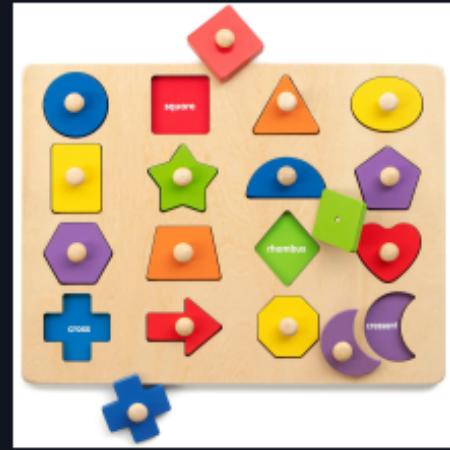


## ■ Data types

Mitsiu Alejandro Carreño Sarabia



## Agenda

- └─ Recap
- └─ Types
- └─ Instalation



## Recap



## Recap

- What is an expression?
- What is a value?
- What does this symbol means?

⇒

- Values are expressions?
- Expressions are values?
- What is imperative programming?
- What is state in computer science?

## Homework

**1. Bring your computer next session**

**2. Master your terminal:**

- Change to a specific directory
- Go to parent directory
- Print current directory

**3. Master your code editor:**

- Search in a single file
- Search in multiple files
- Know filename and file path of open file
- Go to definition

- Split screen
- Go to a specific line in a file
- Find and replace in a single/multiple files

 Homework

## 4. Master your keyword

- How to keypress () [] {}
- <https://monkeytype.com/>
- Practice PascalCase with shift key

## Functional programming

Functional programming allows reasoning about programs and their subcomponents in the same way that you would reason about a mathematical expression.

We're not just in the business of writing code, but correct code!



## Types

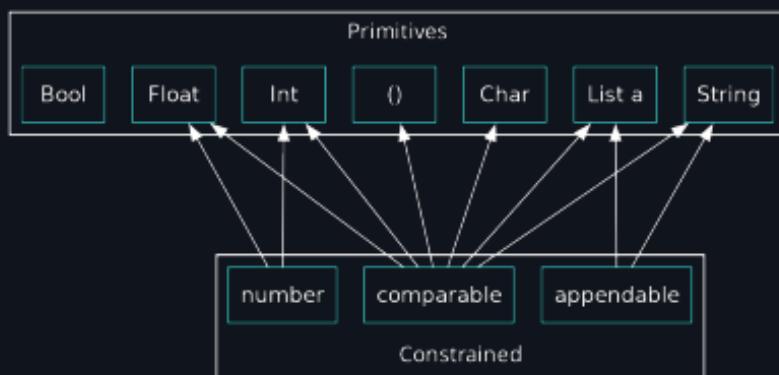
## Types guide structure (shape)

Functional programming places a great emphasis on types, which serve the purpose of documenting the purpose of code, and restricting the range of behaviors that a program is allowed to exhibit.

In this way, **types guide the structure of a program**, by providing clean interfaces for how different parts should interact, and what it should be allowed to do.

## Data types

We will be using a functional programming language called **Elm** which support the following data types:



## Data types

Type: Is a specification of the behavior of a piece of code. It **predicts** what a program is allowed to do.

To say that an expression e has type t we write:

$e : t$

For example:

$(5 + 2) * 3 : number$

We are communicating that the expression  $(5 + 2) * 3$  must produce a value of type number (either Int or Float)

## Data types

Tracing back our value definition:

| Value: The result of a calculation (a **final answer** that cannot be simplified further)

We can exemplify **values** for each data type:

*True : Bool*

*1 : Int*

*3.14 : Float*

*'a' : Char*

*"abc" : String*

## Data types & operators

Elm is a statically typed language, meaning that all typing rules are applied before the program is ever run Let's analyze how Elm enforces it's type rules:

$$(5 + 2) * 3 : \text{number}$$

The typing rule for +  
is:

$e1 + e2 : \text{number}$   
if  
 $e1 : \text{number}$   
and  
 $e2 : \text{number}$

We know

$5 : \text{Int}$   
and  
 $2 : \text{Int}$   
so  
 $5 + 2 : \text{Int}$

The typing rule for \*  
is:

$e1 * e2 : \text{number}$   
if  
 $e1 : \text{number}$   
and  
 $e2 : \text{number}$

## Data types & operators

Now lets learn a new operator `++`

The typing rule for `++`  
is:

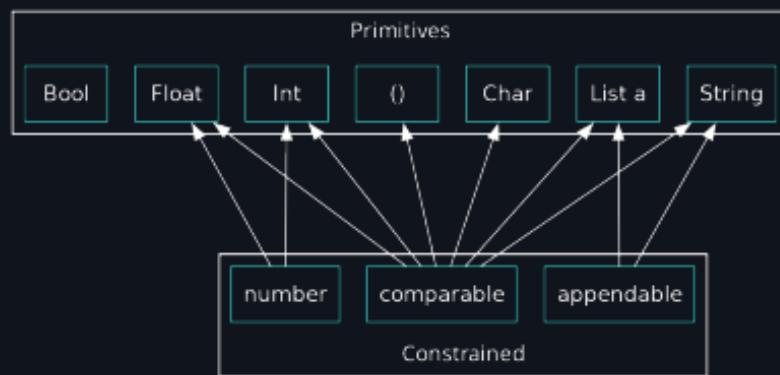
$e1 \text{ } ++ e2 : appendable$

if

$e1 : appendable$

and

$e2 : appendable$



Anyone can figure out a valid expression with `++`?

## Data types & operators

```
"Hello" : String + +"world" : String =>  
    "Helloworld" : String
```

## ■ Data types & operators

Finally let's analyze the expression:

*"Hello" ++2*

*"Hello" ++2 : appendable*

*if*

*"Hello" : appendable* 

*and*

*2 : appendable* 

So "Hello" ++ 2 does not have a type, and we say it's an **ill-typed expression**  
**Ill-typed programs are not evaluated**



## NodeJs

You can download and install the prebuilt <https://nodejs.org/en/download>

The screenshot shows the Node.js download page for Windows. At the top, there are dropdown menus for selecting the version (v24.12.0 (LTS)), platform (Windows), and package manager (Chocolatey). Below these are two informational messages: one about getting new features sooner by using the latest Node.js version, and another about supported installation methods. A code block displays the PowerShell commands to download Chocolatey, Node.js, and verify their versions:

```
1 # Download and install Chocolatey:  
2 powershell -c "irm https://community.chocolatey.org/install.ps1|iex"  
3  
4 # Download and install Node.js:  
5 choco install nodejs --version="24.12.0"  
6  
7 # Verify the Node.js version:  
8 node -v # Should print "v24.12.0".  
9  
10 # Verify npm version:  
11 npm -v # Should print "11.6.2".
```

At the bottom, it says "Chocolatey is a package manager for Windows. If you encounter any issues please visit: Chocolatey's website." Below this, there's a note about getting a prebuilt Node.js® for Windows, running a x64 architecture, with options for "Windows Installer (.msi)" and "Standalone Binary (.zip)".

Verify your installation with these two commands:

`node --version`

`npm --version`



Elm

Choose your os from: <https://guide.elm-lang.org/install/elm>

Verify your installation with the command:

```
elm
```

## Editor integrarion

Choose your editor and follow the instructions at:  
<https://github.com/elm/editor-plugins>

## Elm tooling

Run the command:

```
npm install -g elm-test elm-format elm-review
```