/0` is called after processing the reset message then it is 1000 milliseconds until the tick occurs next.

PREVIOUS QUESTION

ESTABLISHING COMMUNICATION: PIDS AND NAMED PROCESSES