



# Elixir: Scalable and Efficient Application Development

---

*João Gonçalves*

*Standard IO*

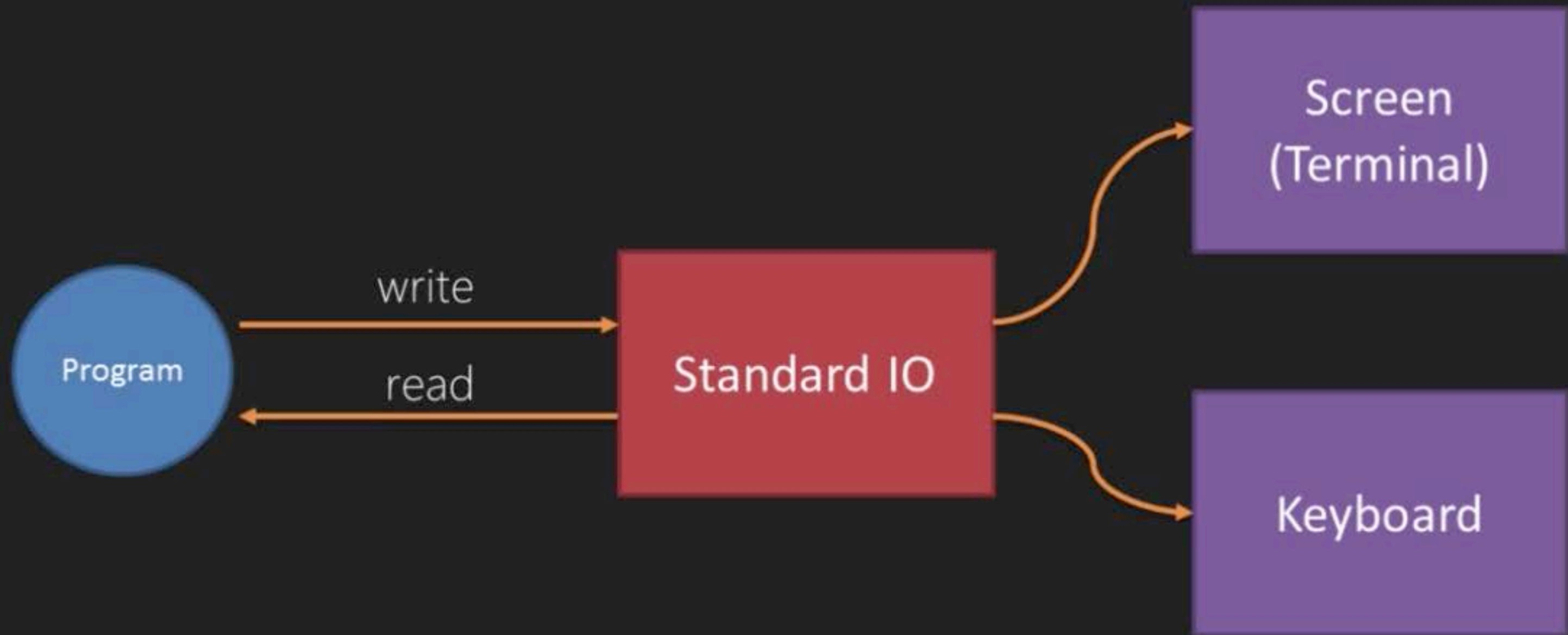




In this video, we are going to take a look at...

- What is the Standard IO
- Writing strings to the screen
- Reading from the keyboard

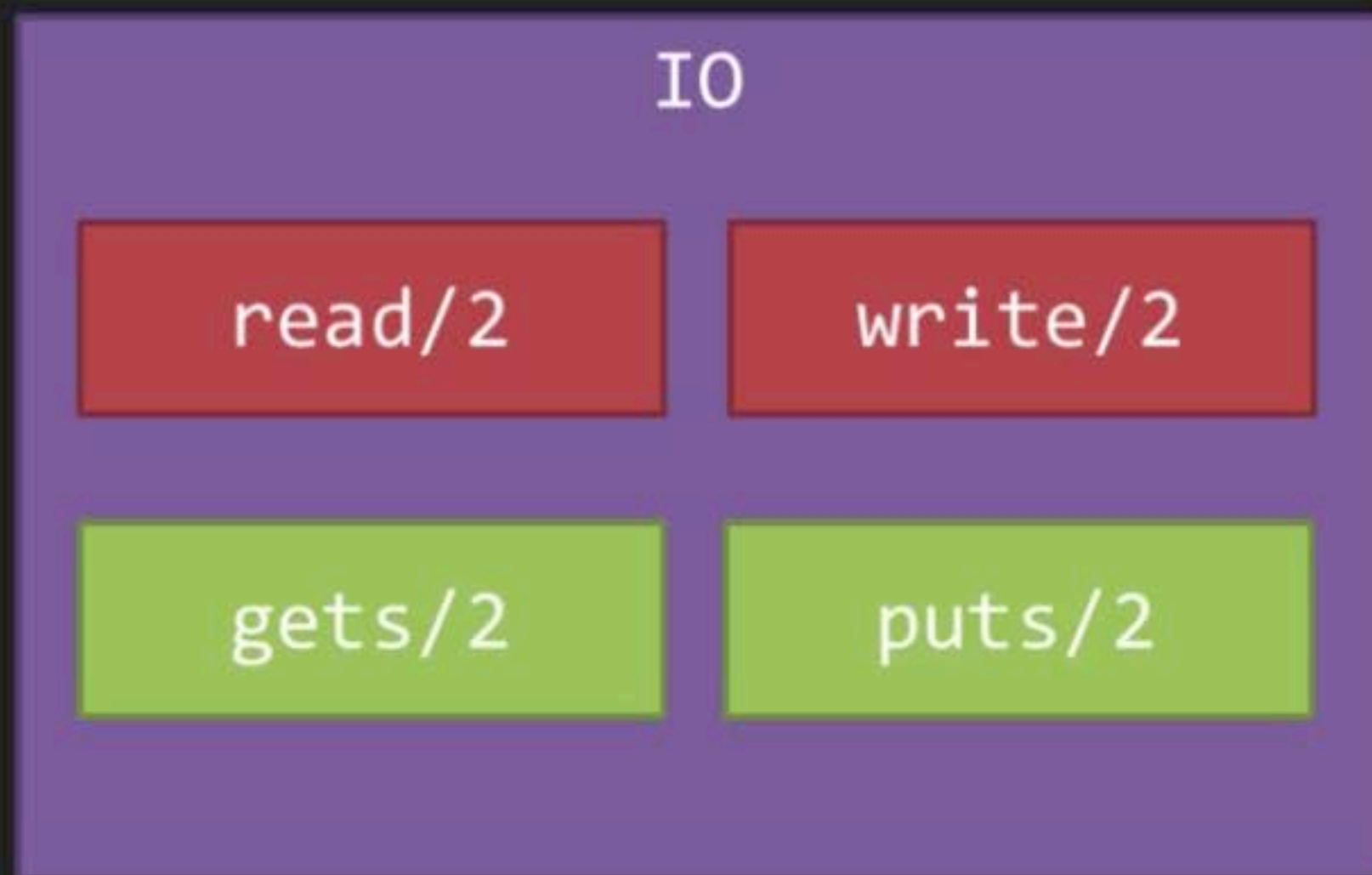
# The Standard IO



# IO Module



# IO Module



## Reading from the Keyboard

### Terminal

```
iex(1)> IO.gets("What is your name?: ")  
What is your name?:
```

## Reading from the Keyboard

### Terminal

```
iex(1)> IO.gets("What is your name?: ")  
What is your name?: John  
"John\n"  
iex(2)>
```



## Reading to the Screen

### Terminal

```
iex(1)> IO.puts("What is your name?: ")  
What is your name?: John  
"John\n"  
iex(2)> IO.puts("Hello!")
```



## Reading to the Screen

### Terminal

```
iex(1)> IO.puts("What is your name?: ")
What is your name?: John
"John\n"
iex(2)> IO.puts("Hello!")
Hello!
:ok
iex(3)>
```

# ANSI Formatting

IO.ANSI

red/0

white/0

blue/0

bold/0

italic/0

format/2

# ANSI Formatting

## Terminal

```
iex(1)> IO.puts([IO.ANSI.red, "Eli", IO.ANSI.blue, "xir"])
```

```
Elixir
```

```
:ok
```

```
iex(2)> IO.ANSI.format([:red, "Eli", :blue, "xir"]) |>
```

```
IO.puts
```

```
Elixir
```

```
:ok
```

```
iex(3)>
```

# Elixir: Scalable and Efficient Application Development

---

*João Gonçalves*

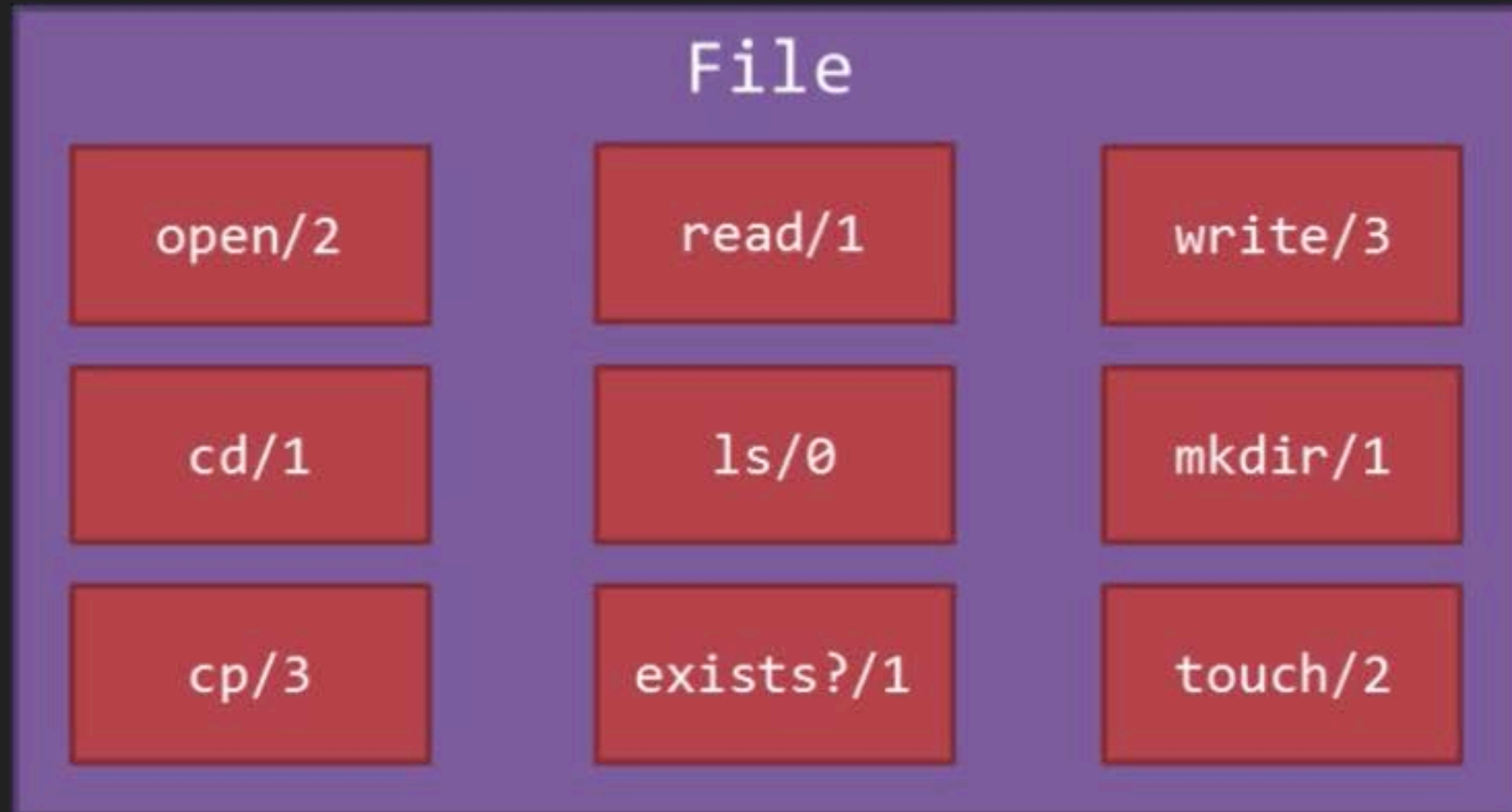
*Files*



In this video, we are going to take a look at...

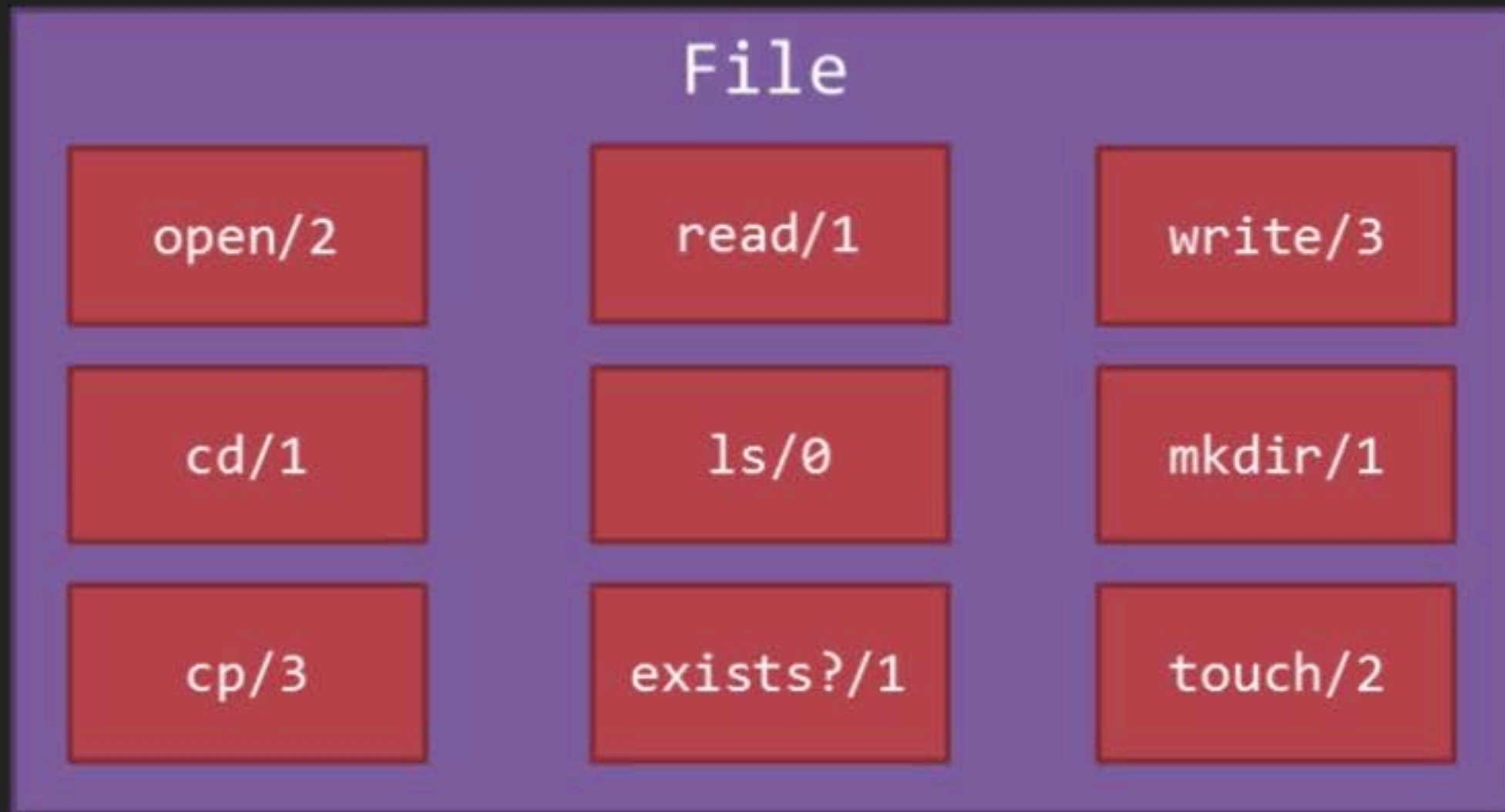
- How to manipulate files in Elixir

# The File Module



# The File Module

- Most of these functions are the same as the commands we'd use on a \*NIX command line





# Creating a Directory



notes

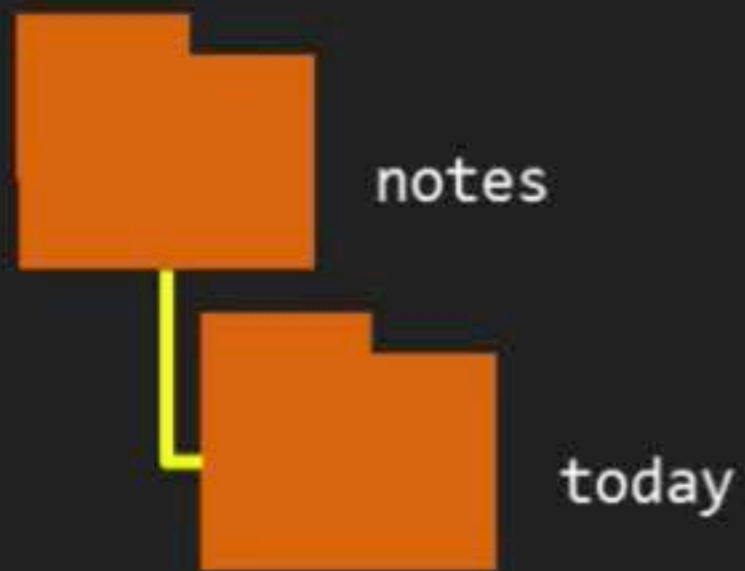
# Creating a Directory



notes

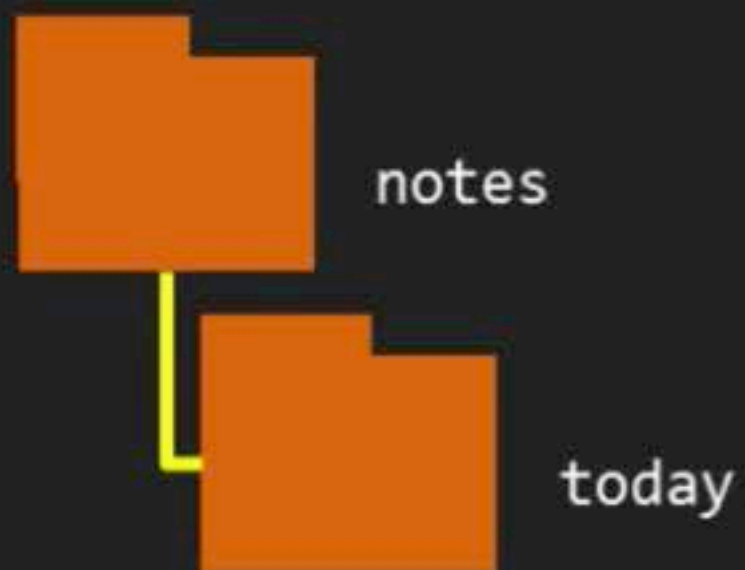
```
File.mkdir("notes/today")
```

# Creating a Directory



```
File.mkdir("notes/today")
```

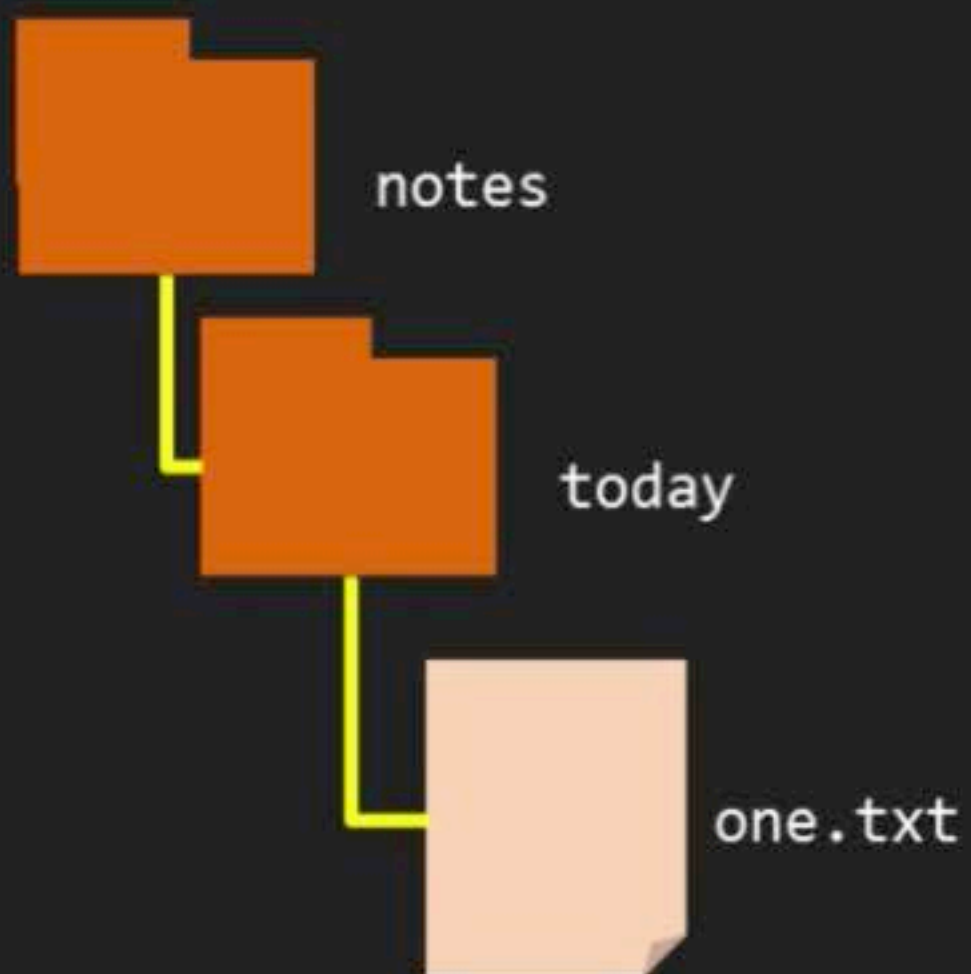
# Creating a Directory



```
File.mkdir("notes/today")
```

```
File.write("notes/today/one.txt", "Hello")
```

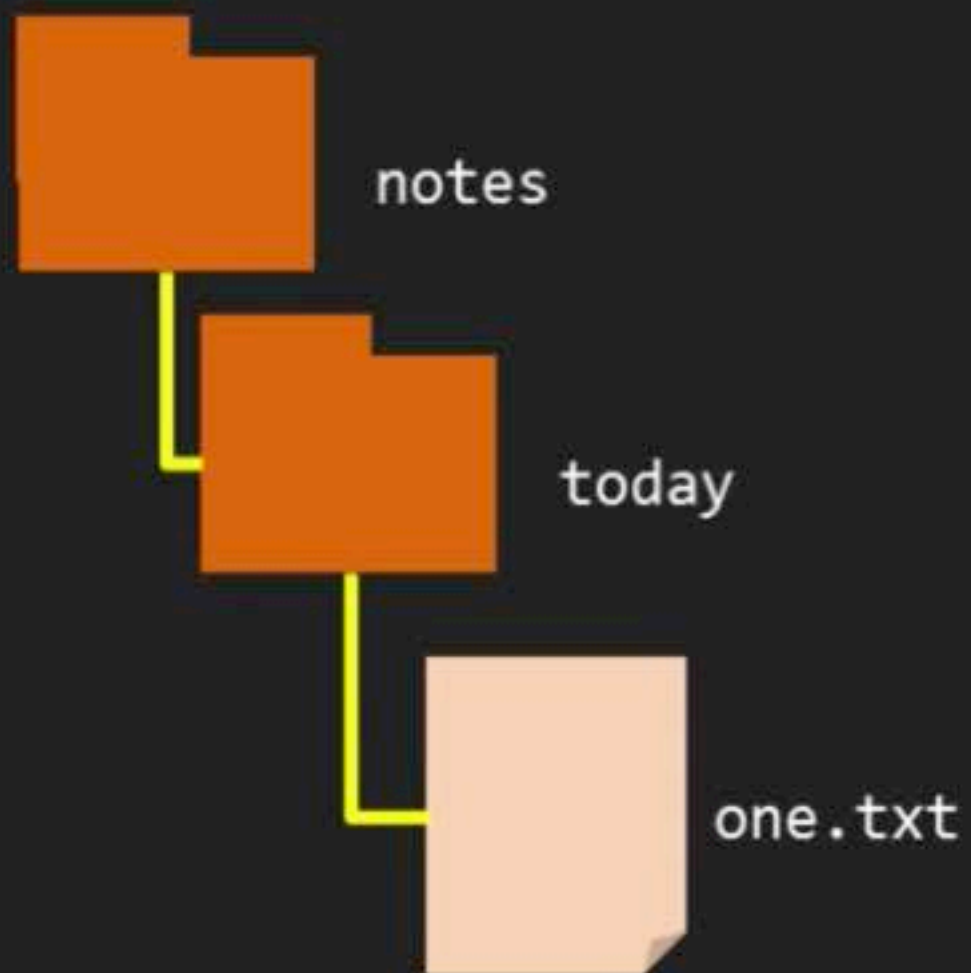
# Creating a Directory



```
File.mkdir("notes/today")
```

```
File.write("notes/today/one.txt", "Hello")
```

# Creating a Directory

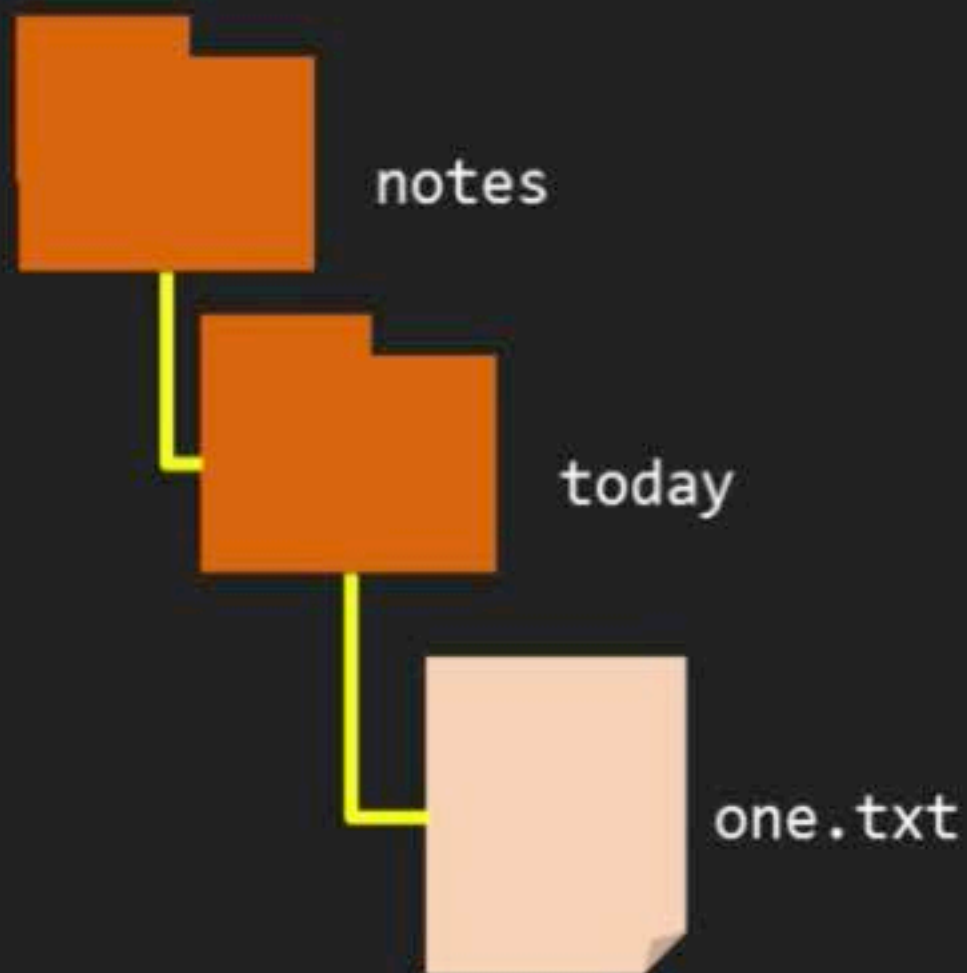


```
File.mkdir("notes/today")
```

```
File.write("notes/today/one.txt", "Hello")
```

```
File.read("notes/today/one.txt")
```

# Creating a Directory



```
File.mkdir("notes/today")
```

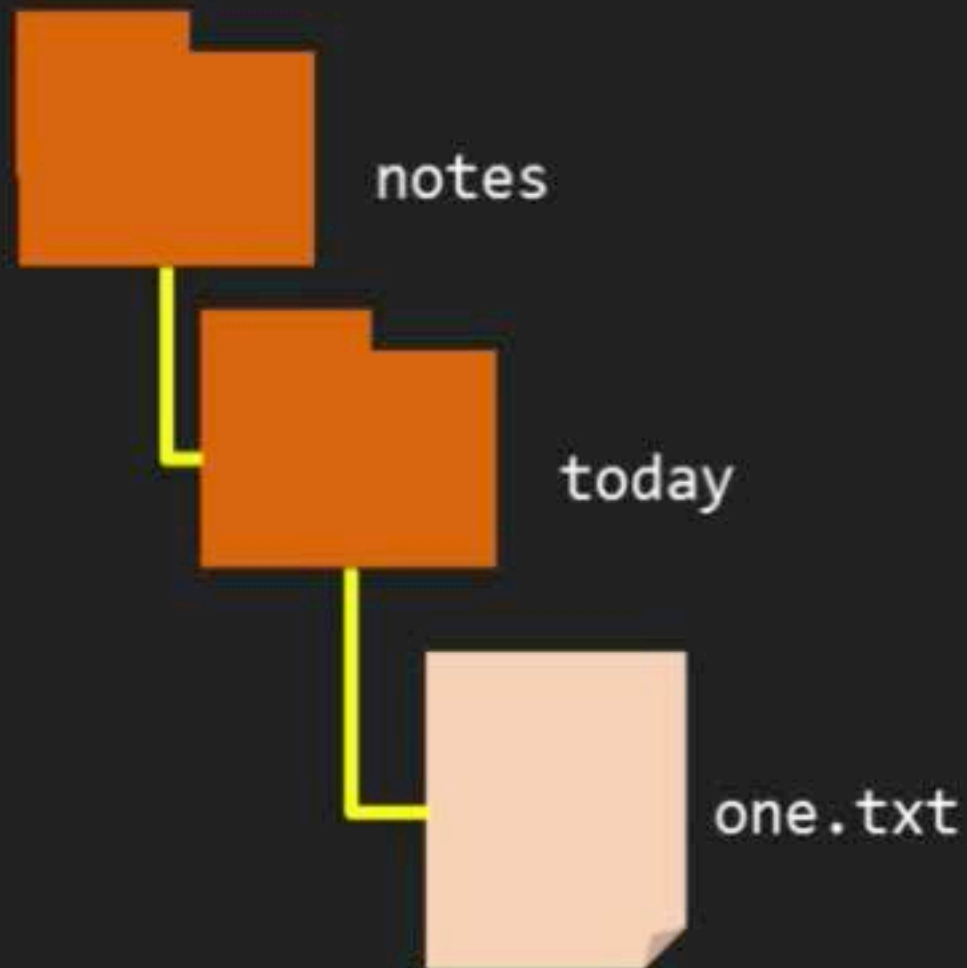
```
File.write("notes/today/one.txt", "Hello")
```

```
File.read("notes/today/one.txt")
```

```
{:ok, "Hello"}
```



## Deleting a File



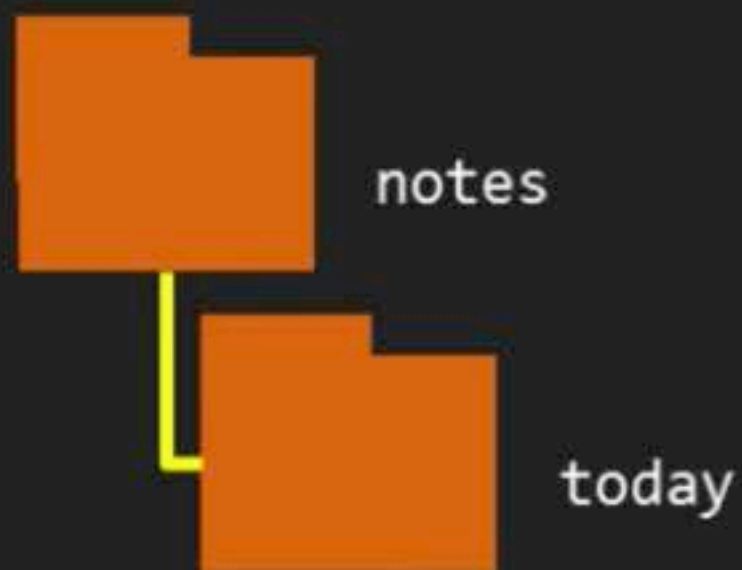
```
File.mkdir("notes/today")
```

```
File.write("notes/today/one.txt", "Hello")
```

```
File.read("notes/today/one.txt")
```

```
File.rm("notes/today/one.txt")
```

## Deleting a File



```
File.mkdir("notes/today")
```

```
File.write("notes/today/one.txt", "Hello")
```

```
File.read("notes/today/one.txt")
```

```
File.rm("notes/today/one.txt")
```

## Open + Close

- Use this principle when doing a lot of IO operations on a file

## Open + Close

```
def write_list(path, list) do
  with {:ok, file} <- File.open(path, [:write, :utf8]) do
    list
    |> Enum.each(fn(line) -> IO.write(file, line) end)

    File.close(file)
  end
end
```

# Quiz Time!

Quiz 6 | 2 Questions

Start Quiz

Skip Quiz >

Question 1:

**Which of the following function is included in the IO module?**

☐ delete

☐ view

☐ execute

☐ write

Question 2:

**Which of the following functions is used for accepting user input?**

☐ write

☐ read

☐ gets

☐ puts