```
receive
    {one, 1} ->
        ...
    X ->
        ...
end

receive
    {one, 1} ->
        ...
    timeout (0) -> receive
                       X ->
                   end
end.
```

March 30, 2011 - Synchronous Message Passing

- In Erlang we have what's called Asynchronous Message Passing rule.

vs.

- Synchronous M.P.
  To send a message, the sender must be at a send operation and the receiver at a receive operation <u>at the same time</u>: synchonized.
  - Synchronous Queue in Java.

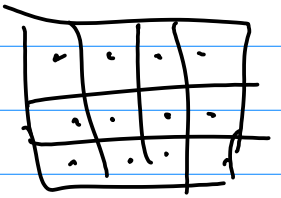
History:

Anthony Hoare : Communicating Sequential Processes
~ 1976

→ invented the monitor concept that's the historical root of Java's intrinsic synchronization.

→ Ada — Invented at request of U.S. DoD to be the be-all & end-all language for business applications thru avionics and embedded systems. about 1978-1980
Competitor PL/I — IBM's attempt to merge COBOL & FORTRAN

→ OCCAM — Transputer



Early 1980's Robin Milner — Calculus of Communicating Systems (CCS)
Pi-calculus
ML Programming Language
π-calculus

Synch MP design that we'll talk about comes from a language called Concurrent ML

↳ John Reppy

- channels —

  send
  recv.

  the values sent on a channel must all have the same type

  int channel
  string channel

channels are first-class values.

Send operation takes a channel and value and returns void / unit

recv operations takes a channel as argument and returns a value received on that channel

send: $(\alpha \text{ channel} * \alpha) \rightarrow \text{unit}$

recv: $(\alpha \text{ channel}) \rightarrow \alpha$

P ! M

Execution rules for send and receive:

send (c, v):

wait until some other process is executing recv(c), Then both the send and recv return, and the value returned by send is void and the valued returned by recv. is v.

Problems:

- if you are receiving messages on different channels you have to know the order in which they will be sent.

  - dead lock possibility.

Concurrent ML also has a select function operation that implements allowing a process to wait for multiple channels at the same time

(Other languages using Synch MP may or may not implement this —
Ada - rendezvous
case-like expression
select.