Message passing variability:

- Erlang - asynchronous message passing
  - per-process mail boxes

asynchronous means senders don't wait when sending.

for receiving:
receive is synchronous, even in Erlang.

- Many other combinations are possible!
  - Synchronous sends.
  - Channels instead of mailboxes:
    Channel is itself a value in the language.
    A process may receive or send to different channels at different times.

A bit of history:

Anthony Hoare : ~1976 - Communicating Sequential Processes. — CSP
— channels & synchronous MP.

Ada - late 1970s : Rendezvous —
mult.-way synchronous
send and receive mechanism.

* OCCAM - based on CSP - Transputer
Multi-processor — mid '80s

Theory: { Calculus of Communicating Systems (CCS) — Robin Milner
Pi Calculus
Process Calculus

Now we'll look at a real language using synchronous message passing and channels.

Language Concurrent ML
- based on ML language (Milner)
- extensions by John Reppy.

ML: statically typed, mostly-functional language
(contrast Erlang: dynamically typed, mostly functional)

Concurrent ML:
typed channels: each channel can only carry values of one particular type.

type langage

int channel		int list
string channel	$\approx$	string list
bool channel		bool list

And then we have operations to send and receive on channels

args

result
no result

Send (c, v) : α channel * α -> unit

recv (c) : α channel -> α

Synchronous