## Using the Console

**Uses REPL** 

Read-Evaluate-Print-Loop

. ntml ?
. css



# 3 Issues: 📃 3

- > 1
- √ 1
- > 2
- √ 2
- > 1 + 2
- √ 3
- >

## **Using the Console**

Uses REPL

Read-Evaluate-Print-Loop

comptets

, rume

. c s s . **i** s clear - cond + K
ctrl +d

#### What is a Variable?

A variable is simply the name of a storage location.

a=10

crge = 23

Tral

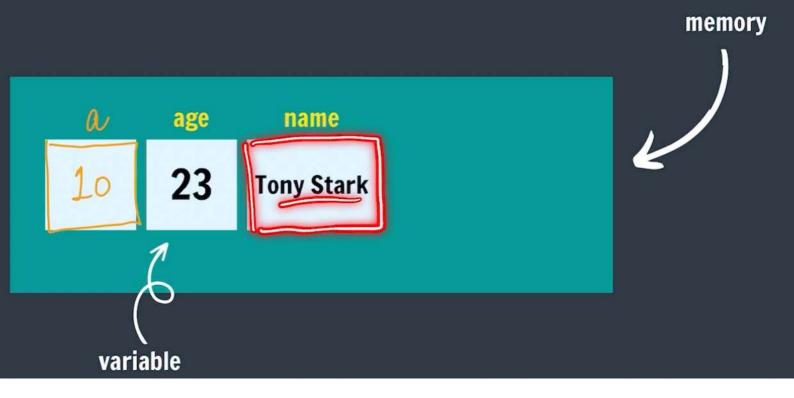
vas ral

name = "Tony

memory

### What is a Variable?

A variable is simply the name of a storage location.

















3 Issues: 3



> a = 10

· 10

> b = 5

< 5 €

> age = 23

√ 23

> name = "tony"

'tony'

> a

< 10 B

## **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 + bsaetsagupta73 a - \ Uncaught Syn a <· 5
  - > a \* b
  - < 50
  - > a / b
  - < 2
  - > 10 \* 2
  - <· 20

#### 2

### **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)

• Exponentiation (power operator)

## NaN in JS

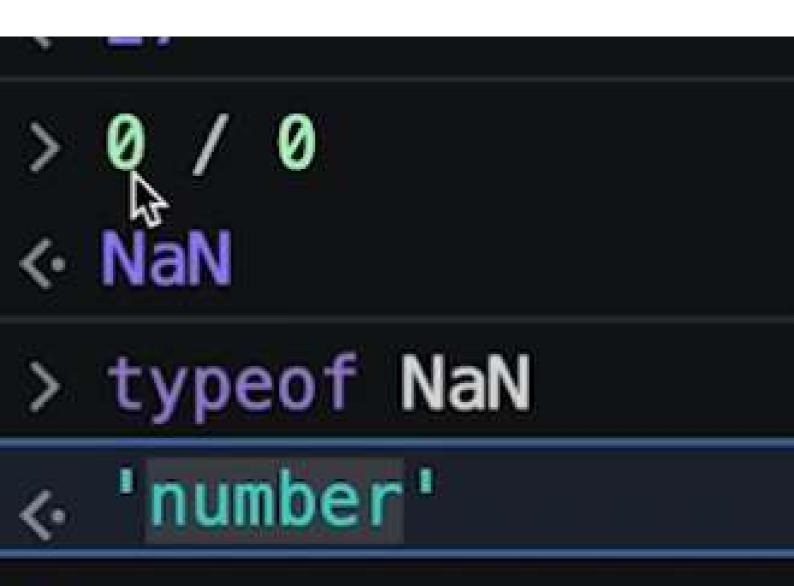
The NaN global property is a value representing Not-A-Number.

010

NaN - 1

NaN \* 1

NaN + NaN



- > typeof NaN
- 'number'
- > NaN + 1
- NaN
- > NaN 1
- < NaN
- > NaN \* 1
- < NaN
- > NaN \* NaN
- NaN

## **Operator Precendence**

This is the general order of solving an expression.

()

\*\*

\*, 1, %

+, -

## **Operator Precendence**

#### **Evaluate**

$$(2+1)*3$$

## let keyword

#### Syntax of declaring variables

```
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
```

#### **const** keyword

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

```
const year = 2025;
```

year = 2026

. . . . . .

**II Error** 

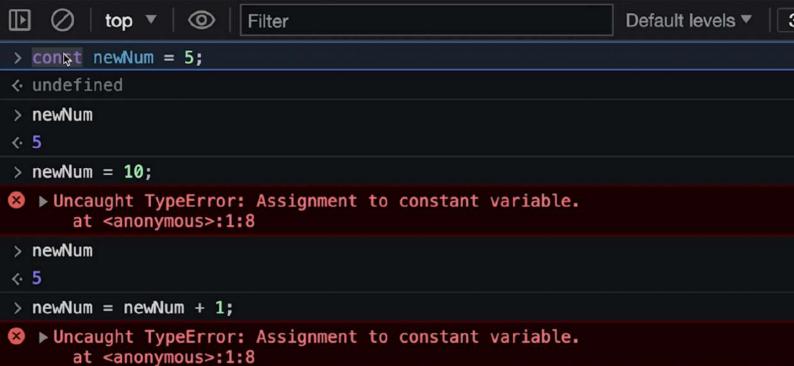
year = year + 1 // Error

**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
```



```
> 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</pre>
```

# var keyword

#### Old Syntax of writing variables

```
var age = 23;
```

```
var cgpa = 8.9;
```

```
var num1 = 1;
```

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

- > var x = 5;
- undefined
- > var y = 10;
- undefined
- > var sum = x + y;
- undefined
- > sum
- <. 15 ₽

# **Assignment Operators**

# Unary Operators

age = age + 1

age += 1

age++ // increment

age = age - 1

age -= 1

age-- II decrement

binary operator upta738500 operands

ط عد ۵

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) snehagupta738(pre-zite)
a++ (fost-inc) a-- (post-del)

#### **Identifier Rules**

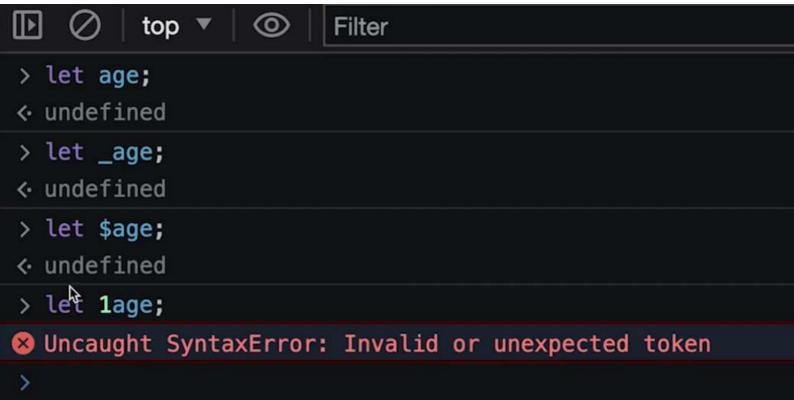
All JavaScript variables must be identified with unique names (identifiers).

- 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_;
< undefined
> let price_;
< undefined
> let price-;

 Uncaught SyntaxError: Unexpected token '-'
> let old price;

Uncaught SyntaxError: Unexpected identifier 'price'
```



## camelCase

Way of writing identifiers

cameCase (JS naming convention)

snake\_case

**PascalCase** 

### camelCase

Way of writing identifiers

came Case (IS naming convention) camel Case let full Jame;

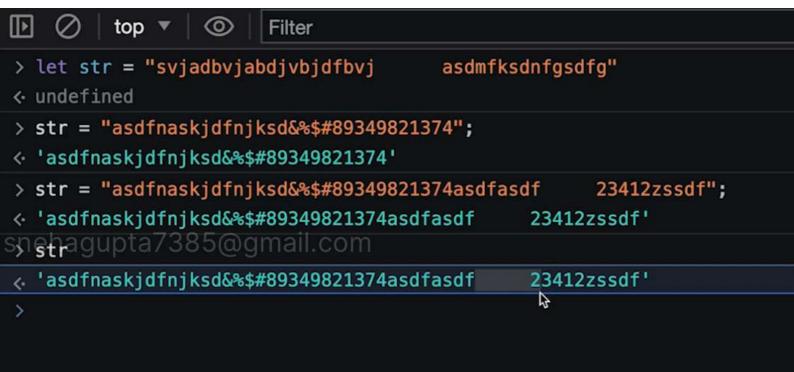
snake\_case

**PascalCase** 

### **Boolean in JS**

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

```
let age = 23 ;
let isAdult = true ;
let age = 13 ;
let isAdult = false ;
```



## **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?

```
> "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'

>
```



### **String Indices**

let name = "TONY STARK";



name[0] -> 'T'

name[1] -> '0' ...

### null & undefined in JS

#### undefined

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

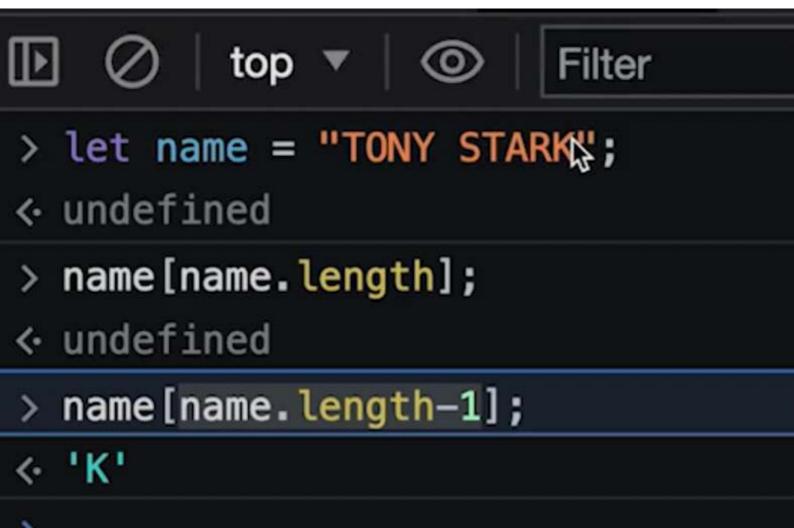
```
> let a;
< undefined
> a
< undefined
```

#### null

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

To be explicitly assigned.

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



# What is TypeScript?

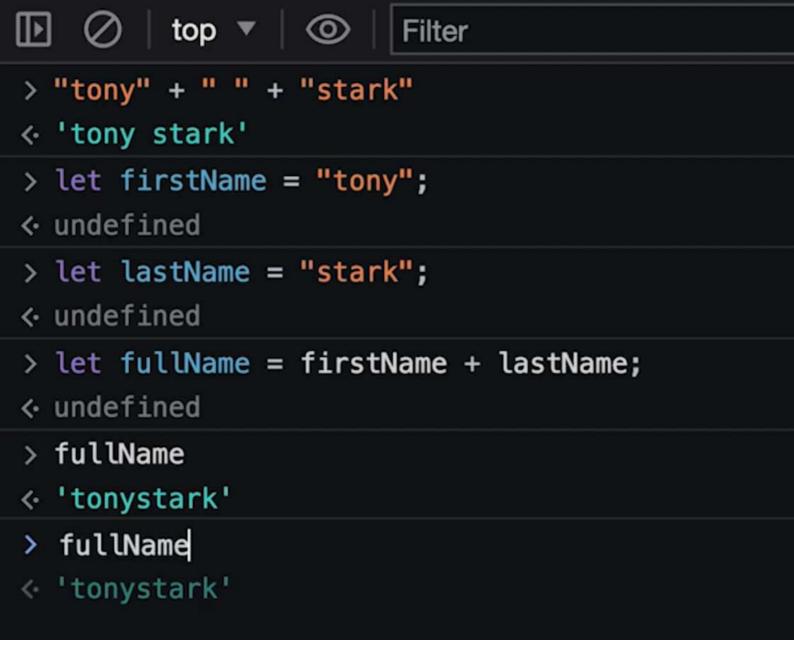
Static Typed, where JS is dynamic typed

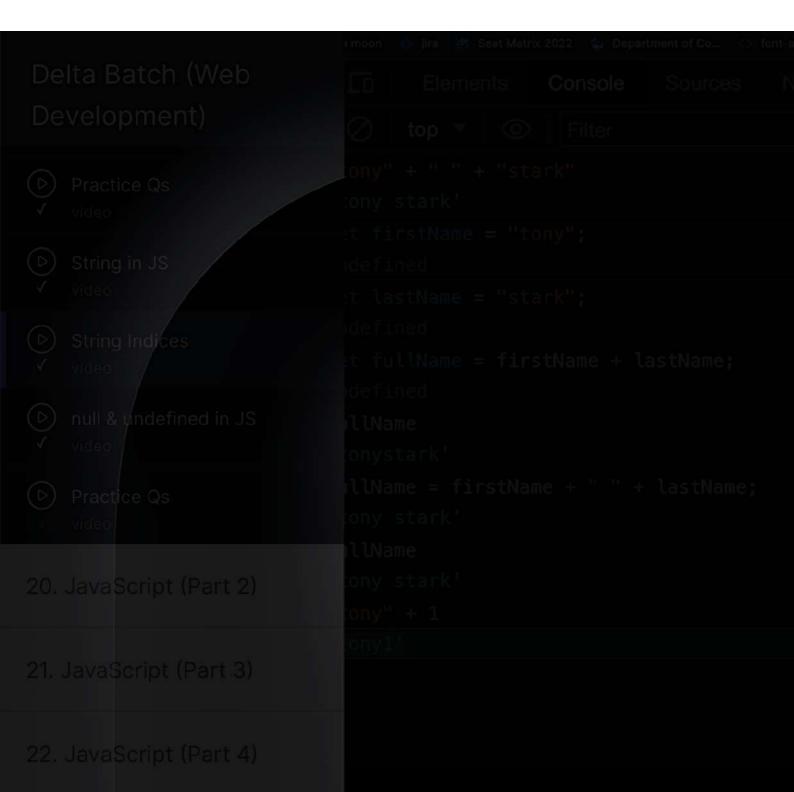
**Designed by Microsoft** 



# Concatenation

adding strings together



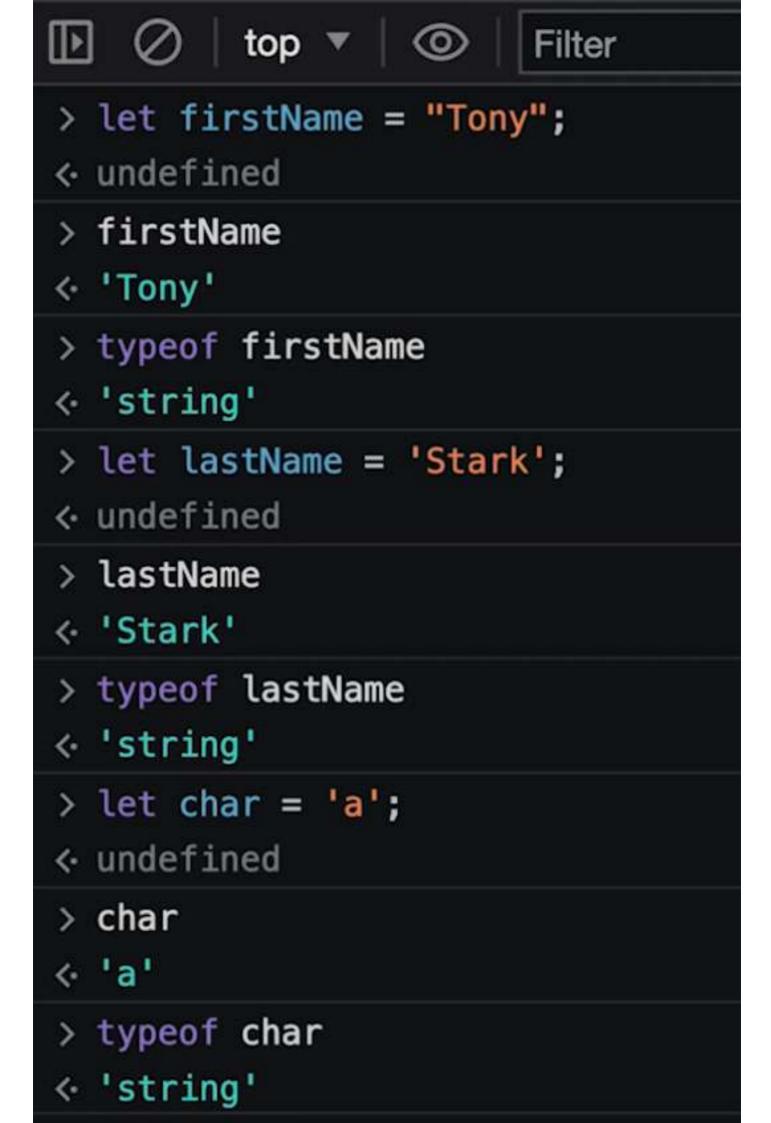


## Practice Qs

Qs. Find the errors in the following code?

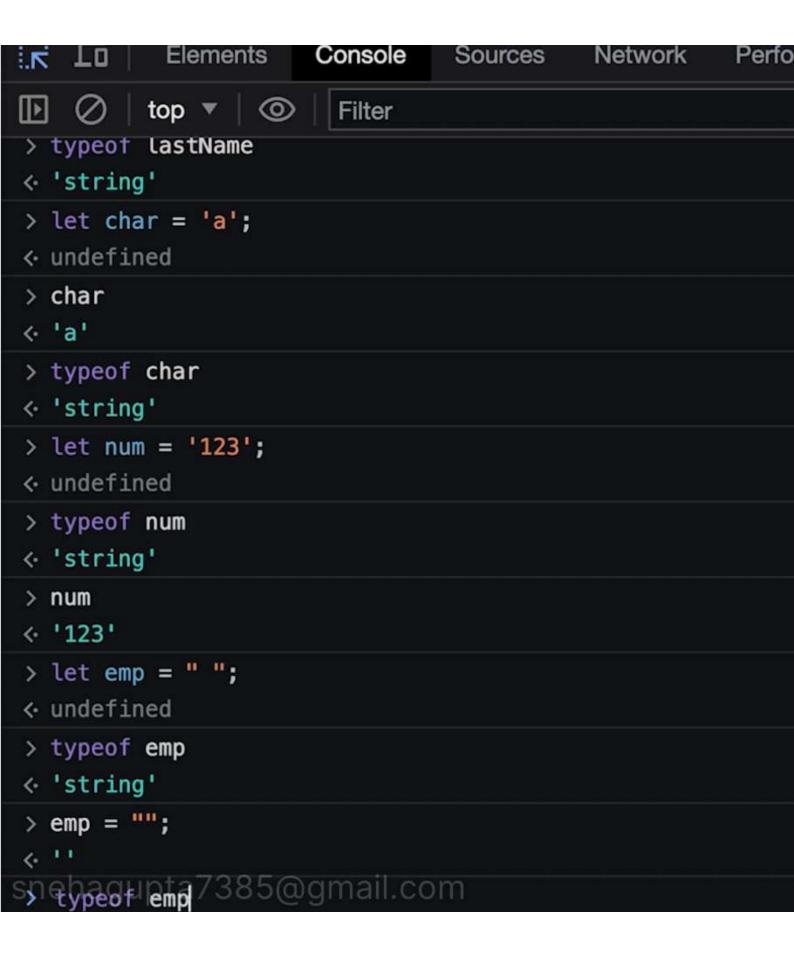
```
let marks = 75;
let isPass = True;
```

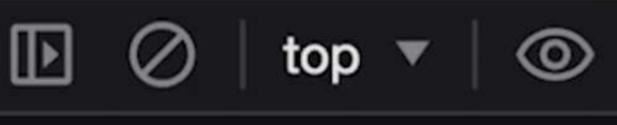
```
c) let isPass = 'true';
```



TS

Shellagupta/383@9 a = 5 a = 5 a = t a = 10 a > 10





**Filter** 

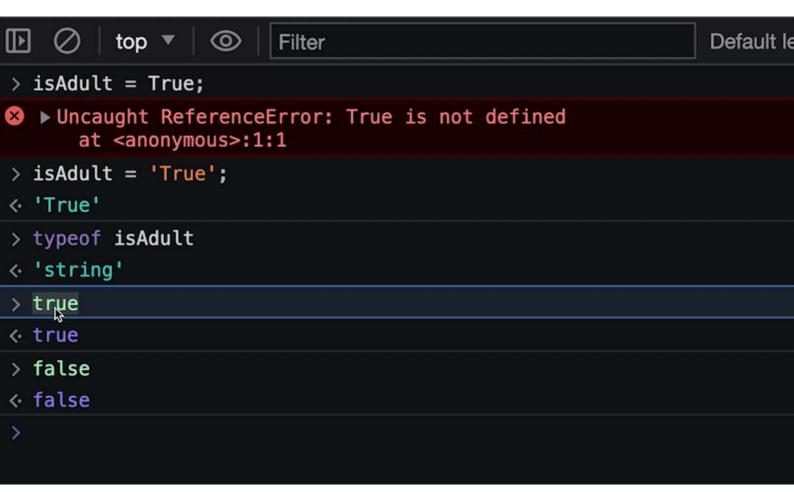
- > "shradha".length
- √ 7
- > "shradha" [6]
- < 'a'
- > "shradha"[0]
- <- 's'
- >

```
name[1]
  '0'
  name[2]
  'N'
 name[3]
  name[4]
 name[1000]

← undefined

  name.length
• 10
  typeof name.length
  'number'
  name[name.length];

← undefined
```



```
let name;
undefined
  name

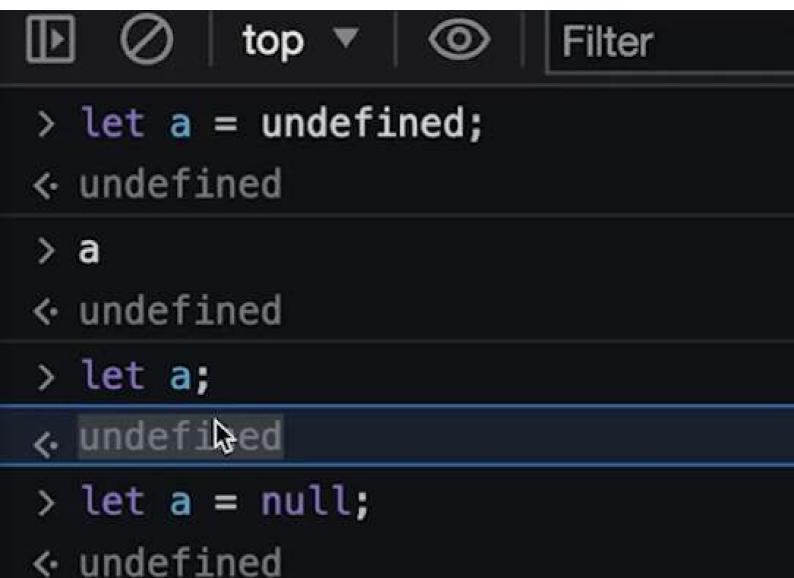
    undefined

> typeof name
'undefined'
> name = "apnacollege";
'apnacollege'
> name[1000];

    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?