

Olá mundo!

O Olá mundo em Gleam

```
import gleam/io

pub fn main() {
  io.println("Hello, world!")
}
```

Em Gleam, declaramos uma função nomeada usando a keyword `fn`, seguida pelo nome da função, seus parâmetros e por fim, seu corpo delimitado por chaves.

Já funções anônimas são criadas com a sintaxe `fn(argumentos) {corpo da função}`



```

// Funções nomeadas – são criadas utilizando a keyword fn seguido do nome da função
pub fn hello_world() -> String {
  "Hello, from Gleam!"
}

pub fn add(x: Int, y: Int) -> Int {
  x + y
}

pub fn multiply(x: Int, y: Int) -> Int {
  x * y
}

// Funções anônimas – são criadas com a sintaxe fn(argumentos) {corpo da função}
pub fn run() {
  let add = fn(x, y) { x + y }

  add(1, 2)
}

// Funções podem receber como argumento outra função e podem ser encadeadas
pub fn twice(f: fn(t) -> t, x: t) -> t {
  f(f(x))
}

pub fn add_one(x: Int) -> Int {
  x + 1
}

pub fn add_two(x: Int) -> Int {
  twice(add_one, x)
}

```

```

import gleam_lab
import gleam/should

pub fn hello_world_test() {
  gleam_lab.hello_world()
  |> should.equal("Hello, from Gleam!")
}

pub fn add_test() {
  gleam_lab.add(2, 4)
  |> should.equal(6)
}

pub fn multiply_test() {
  gleam_lab.multiply(2, 4)
  |> should.equal(8)
}

pub fn add_two_test() {
  gleam_lab.add_two(1)
  |> should.equal(3)

  gleam_lab.add_two(2)
  |> should.equal(4)
}

```