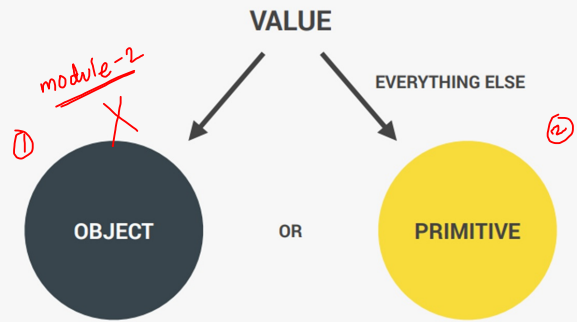
OBJECTS AND PRIMITIVES



THE 7 PRIMITIVE DATA TYPES

small datatypes

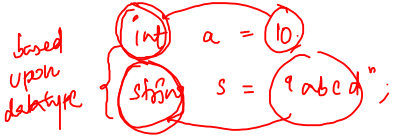
1. **Number:** Floating point numbers 👉 Used for decimals and integers `let age = 23;`
2. **String:** Sequence of characters 👉 Used for text `let firstName = 'Jonas';`
3. **Boolean:** Logical type that can only be `true` or `false` 👉 Used for taking decisions `let fullAge = true;`

4. **Undefined:** Value taken by a variable that is not yet defined ('empty value') `let children;`
5. **Null:** Also means 'empty value' ← both are similar (empty)
6. **Symbol (ES2015):** Value that is unique and cannot be changed *[Not useful for now]*
7. **BigInt (ES2020):** Larger integers than the Number type can hold

we will not come across these much

☝ **JavaScript has dynamic typing.** We do ***not*** have to manually define the data type of the value stored in a variable. Instead, data types are determined **automatically**.

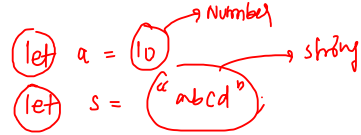
Value has type, NOT variable!



- C++/Java doesn't automatically know about the datatype of value
= Variables are attached to datatypes

```
int a = 10;
a = 20;
a = "xyz";
```

a can only hold Integer (int)



- automatically knows
- values are attached to datatypes

```
let a = 10;
a = 20;
a = "xyz";
```

a can hold anything because datatypes are for values.

Other ways to declare a variable :

1. let year = 2001;
2. const year = 2001;
   - → cannot be changed
   - → cannot be empty / undefined.
3. var year = 2001; ✗

(1) Always 1st prefer using const.

(2) If you feel that the value of a variable needs to be changed then only use let.

→ Assume var, let works the same for now, differences are in scopes (later in functions lecture)

→ var is not at all used in present time. Only use let / const.

9:05 PM — 9:20 PM
BREAK


back tick
~ `
! 1
@ 2
# 3
$ 4
% 5
^ 6
& 7
Q W E R T Y
A S D F G H
Z X C V B N
US English layout
Proudly made by Keyshorts.

★ operator precedence :

BODMAS

true - on - 1

false - off - 0

const currYear = 2024

currYear - 1991 > currYear - 2018

① 2024 - 1991 > 2024 - 2018

33 > 2024 - 2018

false - 2018

0 - 2018

= - 2018

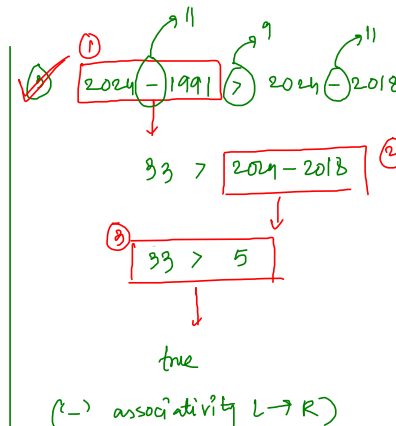
② 2024 - 1991 > 2024 - 2018

2024 - 1991 > 6

2024 - true

2024 - 1

= 2023



(-) associativity L → R