

Processes

- All code runs in Processes
- Isolated and communicate through messages
- Provide means for fault-tolerant and distributed programs
- Different from system processes
- Extremely light weight.

Pid and self/0

- pid is the process unique identifier
- self/0 returns the current process id

```
iex(1)> self
#PID<0.82.0>
```

Process module functions

spawn/1

- Take a function as argument and returns a pid `spawn((() -> any)) :: pid`
- executes function and exits
- Typically not used in favor of higher level abstractions such as `task`, `GenServer` and `Agent` (don't worry we will learn about then)

```
iex>pid = spawn fn -> IO.puts "oi" end
oi
#PID<0.86.0>
```

spawn/3

- Takes a module, a function (as an atom) and a list of arguments for the function.

```
spawn(module, atom, list) :: pid
spawn(SomeModule, :function, [1, 2, 3])
```

Links

- functions: `spawn_link/1` & `spawn_link/3`
- works like `spawn` but links created processes to current
- Normal processes are isolated so errors don't propagate.

```
iex> spawn fn -> raise "oops" end
#PID<0.58.0>
```

```
[error] Process #PID<0.58.00> raised an exception
** (RuntimeError) oops
    (stdlib) erl_eval.erl:668: :erl_eval.do_apply/6

    Only logged error
```

- Linked processes propagate errors to it's parents.

```

iex> self()
#PID<0.41.0>
iex> spawn_link fn -> raise "oops" end

** (EXIT from #PID<0.41.0>) evaluator process exited with reason: an exception was raised:
    ** (RuntimeError) oops
        (stdlib) erl_eval.erl:668: :erl_eval.do_apply/6

[error] Process #PID<0.289.0> raised an exception
** (RuntimeError) oops
    (stdlib) erl_eval.erl:668: :erl_eval.do_apply/6
    Parent exited

```

send & receive

- send messages with send/2 to another process mailbox
- receive messages with receive/1
- goes through the mailbox for a message that pattern matches
- if there are no message the process waits until a messages arrives

```

iex> send self(), {:hello, "world"}
{:hello, "world"}
iex> receive do
...>   {:hello, msg} -> msg
...>   {:world, msg} -> "won't match"
...> end
"world"

```