Session – 4 31 July 2022

~~Cons~~ console.log ("Hello World");

⇒ node index.JS

console.log (2+2) ⇒ node index.JS

num = 3 num ⇒ variable

console.log ~~(②)~~ (num) = assigned

3 Stored a value

$\boxed{3}$ → Variable

num

⇒ num=3

num = 7 output – 7

console.log (num)

comment → // Single line 4 $\boxed{\text{Ctr+/}}$

/* } multiline

*/

text is called string.

num = 7 // number

num = "seven" // string

⇒ let Keyword

DATE : / /

PAGE NO. :

① Compiler
② Interpreter

High level language → Machine language

Compiler – A compiler is a computer programme that translate computer code written in one programming language to another language. use → High level to low level.
Single run.

Interpriter – A code run line by line

Every language
- Structure → Syntax
- Variable → Data
- Decison →
- repeat → loops

Single quote
double qoute
back qoute
JS

→ Syntax error → Syntax error
→ Logical erro → code run but output wrong

* let declaring variable
↓
other language → int, var.

* "use Strict" one top line

1) Statically typed language
whenever whe a create a variable
show the type. Ex - number, string, boolean

"A statically-type language is (suchas C++, C, Java) where
variable types are known as compile type.
"Compile time"

2) Dynamically typed language
Dynamally typed language are those (like Java script)
where at a run time based value.
'runtime'

⇒ type check - it is check a datatypes

```
let x = 6
console.log (x)
console.log (typeof x)
```

⇒ Conversion - Converted to a new variable and another datatype.

```
let value= Number ("987");
console.log (~~type of value~~);
             (typeof value);

let value = Number ("987 Navin");

console.log (value, typeof value);
```

output

```
Nan Number
```

NaN— Not a Number (illegal number)
Or
incorrect number

⇒ parseInt → It is find para only number
to

let value = parse Int ("987 number");
console.log (~~value type of~~ typeof value);

increament operator

let num = 1
// num = num+2 } same & short
num+ = 2
console.log (num);

// num = num +1
num + = 1
num++

All Same

++num — pre increament

⇒ num = 8
num1= num++ past → ① Assing
② Increment

output
9,8

num1 = ++num pre → ① Increamt
② Assing

Relational operator

let num1 = 8
let num2 = 6
let result = num1 > num2
console.log (result);

'==' in programming
'=' in general

⇒ Same question of one string "8"

double '==' compare the values not types
triple '===' compare the "values and types" as well as

templete literale

- ${} & back sticks

This ~~us~~ ~~usea~~ ~~ang~~ any print ~~gro~~ a line
Value used →

let user1 = "[illegible] 8
user2 = 7
result1 = num1 + num2

console.log (` The addition ${num1} & ${num2}
is the ${result}')

backtype

}

Condition

```
let num = 5
  if (num % 2===0)
        console.log ("Even");
 else
        console.log ("odd");
```

loops — While, do while, for

While

```
let i = 1
 while (i<=5)
   {
     console.log ("Hello");
     i++;
   }
```

Debugging check

Assignment

⇒ Calculator with all basic operato
⇒ print all even number from 1 to 100
⇒ Find a given number is prime or not

⇒ function, object, oops concept, Arrays