types.

not

types

TYPES, TYPE INFERENCE AND UNIFICATION

WHAT IS A TYPE?

collection of computable values that share some structural property

ex int String

 \dot{c} nt \rightarrow book

(int → int) → book

[a] -> a

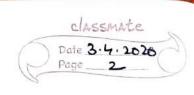
[a] *a -> [a]

& 3, true, \x + x }

even integers

 ξ_f : int \rightarrow int

1x73 => 6(x) > x* (x+1) 3



distinction between

-sets of values that are types, and

- Sets that are not types

is language dependent.

types help identify & prevent errors.

TYPE ERRORS ARE LANGUAGE DEPENDENT

array out of bound access

G/ C++ runtime error

Ocamel/ Java dynamic type error.

10 null pointer defregerence

C/Gtt run-time error

Ocamb pointers are hidden inside datatypes