

Using the Console

Uses **REPL**

Read-Evaluate-Print-Loop

→ vs code
.html
.css
.js }



top



3 Issues:  3

> 1 

< 1

> 2

< 2

> 1 + 2

< 3

> |

Using the Console

→ vs code

Uses REPL

Read-Evaluate-Print-Loop



computes
calculations

.html
.css
.js



clear → cmd + K
ctrl + d

What is a Variable?

A variable is simply the name of a storage location.

memory

$a = 10$
 ~~$age = 23$~~
 \downarrow \downarrow
 var val
 $name = "Tony"$

What is a **Variable**?

A variable is simply the name of a storage location.



variable

memory



Elements

Co



top



3 Issues:  3

> a = 10

< 10

> b = 5

< 5

> age = 23

< 23

> name = "tony"

< 'tony'

> a

< 10

>

Data Types in JS

Primitive Types

- Number
- Boolean
- String
- Undefined
- Null
- Bigint
- Symbol

3 Issues:  3

```
> a = 25
```

```
< 25
```

```
> typeof a
```

```
< 'number'
```

3 Issues:  3

> typeof 5.9

< 'number'

> typeof 'a'

< 'string'

Numbers in JS

- Positive (14) & Negative (-4)
- Integers (45, -50)
- Floating numbers - with decimal (4.6, -8.9)

```
< 16
```

```
> typeof a
```

```
< 'number'
```

```
> price = 99.99
```

```
< 99.99
```

```
> typeof price
```

```
< 'number'
```

```
> displacement = -25
```

```
< -25
```

```
> typeof displacement
```

```
< 'number'
```

```
> a = 10
```

```
< 10
```

```
> b = 5
```

```
< 5
```

```
> a + b
```

```
< 15
```

```
> a - \
```

```
✖ Uncaught SyntaxError: Unexpected token \
```

```
> a - b
```

```
< 5
```

```
> a * b
```

```
< 50
```

```
> a / b
```

```
< 2
```

```
> 10 * 2
```

```
< 20
```

```
> a = 0.999999999999999
```

< 0.999999999999999

```
> a = 0.9999999999999999999999999999999
```

nehagupta7385@gmail.com

Operations in JS

```
a = 20
```

```
b = 10
```

```
//addition
```

```
sum = a + b
```

```
//subtraction
```

```
diff = a - b
```

```
//multiplication
```

```
prod = a * b
```

```
//division
```

```
div = a / b
```

```
//modulo
```

```
rem = a % b
```

- Modulo (remainder operator)

$$12 \% 5 = 2$$

- Exponentiation (power operator)

$$2^{**}3 = 8$$

NaN in JS

The NaN global property is a value representing Not-A-Number.

0 / 0

NaN - 1

NaN * 1

NaN + NaN

> 0 / 0

< NaN

> typeof NaN

< 'number'

>

```
> typeof NaN
```

```
< 'number'
```

```
> NaN + 1
```

```
< NaN
```

```
> NaN - 1
```

```
< NaN
```

```
> NaN * 1
```

```
< NaN
```

```
> NaN * NaN
```

```
< NaN
```

Operator Precedence

This is the general **order** of solving an expression.

()

***, /, %**

+, -

Operator Precedence

Evaluate

$$(2 + 1) * 3 \longrightarrow$$

ehagupta7385@gmail.com

$$3 / 1 + 2 ** 2$$

$$4 + 1 * 6 / 2$$

let keyword

Syntax of declaring variables

```
let age = 23 ;
```

```
age = age + 1 ;
```

```
let cgpa ;
```

```
cgpa = 8.9
```

```
let num1 = 1 ;
```

```
let num2 = 2 ;
```

```
let sum = num1 + num2 ;
```

age

23

age = age + 1;

24

age

24

age = age - 1;

23

age

23

let cgpa = 8.9;

undefined

cgpa

8.9

cgpa = cgpa * 10;

89

|

```
> let num1 = 1;
```

```
< undefined
```

```
> let num2 = 2;
```

```
< undefined
```

```
> let finalSum = num1 + num2;
```

```
< undefined
```

```
> finalSum
```

```
< 3
```



const keyword

values of **constants** can't be changed with re-assignment & they can't be re-declared

```
const year = 2025;
```

```
year = 2026      // Error
```

```
year = year + 1  // Error
```

```
const pi = 3.14 ;
```

```
const g = 9.8 ;
```



```
const newNum = 5;
```




```
undefined
```

```
newNum
```

```
5
```

```
newNum = 10;
```

```
► Uncaught TypeError: Assignment to constant variable  
  at <anonymous>:1:8
```

 top  Filter Default levels  3

```
> const newNum = 5;  
< undefined  
> newNum  
< 5  
> newNum = 10;  
✖ ▶ Uncaught TypeError: Assignment to constant variable.  
   at <anonymous>:1:8  
> newNum  
< 5  
> newNum = newNum + 1;  
✖ ▶ Uncaught TypeError: Assignment to constant variable.  
   at <anonymous>:1:8
```

```
> const pi = 3.14;
```

```
< undefined
```

```
> let rad = 4;
```

```
< undefined
```

```
> let area = pi * r ** 2;
```

```
✖ ▶ Uncaught ReferenceError: r is not defined  
   at <anonymous>:1:17
```

```
> let area = pi * rad ** 2;
```

```
< undefined
```

```
> area
```

```
< 50.24
```

var keyword

Old Syntax of writing variables

```
var age = 23 ;
```

```
var cgpa = 8.9 ;
```

```
var num1 = 1 ;
```

```
var num2 = 2 ;
```

```
var sum = num1 + num2 ;
```

```
> var x = 5;
```

```
< undefined
```

```
> var y = 10;
```


```
< undefined
```

```
> var sum = x + y;
```

```
< undefined
```

```
> sum
```

```
< 15
```



```
>
```

Assignment Operators

`age = age + 1`

`age += 1`

`age = age - 1`

`age -= 1`

`age = age * 1`

`age *= 1`

Unary Operators

age = age + 1

age += 1

age++ // increment

age = age - 1

age -= 1

age-- // decrement

binary operators
2 operands

$a + b$

$a \times b$

a / b

$a = b$

Unary Operators

Pre-increment (Change, then use)

```
let age = 10 ;
```

```
let newAge = ++age ;
```

Post-increment (Use, then change)

```
let age = 10 ;
```

```
let newAge = age++ ;
```


`++a` (pre-inc)

`a++` (post-inc)

`--a` (pre-dec)

`a--` (post-dec)

Identifier Rules

All JavaScript variables must be identified with **unique names** (identifiers).

- Names can contain letters, digits, underscores, and dollar signs. (no space)
- Names must begin with a letter.
- Names can also begin with \$ and _ .
- Names are case sensitive (y and Y are different variables).
- Reserved words (like JavaScript keywords) CANNOT be used as names.

```
> let price;
```

```
< undefined
```

```
> let price123
```

```
< undefined
```

```
> let price$;
```

```
< undefined
```

```
> let price_;
```

```
< undefined
```

```
> let price-;
```

```
✖ Uncaught SyntaxError: Unexpected token '-'
```

```
> let old price;
```

```
✖ Uncaught SyntaxError: Unexpected identifier 'price'
```

```
>
```

  top ▼  Filter

> let age;

< undefined

> let _age;

< undefined

> let \$age;

< undefined

> let 1age;

✖ Uncaught SyntaxError: Invalid or unexpected token

>

camelCase

Way of writing identifiers

cameCase **(JS naming convention)**

snake_case

PascalCase

camelCase

fullName

Way of writing identifiers

camelCase (JS naming convention)

camelCase

let fullName;

snake_case

let full_name;

PascalCase

let FullName;

Boolean in JS

Boolean represents a truth value -> **true or false** / yes or no

```
let age = 23 ;
```

```
let isAdult = true ;
```

```
let age = 13 ;
```

```
let isAdult = false ;
```

top ▾

Filter

> let str = "svjadbvjabdjvbjdfbvj asdmfksdnfgsdfg"

< undefined

> str = "asdfnaskjdfnjksd&%\$#89349821374";

< 'asdfnaskjdfnjksd&%\$#89349821374'

> str = "asdfnaskjdfnjksd&%\$#89349821374asdfasdf 23412zssdf";

< 'asdfnaskjdfnjksd&%\$#89349821374asdfasdf 23412zssdf'

> str

< 'asdfnaskjdfnjksd&%\$#89349821374asdfasdf 23412zssdf'

>

String in JS

Strings are **text** or sequence of characters

```
let name = "Tony Stark" ;
```

```
let role = 'ironman' ;
```

```
let char = 'a' ;
```

```
let num = '23' ;
```

```
let empty = " " ;
```

Practice Qs

Qs. Find the errors in the following code ?

a) `let ①age = 5;` ← `let age1 = 5;`
`let ②age = 10;` ← `let age2 = 10;`

b) `let marks = 75;` ←
`let isPass = True;` ←
true

c) `let isPass = ①true①;` } ' '] String
↑ boolean true " "

```
> "tony" + " " + "stark"
< 'tony stark'
> let firstName = "tony";
< undefined
> let lastName = "stark";
< undefined
> let fullName = firstName + lastName;
< undefined
> fullName
< 'tonystark'
> fullName = firstName + " " + lastName;
< 'tony stark'
> fullName
< 'tony stark'
> "tony" + 1
< 'tony1'
>
```



top



Filter

```
> let name = "TONY STARK";
```

```
< undefined
```

```
> name
```

```
< 'TONY STARK'
```

```
> name[0]
```

```
< 'T'
```

```
> name[1]
```

```
< 'O'
```

```
> name[2]
```

```
< 'N'
```

```
> name[3]
```

```
< 'Y'
```

```
> name[4]
```

```
< ' '
```

```
> name[1000]
```

```
< undefined
```

```
>
```

String Indices

```
let name = "TONY STARK" ;
```

| | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <u>T</u> | <u>O</u> | <u>N</u> | <u>Y</u> | <u> </u> | <u>S</u> | <u>T</u> | <u>A</u> | <u>R</u> | <u>K</u> |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

name[0] -> 'T'

name[1] -> 'O' ...



null & undefined in JS

undefined

A variable that has not been assigned a value is of type undefined.



```
> let a;  
< undefined  
  
> a  
< undefined
```

null

The null value represents the **intentional** absence of any object value.

To be explicitly assigned.

```
> let a = null;  
< undefined  
  
> a  
< null
```

  | top ▼ |  | Filter

```
> let name = "TONY STARK";
```

```
< undefined
```

```
> name[name.length];
```

```
< undefined
```

```
> name[name.length-1];
```

```
< 'K'
```


What is TypeScript?

Static Typed, where JS is dynamic typed

Designed by Microsoft



Concatenation

adding strings together

"tony" + " " + "stark" = "tony stark"

"tony" + 1 = "tony1"



top



Filter

```
> "tony" + " " + "stark"
```

```
< 'tony stark'
```

```
> let firstName = "tony";
```

```
< undefined
```

```
> let lastName = "stark";
```

```
< undefined
```

```
> let fullName = firstName + lastName;
```

```
< undefined
```

```
> fullName
```

```
< 'tonystark'
```

```
> fullName
```

```
< 'tonystark'
```

Delta Batch (Web Development)

Practice Qs
✓ video

String in JS
✓ video

String Indices
✓ video

null & undefined in JS
✓ video

Practice Qs
✓ video

20. JavaScript (Part 2)

21. JavaScript (Part 3)

22. JavaScript (Part 4)

moon jira Seat Matrix 2022 Department of Co... fon



Elements

Console

Sources

N



top



Filter

```
ony" + " " + "stark"
```

```
ony stark'
```

```
let firstName = "tony";
```

```
undefined
```

```
let lastName = "stark";
```

```
undefined
```

```
let fullName = firstName + lastName;
```

```
undefined
```

```
llName
```

```
tonystark'
```

```
llName = firstName + " " + lastName;
```

```
ony stark'
```

```
llName
```

```
ony stark'
```

```
ony" + 1
```

```
ony1'
```

Practice Qs

Qs. Find the errors in the following code ?

a)

```
let 1age = 5;
let 2age = 10;
```




 ← ←

b)

```
let marks = 75;
let isPass = True;
```

c)

```
let isPass = 'true';
```

  | top ▼ |  | Filter

```
> let sentence = 'this is "apple"';  
< undefined  
  
> sentence  
< 'this is "apple"'  
  
> let sentence = "this is 'apple'";  
< undefined  
  
> sentence  
< "this is 'apple'"  
  
> let sentence = 'this is 'apple'';  
✖ Uncaught SyntaxError: Unexpected identifier 'apple'  
>
```



top ▼



Filter

```
> let firstName = "Tony";
```

```
< undefined
```

```
> firstName
```

```
< 'Tony'
```

```
> typeof firstName
```

```
< 'string'
```

```
> let lastName = 'Stark';
```

```
< undefined
```

```
> lastName
```

```
< 'Stark'
```

```
> typeof lastName
```

```
< 'string'
```

```
> let char = 'a';
```

```
< undefined
```

```
> char
```

```
< 'a'
```

```
> typeof char
```

```
< 'string'
```

TS

$a = 5$ ✓

$a = \text{true}$ X ← ✓ IS

$a = 10$ ✓

Elements Console Sources Network Perfo

top Filter

```
> typeof lastName
```

```
< 'string'
```

```
> let char = 'a';
```

```
< undefined
```

```
> char
```

```
< 'a'
```

```
> typeof char
```

```
< 'string'
```

```
> let num = '123';
```

```
< undefined
```

```
> typeof num
```

```
< 'string'
```

```
> num
```

```
< '123'
```

```
> let emp = " ";
```

```
< undefined
```

```
> typeof emp
```

```
< 'string'
```

```
> emp = "";
```

```
< ''
```

```
> typeof emp
```


 top  Filter

```
> "shraddha".length
```

```
< 7
```

```
> "shraddha"[6]
```

```
< 'a'
```

```
> "shraddha"[0]
```

```
< 's'
```

```
>
```

```
> name[1]
```

```
< '0'
```

```
> name[2]
```

```
< 'N'
```

```
> name[3]
```

```
< 'Y'
```

```
> name[4]
```

```
< ' '
```

```
> name[1000]
```

```
< undefined
```

```
> name.length
```

```
< 10
```

```
> typeof name.length
```

```
< 'number'
```

```
> name[name.length];
```

```
< undefined
```

  | top ▼ |  | Filter Default le

```
> isAdult = True;
```

```
✖ ▶ Uncaught ReferenceError: True is not defined  
   at <anonymous>:1:1
```

```
> isAdult = 'True';
```

```
< 'True'
```

```
> typeof isAdult
```

```
< 'string'
```

```
> true
```

```
< true
```

```
> false
```

```
< false
```

```
>
```

```
> let name;
```

```
< undefined
```

```
> name
```

```
< undefined
```

```
> typeof name
```

```
< 'undefined'
```

```
> name = "apnacollege";
```

```
< 'apnacollege'
```

```
> name[1000];
```

```
< undefined
```

```
> let a = null;
```

```
< undefined
```

```
>
```



top



Filter

```
> let a = undefined;
```

```
< undefined
```

```
> a
```

```
< undefined
```

```
> let a;
```

```
< undefined
```

```
> let a = null;
```

```
< undefined
```

Practice Qs

Qs. Declare your name as a string and print its length in JS.

Qs. Declare your first name as a string and print its first character.

Qs. Declare your first name as a string and print its last character.

Qs. What is output of following code :

```
"apnacollege"+123|
```

Qs. What are lengths of an empty string & a string with a single space?