

Session - 4
~~XXXXXX~~31 July 2022~~console.log~~ console.log("Hello World"); \Rightarrow node index.js

console.log(2+2)

 \Rightarrow node index.js

num = 3

num \Rightarrow variableconsole.log(~~(3)~~)(num) \equiv assigned
3 stored a value

3

num

 \rightarrow Variable \Rightarrow

num = 3

num = 7

output - 7

console.log(num)

comment \rightarrow //

/*

*/

}

Single line

multiline

4

Ctrl + /

text is called string.

num = 7

// number

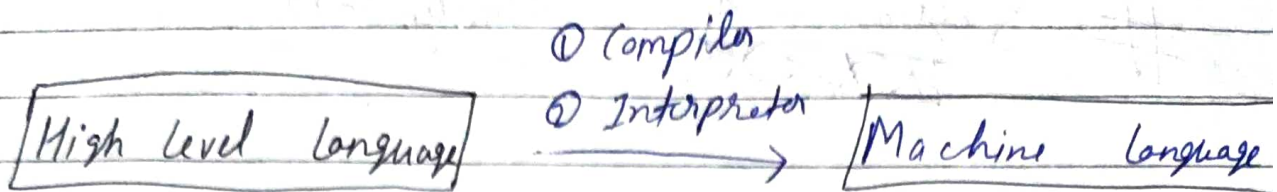
num = "seven"

// string

⇒ Let Keyword

DATE: / /

PAGE NO.:



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

Single run.

Interpreter - A code run line by line

Every language

- Structure → Syntax
- Variable → Data
- Decision →
- Repeat → loops

Single quote JS
double quote
back quote

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

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

* //66 use Strict " one top line

1) Statically typed language

Whenever we create a variable
show the type. (x = number, string, boolean)

6 A statically-typed language is (such as C++, C, Java) where
variable types are known as compile time.
66 compile time

2) Dynamically typed language

Dynamically typed language are those (like javascript)
where at a runtime 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(typeof 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

```
let value = parseInt("987 number");
console.log(value);
```

increment
operator

```
let num = 1
```

```
// num = num + 2 } same & short
```

```
num += 2
```

```
console.log(num);
```

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

```
num += 1
```

```
num++
```

All
Same

++num

— pre increment

⇒

```
num = 8
```

```
num = num++
```

←

post → ① Assign

② Increment

```
num = ++num
```

pre → ① Increment

② Assign

output
9, 8

Relational operator

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

'==' in programming
'=' in general

⇒ Same question of one string "68"

double '==' Compare the values not types

triple '===' Compare the "values and types" as well as

template literals

- \${ } back stick

This is used any any print a line
value used →

```
let user1 = "Avatar 8"
```

```
user2 = 7
```

```
result1 = num1 + num2
```

console.log (1) The addition `\${num1} + \${num2}`
is the `\${result}`)

backtype

{

DATE: / /

PAGE NO.:

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
checkAssignment

- ⇒ Calculator with all basic operators
- ⇒ print all even numbers from 1 to 100
- ⇒ find a given number is prime or not

⇒ function, object, oops concept, Arrays