12 → **IS Number** 2 ((Cefer fg 6q. Primitive values

1 → Always 6q bit Floating Point

2 → **NaN** → Is Reserved → Indicate → Number → Not → legal no.
Word

3 eg:
```
let x = 100 / "Hello";
document.write(x);
```
→ o/p → NaN
→ If string → result → No
→ Numeric

4 → **is NaN()** → Global IS
Function → To find out → if → value → is not a number.

5 Eg:
```
let x = "a";
document.write(isNaN(x));
```
→ o/p: True

6

7 True → Not a No
False → Is a no.

→ NaN → as → Mathematical operation
It → → Result → also ne → NaN
or
NaN
(If No)

NaN	It is a number	
typeof	number	eg →

eg →
```
let x = NaN
document.write(typeof(x))
```
o/p: number

11

Thursday

↱) Number
(─) type of ─) number   outside
↱) Infinity / ─Infinity   Largest possible no ↑
                            ↱) no
Value ─) Js ─→ return ─) if ─) calculate

↱) **Hexadecimal** :- If ⁿ preceded by ──→ ⬚ 0 x ⬚

eg. ⎡ let x = 0 x FF ; ⎤

↱) By default ᵃ) Js ⁿ) no ᵃˢ) Base 10 decimal

Hexadecimal	Base 16
Decimal	Base 10
Octal	Base 8
Binary	Base 2

↱) **Number as objects**   ᶜ) ⎡ let (variable name) = new Number(); ⎤

ᵉ) BigInt ↱) Big Int+c ↱) Append ⓝ to end of integer
         ↓↑
         typeof
         ↳ BigInt

↱) **Number Methods**

Methods	Return Number as
ᵃ) to string()	String
ᵇ) to Exponential()	Written ⁿ Exponential Notation
ᶜ) to Fixed()	written ⁿ) No of decimals
ᵈ) to Precision()	written ⁿ Specified length
ᵉ) value of()	Number

eg:- 1) (refer also pg 75)
     to string()   eg:- ⎡ let x = 123 .
                         let y = x . to string() ;

No, ⁿ 123   document . write (" x is . " + y
type string ──→   + " and type is " + typeof (y)) ;

2) to Exponential() parameter

(^) Optional ☑ defines no of digits point

(^) If don't specify → won't round behind decimal

let x = 9.656 109 ;
let y = x . to Exponential (4) ;

O/p : 9.6561 e +0
type : string

☐ Change 4 to any no (^) Change 4 to

$\frac{1}{3}$ → 9.656 e +0

2 → 9.66 e +0

3) to Fixed() ☑ Rounds no to given no of digits.

(^) to Fixed(2) ☑ perfect for working with money.

(^) Convert Variable to number (^) Number() (^) Convert value is variable to number

↳ parse Int()

parse string return whole no
Space allowed
Only 1st no is returned

↳ parse Float()

) parse String & return whole no
) Space allowed
Only 1st number allowed

(^) Js Type Conversion (^) Use of Js function
(^) Auto → Js itself

(^) Number() (^) Numeric String → eg : 3.14 ⟶ 3.14

Empty string (^) eg : "" ⟶ 0

Non Numeric String (^) eg : "John" ⟶ NaN

(^) Convert Number to string (^) String() ⟶ ⌠ global method

(^) to String()
↳ String()

JUNE   2020
Su Mo Tu We Th Fr Sa
     1   2   3   4   5   6
7   8   9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

(^) Convert Date to Number (^) Number()
(^) get Time()
↳ to string()

449

13

165-201 | Week 24

Saturday

9

᷁ Convert Boolean to Number

᷁ Number()

to String ᷁ String()

⊕ or
String()