Bootstrap

1. TO ADD BOOTSTRAP WITH HTML ➔ head e link set kore webside theke href niye aste hobe
   body te script add kore src te website hote link eke paste

2. COLOR ➔ primary, secondary, success, info, danger, warning, dark, light, black
   muted (text er opacity halka komay), transparent, gradient

3. OPACITY ➔ In CLASS → bg-opacity-10/25/50/75/100 (bg/border/text)
   In STYLE → --bs-bg-opacity: .5/50%; (point/percentage)

4. CENTERING ➔ class="w-50 mx-auto" → div/section center korar jonno width set kore mx-auto dibo
   class="h-100 my-auto → container er height age set kore then tar div e eita
   class="text-center" → text er design center diye sob kichu center

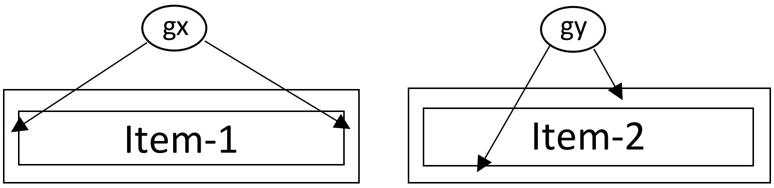
5. HEIGHT/WIDTH ➔ h/w-0/25/50/75/100 , mh/mw(min h. w.) , vh/vw(viewport h. w.)

   height set korte hole jar height set korbo take ekta container er maje rekhe setar height age set kore then child er hright property dile kaj kore thake

6. SPACING Margin/padding → m/p-0/1/2/3/4/5 (t,b,s,e,x,y) mt,pt,mb……

   Gap-1/2/3/4/5 → for grid gap

   g-1/2/3/4/5 → Gutters for grid gap gx/gy-1,2,3,4,5 (hori./ver.)



7. BORDER ➔

border	to add a border
border-top/end/bottom/start	" " " " on a particular side
border-top/end/bottom/start-0	to remove border
border-primary/……	to add border color
border-1/2/3/4/5	to set border width
border-rounded-0/1/2/3/4/5	to set border-radius
rounded-start/end/bottom/top	
rounded-circle/pill	

border e age border likhe border add kote then border-0/1/2/3/4/5 likhe sizing korte hoy.

8. VISIBILITY ➔ visible / invisible

9. TEXT/FONT →

text-start//end/center
text-wrap/nowrap
text-uppercase/lowercase/capitalize
text-break
text-decoration-none/underline/line-through
user-select-none
fs-1/2/3/4/5/6 font size 1 e boro, 6 e ekebare chuto
fw-light/normal/semibold/bold
fst-italic
lh-1/sm/base/lg

10. BORDER_RADIUS →

rounded-0/1/2/3/4/5
rounded-top/bottom/end/start/pill/circle

11. TYPOGRAPHY →

ALL CLASS ITEMS

h1/h2/h3/h4/h5/h6	p tag er mode use kore heading convert
display-1/2/3/4/5/6	display r size chuto boro kora jay
lead	onekgula p tag er mode ekjonke leader class dile se leader hisebe halka size e boro hoy
blockquote	text blockquote er style hoy
blockquote-footer	footer e kisu add kore jemon writter er nam

```
<figure>          <blockquote class="blockquote">    <p>A well-known quote,
contained in a blockquote element.</p>           </blockquote>
 <figcaption class="blockquote-footer">  Someone famous in
<cite title="Source Title">Source Title</cite>      </figcaption>        </figure>
```

12. POSITION →

position-static/relative/absolute/sticky/fixed
top/bottom/end/start-0/50/100
for centering → translate-middle-x, translate-middle, translate-middle-y

----See Bootstrap Website

13. VISIBILITY →

visible/invisible

14. BOX-SHADOW →

shadow / shadow-sm / shadow-lg / shadow-none

15. DISPLAY →

d-none/inline/inline-block/block/grid/flex

16. OVERFLOW →

overflow-auto/visible/scroll/hidden

17. BUTTON →

All Class Items

btn	auto button er ekta design dey
btn-primary/secondar…	button er color set
btn-close	cross er moto ekta icon/button hoy
btn-lg/sm	button er size halka edik sedik arki
disabled	button disable kore dey

18. LIST ➔

list-unstyled	list er default style remove kore dey
list-inline	ul e use kora hoy
list-inline-item	li e use kora hoy. Ei 2 ta eksathe use kore list je inline kora jay

19. ALART ➔

alart alart-primary….	Alart ekta create kore eke modify kora

20. TABLE ➔

class in table ==>

table	auto halka design kore ney
table-primary/dark..	table er color set
table-striped	zebra design in row
table-striped-columns	zebra design in columns
table-hover	hover korle ekta effect dey
table-bordered	ekta border hoy table er
border-danger/primary..	oi border er color set kora
table-borderless	table er border remove
table-sm	table er size chuto kore
caption-top	normally caption dile ta table er niche take . eita use korle upore ashe

class in tr,th,td ==>

table-active	eita use korle oi tr,th,td different color dekay
table-group-divider	group kke devide kore . normally tbody/thead e use kora hoy jate onno group theke devide korte pari
table-primary/dark..	particularly oi tr,th,tf color set kora hoy

21. IMAGE ➔

img-fluid	for responsiveness
img-thumbnail	4 side e ekta border style hoye jay passport picture er moto
w-0/25/50/75/100	
float-start/end	image left/right korar jonno
mx-auto d-block	age jevabe width,margin:0 auto diye center kortam tik emoni otoba ekta div er under e image ke niye div er class e text-center dibo

22. FLEX ➔

Parent property →

d-flex

d-sm/md/lg/xl/xxl-flex

flex-column/column-reverse/row/row- reverse

justify-content-start/center/end/between/around/evenly

align-items-start-center-end-stretch/baseline

d-inline-flex

d-sm/md/lg/xl/xxl-inline-flex

flex-sm/md/lg/xl/xxl-column/column-reverse/row/row- reverse

justify-content-sm/md/lg/xl/xxl-center

align-item-sm/md/lg/xl/xxl-center

align-self-start/center/end/baseline/stretch

Child Property →

flex-fill	flex er baki sob jayga fillup kore ney
flex-sm/md/lg/xl/xxl-fill	agertar sathe ekta size add kore
flex-grow-1	
flex-sm/md/lg/xl/xxl-grow-1	oi child oportike chuto kore nije boro hoye jay
flex-shrink-1	
flex-sm/md/lg/xl/xxl-shrink-1	oi child nije chuto hoye onnoderke boro kore dey
me-auto	
ms/me/mt/mb-auto	oi child er end er dike sob kichute auto margin hoye dure chole jabe
flex-wrap/nowrap/wrap-reverse	
order-0/1/2/3/4/5/6	child er order set kore position er jonno

23. GRID ➔

d-grid container>row>col It’s follow the 12 column meathod

gap

mini size	small size	medium size
col-12 (full space of a row)	col-sm-6 (half of a row)	col-md-4 (1/3 of a row)
large size	xl size	xxl size
col-lg-3 (1/4 of a row)	col-xl-2 (1/6 of a row)	col-xxl-1 (1/12 of a row)

- Row e customize kora → class="row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 row-cols-xl-5 row-cols-xxl-6".

  Eketre 1,2,3,4 means prottek row te 1,2,3,4 ta colume thakbe.
  Such as :: row-cols-xxl-6 mane xxl size er container e 6 ta column thakbe.

- Every Col e same customize → class=”col-12 col-sm-6 col-md-4 col-xl-2 col-xxl-1”

  Eketre 12,6,4,3,2,1 means 12 ta space er maje kototir jayga dokol korbe ta.
  Such as :: col-xxl-1 means xxl size er container e per column 1 ta space nibe.
  col-lg-3 means lg size er per column 3 ta space nibe ebong column sonka (12/3=4)

TailWind

1. To use it we need
   - Docs > play cdn > Try customizing your config > COPY BOTH SCRIPT and paste it on head part
   - **npx tailwindcss init** → jodi na snippets na ase tahole terminal e eita run korate hobe

2. COLORS → black white transparent current

   state
   gray
   zinc
   neutral

   red
   orange
   amber
   yellow

   lime
   green
   emerald
   teal

   cyan
   sky
   blue
   indigo

   violet
   purple
   fuchsia
   pink
   rose

   value → (100-900)

   bg-/text-/decoration-/ border-

3. OPACITY → opacity-0/5/10/20/25/30/40/50/60/70/75/80/90/95/100
   2 meathod e use kora hoy →
   - Upore jevabe leka hoyeche sevabe . Ei meathod e full background er opacity control kore thake.
   - Jodi particular kunu text/bg/…… er opacity control korte chai tahole tader color er sathe include kore dite hoy.
     Such As. bg-red-800/80
     text-orange-600/50

     eigulay opacity-r value diye control kora hoy.

   Values → 0 , 0.5 , 1 , 1.5 , 2 , 2.5 , 3 , 3.5 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 12 , 14 , 16
   20 , 24 , 28 , 32 , 36 , 40 , 44 , 48 , 52 , 56 , 60 , 64 , 72 , 80 , 96

4. PADDING → p/p-x-value/p-y-value/p-t-value/p-b-value/p-l-value/p-r-value
   MARGIN → m/m-x-value/m-y-value/m-t-value/m-b-value/m-l-value/m-r-value
   For negative margin/padding → (-) add before. Such as. (-m-x-5)

5. SPACE-BETWEEN → space-x/y-value → 2 ta container x/y axes borabor pasapasi takle tader moddokar gap dite chaile use it
   space-x/y-reverse → jodi container gulake reverse korte chai tahole eita.

6. CENTERING →
   i. height diye my-auto
   ii. width diye mx-auto
   iii. container mx-auto
   iv.

TYPOGRAPHY

7. FONE-SIZE ➔ text-xs /sm /base /xl /2xl /3xl /4xl /5xl /6xl /7xl /8xl /9xl
8. FONT_FAMILY ➔ font-sans/serif/mono
9. FONT_STYLE ➔ italic / non-italic
10. FONT-WEIGHT ➔ font-thin/ extralight/ light/ normal/ medium/ semibold/ bold/ extrabold/ black

    100 200 300 400 500 600 700 800 900

11. LETTER-SPACING ➔ tracking-tighter / tight / normal / wide / wider / widest
12. LINE-HEIGHT ➔ leading-3 / 4 / 5 / 6 / 7 / 8 / 9 / 10
13. LIST-STYLE ➔ list-none/disc/decimal
14. LIST-STYLE-POSITION ➔ list-inside / outside

    Inside dei tahole list gula disc hole porer line disc er niche jay . jodi outside dei tahole porer line dics er niche na ese halka sore ase. Full ek line e take.

15. TEXT-ALIGN ➔ text-center/justify/left/right/start/end
16. TEXT-COLOR ➔ text-color-opacity
17. TEXT-DECORATION ➔ underline/overline.line-through
18. TEXT-DECORATION-COLOR ➔ decoration-color
19. TEXT-DECORATION-STYLE ➔ decoration-solid/dotted/dashed/double/wavy
20. TEXT-DECORATION-THICKNESS ➔ decoration-0/1/2/4/8
21. TEXT-UNDERLINE-OFFSET ➔ underline-offset-0/1/2/4/8 text er nicher underline er durotto
22. TEXT-TRANSFORM ➔ uppercase , lowercase , capitalize , normal-case
23. TEXT-OVERFLOW ➔ text-ellipsis , text-clip , truncate
24. TEXT-INDENT ➔ indent-value text er age padding add korar moto
    Kaj kore take indent
25. VERTICAL-ALIGN ➔ align-top/middle/bottom/text-top/text-bottom/sub/super
26. WHITESPACE ➔ whitespace-normal/nowrap/pre/pre-line/pre-wrap

BORDER

27. BORDER-WIDTH ➔ border-0/2/4/8 , border-x/y/t/b/l/r-0/2/4/8
28. BORDER-RADIUS ➔ rounded , rounded-sm/md/lg/xl/2xl/3xl rounded-x/y/t/b/l/r-……
29. BORDER-COLOR ➔ border-color , border-x/y/t/b/l/r-color
30. BORDER-STYLE ➔ border-solid/dotted/dashed/double/hidden

31. OUTLINE ➔ outline-width → outline-0/1/2/4/8
    Outline-color → outline-colorw

32. **TO ADD CUSTOM COLOR IN BOOTSTRAP ➔**
    - Fisrt check that is the color is available or not tailwind customizing color or not . if available just copy the color name and paste
    - If color are not available there are 2 meathod .
        - ○ Use bg-[#hex_code] in class. But it is not recommended . it can be use if the color are used only one times in the website
        - ○ Use any color name and hex code in script>tailwind.config>extend>colors.
          Example: mahi: ‘#fffbf0’;

FLEX

33. INITIALIZATION : → flex
34. FLEX-DIRECTION: → flex-column/column-reverse/row/row-reverse
35. FLEX-WRAP: → flex-wrap/wrap-reverse/nowrap

CHILD-PROPERTY

36. FLEX-GROW: → grow
37. FLEX-SHRINK: → shrink
38. ORDER: → order-1,2,3,4……..firse,last
39. FLEX-BASIS: → basis- ½ , 1/3 , ¼ ……..

GRID

40. INITIALIZATION : → grid
41. GRID-TEMPLATE-COLUMNS: → grid-cols-1/2/3/4/5/6/7/8/9/10/11/12
42. GRID-TEMPLATE-ROWS: → grid-rows-1/2/3/4/5/6
43. COL-SPAN / ROW-SPAN: → col-span-1/2/3/4/5/6 row-span-1/2/3/4/5/6
44. GAP → gap-value / gap-x-value / gap-y-value
45. JUSTIFY-CONTENT → justify-center/between/evenly/around/start/end/stretch
46. ALIGN-ITEMS → items-center/start/end/stretch
47.

FOR RESPONSIVE

48. For anything in any size

	sm:	md:	lg:	xl:	2xl:
grid-cols-1	sm:grid-cols-2	sm:grid-cols-4	lg:grid-cols-6	xl:grid-cols-8	2xl:grid-cols-12
text-xs	sm:text-base	md:text-lg	lg:text-xl	xl:text-3xl	2xl:text-9xl
text-yellow-100	sm:text-red-300		eivabe solor o set kora jay ekektat jonno ekekta		
p-5	sm:p-10	md:p-30	eivabe padding margin o set kora jay		

tailwind e normally kunu size ba kichu dile ta sob kichur jonno apply hoye jay automatically. Tai amra responsive korar jonno *sm: md: lg: xl: 2xl:* use kora hoye thake.

grid-cols-1	sm:grid-cols-2	mini device e 1 ta colume takle o small e ese er por theke 2 ta colume hoye jabe
text-xs	lg:text-xl	large device thelke xl text apply hobe. Er ager gulay xs font applt hobe.

Eivabe tailwind e easily jekunu kichu responsive kora jay

JAVASCRIPT

1. JavaScript (ES6) code snippets   Extension For JavaScript
2. Converting string into integer   parseInt();   numeric string hole kaj korbe only
   Converting string into float   parseFloat();
3. In js compare with string and int/float   if(‘15’==15)   true
   Are same .   if(’15.15’==15.15)   true
4. Onek somoy pointing value onek lomba diye thake . Eke jodi amra 2 got ba 3 gor porjonto dite chai tahole tofixed(x) name e ekta input dibo jekane x hole joto gor porjonto jabe ta. Tobe eita je value return kore ta string hoye jay means tofixed string value return kore take jake abar amader parseFloat use kore float e convert kore nite hoy. Example:

```
var number1 = 0.1;
var number2 = 0.2;
var sum = number1 + number2;
sum = sum.toFixed(3);          dosomiker por 3 for porjonto jabe but integer
                               string e convert kore fele
sum = parseFloat(sum);         parseFloat use er maddome amora abar float
                               value ante perechi
console.log(sum);
```

5. For string sum in javascript

```
firstName=’mahi’;
lastName=’mashud’;
fullname=firstName+’ ‘+lastName;
```

   jodi amra maje space ne dei tahole 2 ota name eksathe chole asbe

6. Some unique variable ➔
   - Camel-Case → someExample   2nd word theke 1st letter uprcse
   - Pascal-Case → SomeExample   1st word theke 1st letter uprcse
   - Sanke-Case → some_example   _use ko ra space er jaygay
   - (-) → It can’t use in js. You can use (_)

7. To check na variable type use   typeof()

```
var mahi=”mashud”;
console.log(typeof(mahi));
```

8. toLowerCase() ➔   it’s used for lowercase

```
var a=”MASHUD”;
a=a.toLowerCase();
console.log(a);
```

9. toUpperCase() ➔   it’s used for uppercase

```
var a=”mahi”;
a=a.toLowerCase();
console.log(a);
```

ARRAY

	var arr=[1,2,3,4,5], arr1=[2,3];	
10. Define Blank array in js	let arr=[];	
11. To check array or not	Array.isArray(arr)	array hole true, nahole false
12. Get elements value by index	console.log(arr[0/1/2/3/4]);	
13. Set Elements value by index	arr[2]=10;	
14. Find index of an item	console.log(arr.indexof(3));	item array na -1 return kore
15. Array Length	console.log(arr.length);	
16. Array Concat	arr.concat(arr1)	arr named array r sathe arr1 add kore theke
17. Array PUSH	arr.push(10);	a er seshe item add hoy
	arr.push(20,30,40);	ekadik items add
18. Array POP	arr.pop()	laster item remove
	var lastItem=arr.pop()	last item remove hoy kintu onno var e ei meathod e copy kora jay
19. Array UNSHIFT	arr.unshift()	1st e kunu item add kore
20. Array SHIFT	arr.shift()	1st item remove kore
21. Array SLICE	arr.slice(2,5)	**2nd index hote suru kore 4th index porjonto ney Main array change hoy na**
22. Array SPLICE	arr.splice(2,5)	**2nd index hote suru kore arr theke 5 ta items nibe.**
	Arr.splice(2,5,1,2,3,4)	**2nd index hote suru kore 5 ta Items kete niye oi jaygay 1,2,3,4 add kore dey.**
23. INCLUDES	arr.includes(2)	jodi array/variable e ei item Thake tahole trui, noile false.
24. FILTER	arr.filter(Number/String/Boolean)	Array r mode empty items takle tader remove kore array ke show kora thake.

25. Condition using Includes and indexof

• if (arr.indexOf(10) != -1)	jodi arr namok array e 10 available
• if (arr.includes(10) == true)	take ei condition gulute dukbe

STRING ➔ Immutable

let mahi=”Mashudur Rahman mahi”;
let mahi1=’is a good student’;

26. Substring mahi.substring(2,10) 2nd index hote 10th Porjonto nibe
Slice dile o substring er same kaj kore.
27. Slice mahi.slice(2,10);
28. Get index value dekabe mahi[2] cl e dile 2nd index e ja ache ta
But array r moto index number diye index er jinis change kora jayna
29. Get length mahi.length
30. IndexOf mahi.indexof(‘Rah’) index jekan theke suru hoise ta
Dekabe . na takle -1 return korbe
31. Includes mahi.includes(‘rahman’) mahi namok string e jodi rahman
Thake tahole true, naile false return
Mahi.toLowerCase().includes(mahi1.toLowerCase())

Amra normally kunu kichu milanur jonno sobkichuke lowercase/uppercase e convert kore then includes diye check korbo . mille true return korbe ar na mille false return Korbe

32. StartsWith & EndsWith mahi.startsWith(‘Mas’); correct hole true return korbe
Mahi.endsWith(‘.js’); false hole false return korbe

33. String Sum console.log(mahi+’ ‘+mahi1)
34. Concat console.log(mahi.concat(‘ ‘ , mahi1);

Some Condition For String

- if (lyrics.indexOf('sada') != -1) jodi lyrics string e sada available take
- if (lyrics.includes('kala') == true) ei condition gulute dukbe

35. SPLIT let lyrics=’tumi bondhu, sada pakhi , ami jeno ki’;
    - lyrics.split(‘’); lyrics er per character er Array ShowUp
    - lyrics.split(‘,’); lyrics e joto jaygay comma(,) ache er por theke Array ShowUp
    - lyrics.split(‘ ‘); joto jaygar space ache tader ke aladavabe Array ShowUp

SPLIT array type value return kore thake

split(x) → ekane x e je value dibo ta jodi string e take tahole er ager jinis gula array ek index er bitore takbe ebong er porer jinisgula array e porer index e chole jabe. Eivabe string kaj kore take . ar jodi kunu kichu na dei means only quote dei tahole prottekta single character niye array toiri hobe

36. TRIM let mahi1=”   mahi”, mahi2=”mahi   “;
    - mahi1.trimStart() surute space takle ta remove
    - mahi2..trimEnd() seshe space takle ta remove
    - mahi1.trim() suru ebong sese space takle sob remove
37. JOIN let mahi=[‘tumi bondhu’,’kala pakhi’,’ami jeno ki’]
    - mahi.join() array string er sob item ke join kore ney
    - mahi.join(‘ x ’) ekane x e je value dibo ta join er jaygay bosbe

      ANS. tumi bondhu x kala pakhi x ami jeno ki

SPLIT ➔ String ke array te convert kore thake part part kore .

JOIN ➔ Array theke String e convert kore thake

FUNCTION

38. To declare a function

    function function_name(1/more parameter){
    ……arguments……
    return (any) ;

OBJECTS

- Object is a variable that contains different types of huge value

39. To declare a object

    var object_name = { properties : value, properties : value, properties : value }   //
    var student = {
    id: 2020331513,
    name: 'Mashudur Rahman Mahi',
    year: 11,
    marks: 10}
40. To Declare Multiple Object:

    let obj1= [{………………}, {……………..}, {……………}]

41. To see the value of the properties console.log() er bitore nicher sob takbe dore nibo
    - student.name
    - student['name']
    - var mahi='name'; student[mahi];

42. To see all the properties of the object ➔ Object.keys(student)
43. To see all the values of the object ➔ Object.values(student)

NOTE: Object.keys/Object.values array type value return kore thake

44. To set the Property Values
    - student.name='MASHUD';
    - student['name']='MASHUD';
    - var mahi='name'; student[mahi]= 'MASHUD';

45. to delete a object items delete object_name**.**object_items_name

46. In javascript you can add object,array,function in the object. We cann't use loop in directly in javascript object . we can use it in the function in object . if we use function in a object we can call the function by the object name and last side we should use ()

```
let student = {
   name: 'Mashudur Rahma',        id: '2020331513',                 address: 'silverVillageResidentialArea',
   issingle: true,             friends: ['mahi', 'rahi', 'anis'],        movies: [{ name: 'no-1 shakib khan', year: 2012 }],
   car: {   brand: 'tesla',   price: '$56000',      model: '2022',
                   manufacturar: {  name: 'tesla',            cse: 'elon Mask',     country: 'USA' } },
   act: function () {
      console.log("Mashudur Rahman Mahi");  }  };
```

student.act(); ➔ To call function we should use () adter object property_name

47. Swapping Use Temp

    Use ARRAY **[a,b] = [b,a]**

48. **MATH FUNCTION**
    - Math.PI Math.E
    - Math.sin/cos/tan();
    - Math.round/trunc/ceil/floor();
    - Math.pow(x,y);
    - Math.abs();
    - Math.min(1,2,3,4,5,6); majkane joto item takbe tader
    - Math.max(1,2,3,4,5,6); maje theke min/max output dibe.

49. To Declare Black Array and String

- let myArr=[ ] , myStr=“ ” ;

To add new items on it

- myArr.push(any_thing);
- myStr=myStr+” ”+any_thing
- myStr=myStr.concat(“ ”, any_thing);

50. Datatypes in JS ➔

- Primitive Datatypes → Numeric, String, Boolean
- Trivial.Other Datatypes → Null, Undefined
- Non-Primitive Datatypes → Object, Arrays,

ARGUMENTS ➔

51. It’s an array like object, not fully array. It can be only use into the function. By using it we can found all the value that user gives to us. Example.

```
function name(a,b){
console.log(a,b);
        console.log(arguments/arguments[4]);}
name(1,2,3,4,5,6);
```

normally amra function er bitore 1,2 e kebol a,b er maje pabo kintu arguments ude korar maddome baki upadan guluke o pete pari.

Times In JS

const time = new Date();

1. console.log(time); = = showing current time
2. console.log(time.getDay/…….) = particular;y day hour minute…… show kore
3. time.setDay(21)/.setFullYear(2032) = new day/year ta set kore thake
4. new Date(years,months,days,hours,minutes,seconds,miliseconds) eivabe man diye custom date run
   Ex. const timing1=new Date(2020,10,30,23,59,10,1000); console.log(timing1);
5. new Date(‘year-months-days’) eivabe o custom date set kora jay
   Ex. const timing2 = new Date(‘2020-12-31”) console.log(timing2);

SOME CODE

1.

```
function wordReverse(text) {
let arr1 = [ ];
let words = text.split(" ");
console.log(words);
for (let i = words.length - 1; i >= 0; i--) {
        arr1.push(words[i]);
}
console.log(arr1);
text = arr1.join(' ');
return text;
}

let myStr = "My name is mashudur rahman mahi";
console.log(wordReverse(myStr));
```

2. Searching items in an array type of object

```
const products = [
        { name: "Oneplus", price: 37000, model: 'OnePlus 9r' },
        { name: "Samsung", price: 32000, model: 'Samsung M51' },
         { name: "Xiomi", price: 34000, model: 'Redmi K30 Ultra' }   ]
function isAvailable(products, search) {
        let items = [];
        for (product of products) {
        if (product.name.toLowerCase().includes(search.toLowerCase()))
                items.push(product); }
        return items;  }
console.log(isAvailable(products, 'ONEPLUS'));
```

JAVASCRIPT + HTML + CSS

DOM MANUPULATION

1. JavaScript is a High Level, Interpreted Language.
2. To connect JS with html ,
    - For Internal script use script tag after all the body element

      Ex- `<script>   ………….</script>`
    - For Externam script use it on same position and in scr give the JS file link.   Ex.   `<script src=”JavaScript.js”>………….</script>`

3. From Html we can get elemets with 5 meathods
    - **getElementsByTagName , getElementsbyClassName , getElementById querySelectorAll , quarySelector**

4. JavaScript use kore html theke ekoi tag name er sokon upadanke ekta array like object er maje anar jonno **document.getElementsbyTagName(“tag_name”);** use kora hoye thake.

    Ex.
    ```
    let licollection=document.getElementsByTagName(‘li’);
    for(let li of licollection){            console.log(li);    console.log(li.innerText);  }
    ```

    cl.li use er maddome je array like object amora paisi tar upadan show korbe

    cl.li.innerText use er maddome li er modder text ke show korbe

5. To get class in js we can use **document.getElementsByClassName(“class_name”)**

    Ex.
    ```
    const places = document.getElementsByClassName(“place”);
    for(let place of places){          console.log(“place”):         console.log(place.innerText”);  }
    ```

6. To get ID in js we can use **document.getElementById(“id_name”);**

    Ex.
    ```
    const fruit_title = document.getElementById(“fruits_title”);
    console.log(fruits_title);                        console.log(fruits_title.innerText”);
    fruits_title.innerText=”This title is changer by JavaScript”;
    ```

    Eivabe amra id r maje ja leka ase ta change korte pari by using innerText

7. Someimes we need to use selector by css style . For this we should use (.) before class name and (#) before id name.

- **document.querySelectorAll(.class-name……)**

  Ex.

  ```
  const someLi = document.querySelectorAll('.fruits-container li');   container er bitorer id select
  console.log(someLi);
  for (let li of someLi) {    console.log(li);        console.log(li.innerText);  }
  ```

- **document.querySelector(.class-name……..)**

  Ex.

  ```
  const someLi = document.querySelector('.fruits-container li');
  console.log(someLi);              console.log(someLi.innerText);
  ```

  It's only take the first element of the inside li of the class

8. HTMLCollection vs NodeList

- Similarity →
  - i. Both are array like collection of elements, But not array
  - ii. Both have a length property that returns the number of   elements in the list
- Difference →
  - getElementsByClassName() / getElementsByTagName   → HTMLCollection

    getQuerySelectorAll()   →   NodeList

HtmlCollection	NodeList
Live collection	Static collection
for of	forEach, for of
Accessed their name/id/index number	Accessed only index number
HTMLCollection.length	NodeList.length
Support 2 meathods	Support 5 meathod
HTMLCollection.item()	NodeList.item()
HTMLCollection.nameditem()	Nodelist.forEach()
	NodeList.entries()
	Nodelist.keys()
	Nodelist.values()

- The childNodes property returns a live NodeList.

9. Kunu kichur index er item janar jonno   →   last e ['index_number'] use kore check

10. getAttribute() → er maddome amra getElementById/getElementsByClassName er dara songrihito class/id r upor kaj korte pari. Only id r upor kaj kore

```
document.getElementById("fruits-name").getAttribute("class")
```

```
const title=document.getElementById("fruits-title")
title.getAttribute("class")
```

Both are same, Id er sob class deka jay

For Show Class → classList works on only getElementbyId

title.classList	title er sob class list akare ase
title.classList['2']	classlist er only 2nd index show korbe
title.classList.remove('mahi')	mahi jodi class e take tahole eke class theke remove kore thake
title.classList.add('miku')	miku ke classlist e add kore thake

11. removeAttribute → `document.getElementsById('mahi').removeattribute('disabled');`

tag/id theke kunu attribute like title,disabled,placeholder.. remove

12. setAttribute → `document.getElementsById('mahi').setattribute('disabled',ture);`

ekekte attribute ses korar somoy sathe true dite hoy.

```
const title=document.getElementById("fruits-title")
title.setAttribute('title','this is a tooltip maked by js')
```

13. innerHtml → `document.getElementById("fruits-title").innerHtml`
14. innerText → `document.getElementById("fruits-title").innerText`
15. value → innertext er moto input tag er likha nite hole value use kora lage

document.getElementsBytagName('input')[0].value;

16. Html change by js → `document.getElementById("fruits-title").innerHtml="<h1>mahi</h1>"`
17. Css in JS → `document.getElementById("fruits-title").style` ei id er sob style dekhabe

CSS e amra style korar somoy style er nam jodi ekadik word e takto tahole tader majkhane (-) ekta diye prithok takto . kintu js e eisob word camelCase e thake .

Such as: border-radius → borderRadius , background-color → backgroundColor

Some exapme:

```
document.getElementById("fruits-title").style.backgroundColor="indianred";
document.getElementById("fruits-title").style.color="black";
document.getElementById("fruits-title").style.border="20px solid black";
document.getElementById("fruits-title").style.borderRadius="20px";
document.getElementById("fruits-title").style.display="flex";
document.getElementById("fruits-title").style.justifyContent="center";
document.getElementById("fruits-title").style.alignItems="center";
```

✓ innerText,innerHtml,getAttribute, setAttribute, style,ei sobkichu particular ekta jinis er jonno kaj kore thake tai amora jokhon **getElementById, querySelector use kori tokon sohojei eigula use kora jay** . Baki ketre arki getElementsByClassName, querySelectorAll, getElementsByTagName er ketre **index number/ for loop chaliye eder mode javascript use kora hoy**

18. childNode ➔

- .childNodes → ei id/class er bitorer sob node show kore
- .childNodes[n] → nth childNode show kore thake
- nextSibling → ei childnode er porer sibling ke show kore
- previousSibling → ei childNode er ager sibling ke show kore

19. parentNode ➔ ei id/class je parent er maje thake take shoe kore dey. Maximum highest parentNode document

EX.

```
let child = document.getElementById('places-container');
console.log(child.childNodes);
console.log(child.childNodes[3].childNodes[3].nextSibling.nextSibling.previousSibling.previousSibling.parentNode.parentNode.parentNode.parentNode.parentNode);
```

20. createElement() ➔ er maddome js er sohojugitay html er any tag buildUp kora jay. Ex.

```
document.createElement('h1');         document.createElement('div');
document.createElement('section');    document.createElement('p');
```

21. add text inside createElement ➔

```
let mahi = document.createElement('h1');
h1.innerText="Mashudur Rahman Mahi";
```

22. appendChild/append ➔

```
add.append/appendChild(li);
```

23. Push this html tag by js ➔

```
// where to add
const add = document.getElementById('places-list');
// what to be added
const li= document.createElement('li');
li.innerText="Mashudur Rahman Mahi";
// add the child in it's parentNode
add.appendchild(li);
```

Example:

```
const mainContainer = document.getElementById('main-container');

const section = document.createElement('section');
mainContainer.appendChild(section);

const h1 = document.createElement('h1');
h1.innerText = "My Full List";
section.appendChild(h1);

const ul = document.createElement('ul');
section.appendChild(ul);
const li0 = document.createElement('li');
li0.innerText = "Biriyani";
ul.appendChild(li0);
const li1 = document.createElement('li');
li1.innerText = "Burhani";
ul.appendChild(li1);
const li2 = document.createElement('li');
li2.innerText = "Kabab";
ul.appendChild(li2);
const li3 = document.createElement('li');
li3.innerText = "Dhodhi";
ul.appendChild(li3);
➔      body → section → ul → li*4
```

24. Add html using another Meathod ➔ section create kore tar maje innerHTML use korbo.

innerHTML use korle (` `) use korbo.

Then section ke body te append kore dibo

```
const mainContainer = document.getElementById('main-container');
const sectionDress = document.createElement('section');

sectionDress.innerHTML = `
<h1>My Dress Section</h1>
<ul>
    <li>Biriyani</li>
    <li>Burhani</li>
    <li>Kabab</li>
    <li>Dodhi</li>
</ul>
`
mainContainer.appendChild(sectionDress);
```

25. getElementsByTagName / querySelectorAll / getElementsByClassName ➔

eder niye js e kaj korte chaile index/for loop diye kaj korte hoy

```
const sections = document.querySelectorAll('section');

for (let section of sections) {
    section.style.border = "2px solid red";
    section.style.backgroundColor = "lightgray";
    section.style.fontSize = "30px";
    section.style.fontFamily = "cursive";
    section.style.borderRadius = '10px';
    section.style.padding = "20px";
    section.style.margin = "20px";
    section.style.textAlign = "center";
}

const sections1 = document.querySelectorAll('section ul');
for (let li of sections1) {
    li.style.listStyle = "none";
}
```

26. ONCLICK FUNCTION ➔

- Meathod 1 → direct tag e onclick bosiye functionname likhe take call diye kaj

  <button onclick=”functionName()”> Mahi </button>
  <script> function functionName() { ..Bitorer Kaj ...} </script>

- Meathod 2 → script tag e onclick use korar maddome tobe eketre 1st jaygay

  Function call kora jabe na.

  ```
  document.getElementById('event-listener').onclick = make;
      function make() {
        document.getElementById('handler-status').innerText =
        "MASHUDUR rahman";
      }
  ```

- Meathod 3 → use addEventListener

  ```
  document.getElementById('eventListener1').addEventListener('
  click', function () {
          document.getElementById('handler-status').innerText =
  "MASHUDUR RAHMAN";
      })
  ```

27. addEventListener 1st property te onekgulu function thake . 2nd property te function declare kora

hoye thake. some property lick click :: click, mouseenter, mouseout, mousemove, foucs, blue,

keydown, keypress, keyup

28. Make CommentBox

- Textarea/input hote .value er maddome text copy kora
- Text copy kore uporer jaygay append kora
- Textarear value abar ‘ ’ null set kore return kora

Ex.

```
    document.getElementById("submit").addEventListener('click', function () {

        // step-1 ➔ input er jonno value use kora lage
        let comments = document.getElementById('comment-box');
        let comment = comments.value;

        // step-2 ➔ value je tag e niye push
        const div = document.createElement('div');
        const p = document.createElement('p');
        div.appendChild(p);
        p.innerText = comment;
        document.getElementById('comments').appendChild(div);

        // step-3
        comments.value = '';
    })
```

29. removeAttribute(‘x’) ➔ je attribute remove korbo x e tar nam dibo

30. setAttribute(‘x’,true) ➔ je attribute add korbo ta x e diye tar value true must.

31. Button Disabled-Enabled after typing delete:<button id= “button-type” disabled>Submit</button>

```
  `document.getElementById('delete-type').addEventListener('keyup', function () {

       let text = event.target.value.toLowerCase();
       if (text === 'delete') {
            document.getElementById('delete').removeAttribute('disabled');
       }
       else {
          document.getElementById('delete').setAttribute('disabled', true);
       }
   })
```

32. addEventListener →

- event → returns a object
- event.target → returns the tag property
- event.target.value/innerText → returns the inside text of the property. input er jonno value ar input na hole innerText use .

33. Event Bubble → event bubble html er ekebare vitor theke suru kore . 1st e ekebare vitore r items e jay then er upore ase . then tar upore . eivabe ese kaj kore.

Property →

event.stopPropagation(); → eita use korar maddome ek lavel(ek section/div) er sobgula show korbe kintu parent er kisu show korbena

event.stopImmediatePropagation(); → eita use korle jate use korbo keboi oita show korbe

34. Kunu ekta item remove kora easy but jodi input er maddome amora item ke add korar por remove korte chai tahole remove er jonno parentNode,removeChild use kora lage ebong je document e function call korbo take oboossoi parent tag hote hobe that means ekta div er majer kotogulu p add korle oi p ke remove korte chaile getelementsbuid r maje p diya na kore div diye kora lagbe.

Ex.

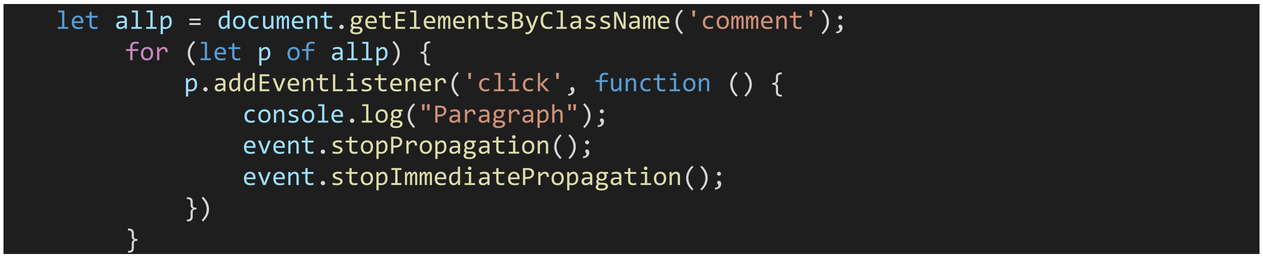
```
    //ekane je division id use kora hoyeche tar maje onek p tag asche . ei
      p tag e giye new item add hoy.
     document.getElementById('division').addEventListener('click', function () {
         event.target.parentNode.removeChild(event.target);
    })
    document.getElementById('btn').addEventListener('click', function () {
        let v1 = document.getElementById('input');
        let v2 = v1.value;

        let p = document.createElement('p');
        p.innerText = v2;
        document.getElementById('division').appendChild(p);

       v1.value = '';
    })
```

35. Amora chaile eventListener kunu for loop er osonko items er majer item gulay marte pari .

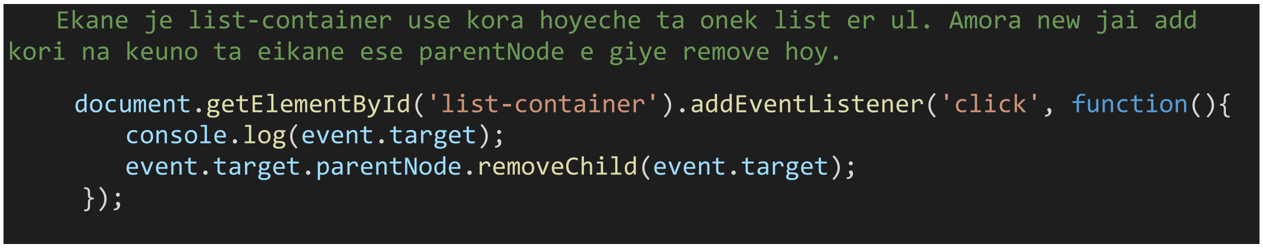
Syntex:



```
    let allp = document.getElementsByClassName('comment');
        for (let p of allp) {
            p.addEventListener('click', function () {
                console.log("Paragraph");
                event.stopPropagation();
                event.stopImmediatePropagation();
            })
        }
```

36. Loop er maje event use korle document use kora lage na . je nam diye loop chalabo tar sathe addEventListener use korleu chole. Ex. UPORE p.addEventListener(‘click’,function(){});

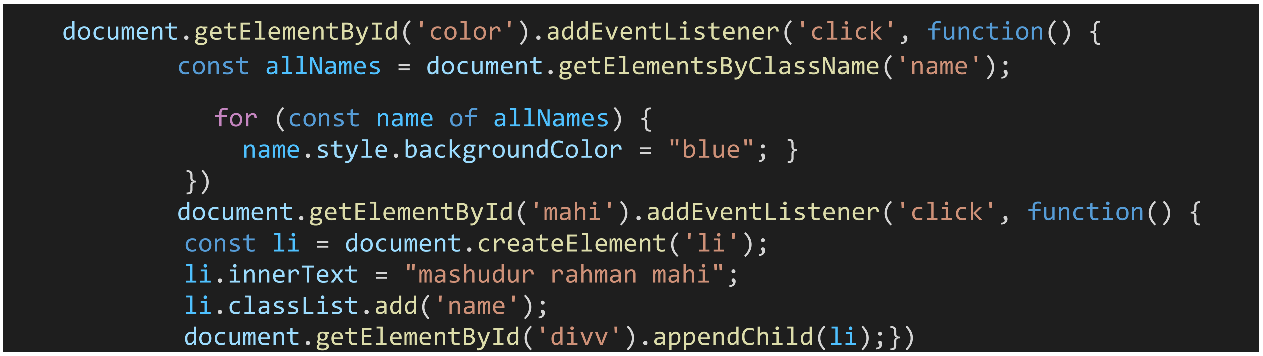
37. New add kora item remove korte chaile tar parent ke niye er maje event.target.parentNode.removeChild(event.target) use kora hoy.



```
   Ekane je list-container use kora hoyeche ta onek list er ul. Amora new jai add
kori na keuno ta eikane ese parentNode e giye remove hoy.

    document.getElementById('list-container').addEventListener('click', function(){
        console.log(event.target);
        event.target.parentNode.removeChild(event.target);
     });
```

38. ul name e jodi ekta create element kori tahole tar modde kunu class set korar jonno ul.classList.add(‘class_name’). eivabe jodi add kori tahole oi class er style o add hobe jemon,



```
    document.getElementById('color').addEventListener('click', function() {
            const allNames = document.getElementsByClassName('name');

              for (const name of allNames) {
                name.style.backgroundColor = "blue"; }
            })
            document.getElementById('mahi').addEventListener('click', function() {
            const li = document.createElement('li');
            li.innerText = "mashudur rahman mahi";
            li.classList.add('name');
            document.getElementById('divv').appendChild(li);})
```

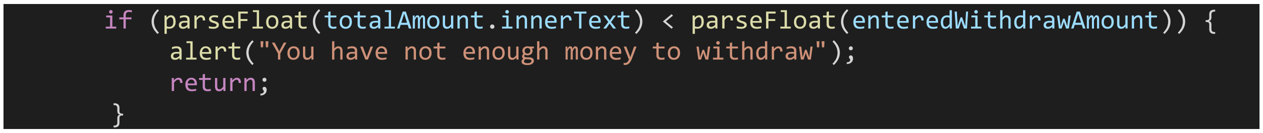
ei class name set korar fole ei class er style o add hobe.

39. To go to one page from another page using JS

- **windiw.location.href= “index1.html”;**
- **window.location= “index1.html”;**

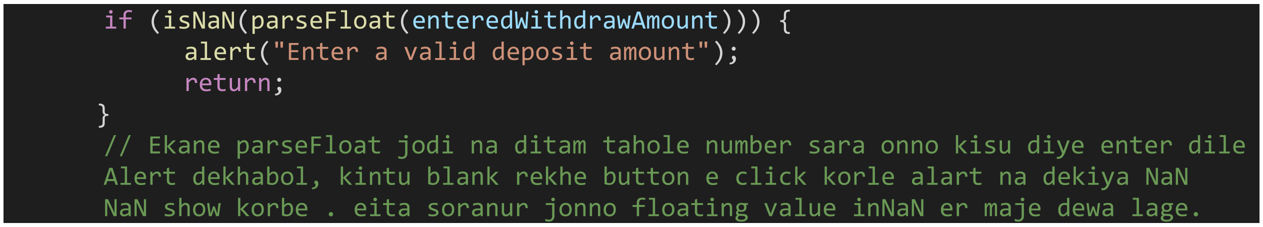
40. event e return use korle jekane use korbo oikanei event break hobe . onekta loop break er moto.

Example::



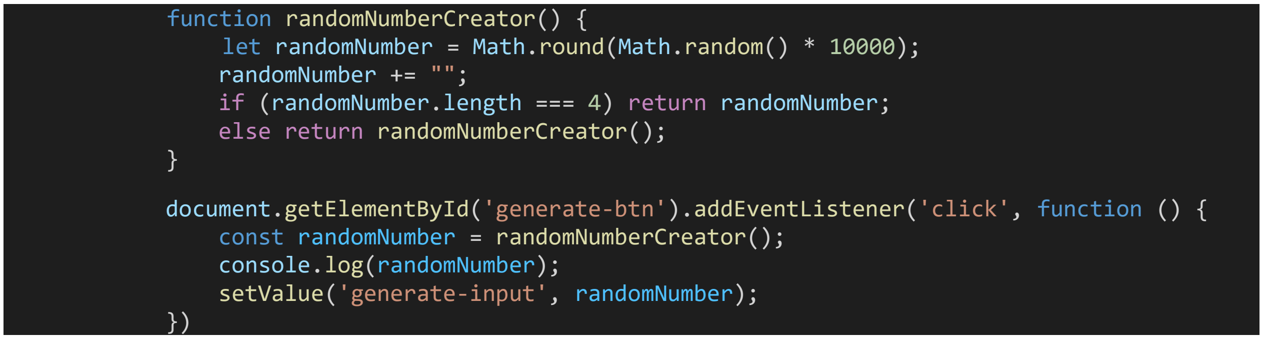
```
    if (parseFloat(totalAmount.innerText) < parseFloat(enteredWithdrawAmount)) {
         alert("You have not enough money to withdraw");
         return;
     }
```

41. isNaN(numbering-value) ➔ er maddome amora input e jodi number chara onno kichu dei tahole NaN show korbe na. Example::



```
    if (isNaN(parseFloat(enteredWithdrawAmount))) {
          alert("Enter a valid deposit amount");
          return;
    }
    // Ekane parseFloat jodi na ditam tahole number sara onno kisu diye enter dile
    Alert dekhabol, kintu blank rekhe button e click korle alart na dekiya NaN
    NaN show korbe . eita soranur jonno floating value inNaN er maje dewa lage.
```

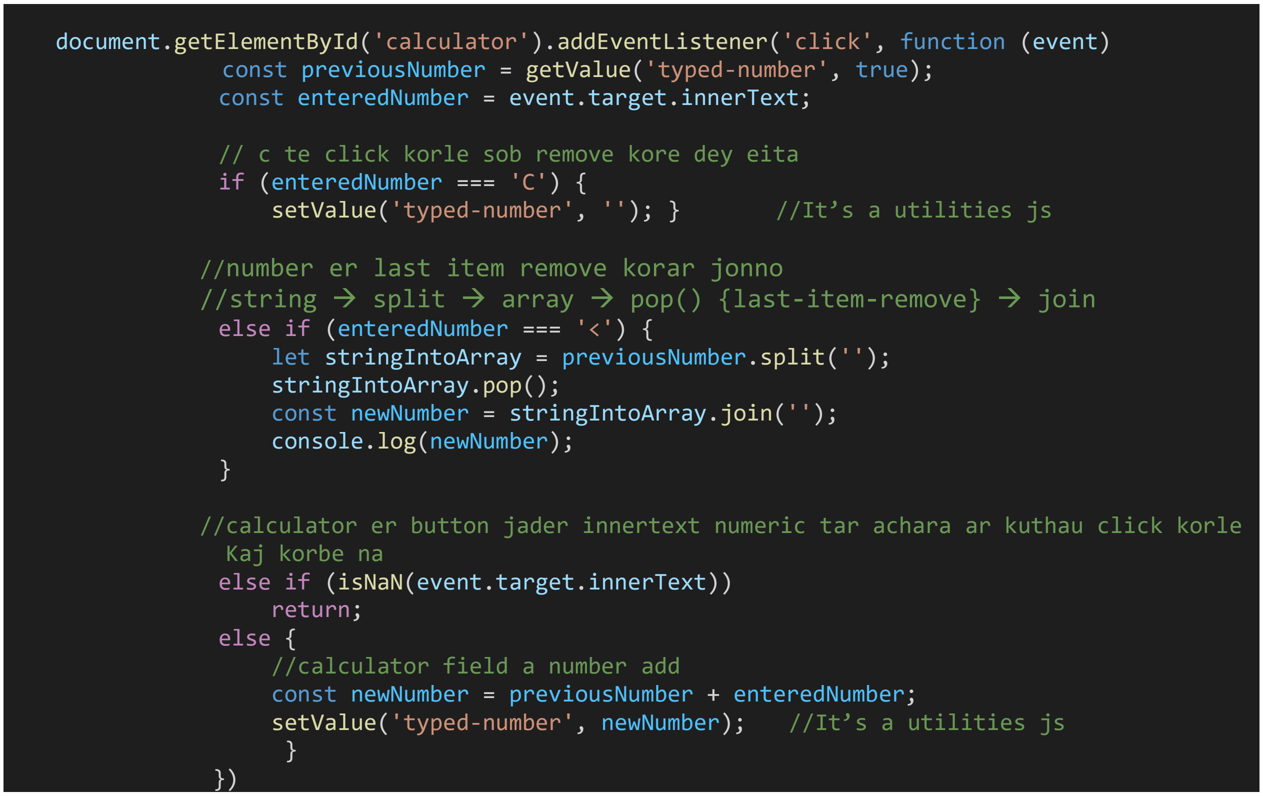
42. Random Number



```
        function randomNumberCreator() {
            let randomNumber = Math.round(Math.random() * 10000);
            randomNumber += "";
            if (randomNumber.length === 4) return randomNumber;
            else return randomNumberCreator();
        }

        document.getElementById('generate-btn').addEventListener('click', function () {
            const randomNumber = randomNumberCreator();
            console.log(randomNumber);
            setValue('generate-input', randomNumber);
        })
```

43. Calculator



```
document.getElementById('calculator').addEventListener('click', function (event)
            const previousNumber = getValue('typed-number', true);
            const enteredNumber = event.target.innerText;

            // c te click korle sob remove kore dey eita
            if (enteredNumber === 'C') {
                setValue('typed-number', ''); }        //It’s a utilities js

          //number er last item remove korar jonno
          //string → split → array → pop() {last-item-remove} → join
            else if (enteredNumber === '<') {
                let stringIntoArray = previousNumber.split('');
                stringIntoArray.pop();
                const newNumber = stringIntoArray.join('');
                console.log(newNumber);
            }

          //calculator er button jader innertext numeric tar achara ar kuthau click korle
            Kaj korbe na
            else if (isNaN(event.target.innerText))
                return;
            else {
                //calculator field a number add
                const newNumber = previousNumber + enteredNumber;
                setValue('typed-number', newNumber);    //It’s a utilities js
                 }
          })
```

44. Try, Catch, Finally

try{ console.log(a);   const a=20; }
catch(error){ console.log(error); }
finally {console.log(‘mahi’); }

try-catch-error use korle sudu try use korle hobe na . jekunu 2 ta function use kora lagbe try er bitore error takle ta catch er maddonme grohon kore show koranu jay . Then finally use korle er error er porer part e jay . Main jinis je js breakdown na hobar jonno try-catch-finally use kora hoy.

45. Date in JS → **momentjs.com**

Date er compare e hisab hoy 1st January theke jemon 2022-2-3 < 2022-2-4 hobe arki je pore hobe se boro hobe karon time besi niche .   date er declaration e new must

const date1 = new Date(‘2022-12-25’) , date2 = new Date(‘2022-12-26’) ;
console.log(date1.getTime( ) > date2.getTime( ));

false karon 25 tarik 26 tarik er age hole o boron na . Boro date hoto date2.

46. Errors In JavaScript ➔

Reference error → variable exist na hole reference error show kore thake
Syntex erroe → tikmoto syntax na dile ei error dey
Range error , cannot read properties of undefined

In this MileStone We Can Learn ➔

- *How to use js with html and css*
- *Get element from dom*
    - *getElementById, getElementsByClassName, querySelector, querySelectorAll*
- *Onclick and eventElements use*
- *Dynamically add element to the dom: document.createElement*
- *Get value from a dom element*
    - *use innerText(h1,p,div..)*
    - *use value(input , textarea)*
- *Add event listenet (event handler)*
    - *click, focus, blur, mouseover, keyup,keydown,keypress*
    - *addEventListener , onclick="myHandler()";*
- *Use function to reduce duplicate code*
- *Event bubble , event deligate*

EcmaScript

1. var/let/const ➔

   i. let,const is a block-scoped data-type that means eder jodi function/ kunu kichur bitore use kora hoye tkahe tahole baire same variable er jodi output dekte chai tahole undefined dekhabe

   ii. var is a global data-type that means eder jodi function ba blacker er maje use kora hoy tahole ta auto upore chole jay end globally declared hoye jay ebong etai main issue var use na korar.

   iii. Const e declaration er sathe sathei initialize kore dite hoy

   iv. For multiple same variable declaration var can be use but let/const cannot. But it's not recommended. Ex.

var a=20; var a=30;	it is right
let a=20, let a=30;	it is identifier error

   v. In const array/object can be undateable but not changalbe eith variable. Ex.

const arr=[1,2,3,4,5];	
arr.push(6,7,8);/arr.pop()/shift()/unshift()	possible
arr[2]=30	possible
🚫arr=[2,3,4,5,6];	not possible
const obj1={name: "mahi", age:21}	
➡ obj1.name= "mashudur"; obj1.age=22;	possible
🚫obj1{name= "mashudur"}	not possible

2. Function Default Parameter means user jodi function e kunu input na ney tahole function auto ekta input default vabe nibe jate code run hoy & NaN na ase.

   i. if(b=== undefined) b=0; es6 asar age ei system e default parameter enroll kora

   ii. b=b||0; hoto function er maje

   iii. function add(a,b=0,c=0){ return a+b+c;} es6 e direct function declaration e enroll hoy

3. String declaration ➔ ` ` er maje string likhe variable ${ } er maje lekha hoy . eivabe multiple line kub easily \n chara eboung (+,) chara variable declaration kora hoy.

   Ex. const a=20,b=30; console.log(`a is ${a} and b is ${b}
   The sum of a+b = ${a+b}`);

4. Array copy & Edit ➔ 3 itiots(…arr) const arr=[1,2,3,4,5]

- console.log(…arr) output: 1 2 3 4 5 array chara tar elements ken niye thake
  console.log(Math.max(…arr)) normally array r maje max/min bar na kora geleo eivabe kora jay
- const arr1=arr; eivabe jodi array copy kori tahole jekunu ekta array te customize korle 2 ta array change hoy karon tara tader id share er maddome kaj kore thake
- const arr2=[…arr] eivabe 1st array man niye take bracket er maje ene array build korle next time e ekta change korle onnoti change hoyna
- const arr3=[1,2,3, …arr ,4,5,6] eivabe edit o kora jay
- function new( ) { ………. return [a,b,c] } eivabe ekta function theke ekadik item return kore
  const [x,y,z] = new( ) array r maddome grohon kora jay

5. Normal & Arrow Function

- Normal function ➔ anonymous function

  i. function add(a, b) { return a + b; }
     console.log(add(1, 2));
  ii. const b = function add(a, b) { return a + b; }
     console.log(b(2, 3));
  iii. const c = function(a, b) { return a + b; } annonymus function
     console.log(c(3, 4));

- Arrow Function ➔ () => use kora hoy | return use hoy na tobe multiline function e hoy er ketre function name e argumanet na diye je varibale er maje function dewa hoy tate argument

i.	Empty arrow function	use blank bracket	() =>
ii.	Single arrow function	bracket can be used or not	(a) => / a =>
iii.	Multi arrow function	bracket must used	(a,b,c,d) =>

```
const d = (a, b) => a + b;              console.log(d(4, 5));

1. single parameter
const e = a => a * a;                   console.log(e(5));
const g = (a) => a * a;                 console.log(g(6));


2. empty parameter
const f = () => "40";                   console.log(f());
const i = () => ("Mashudur Rahman Mahi");          console.log(i());


3. Multi Parameter . Multi Line Function e return use hoy
const j = a => 10;
const k = (a, b = 0, c = 0) => {
 const sum = a + b + c;
return sum;
}
console.log(j(10));
console.log(k(12));
```

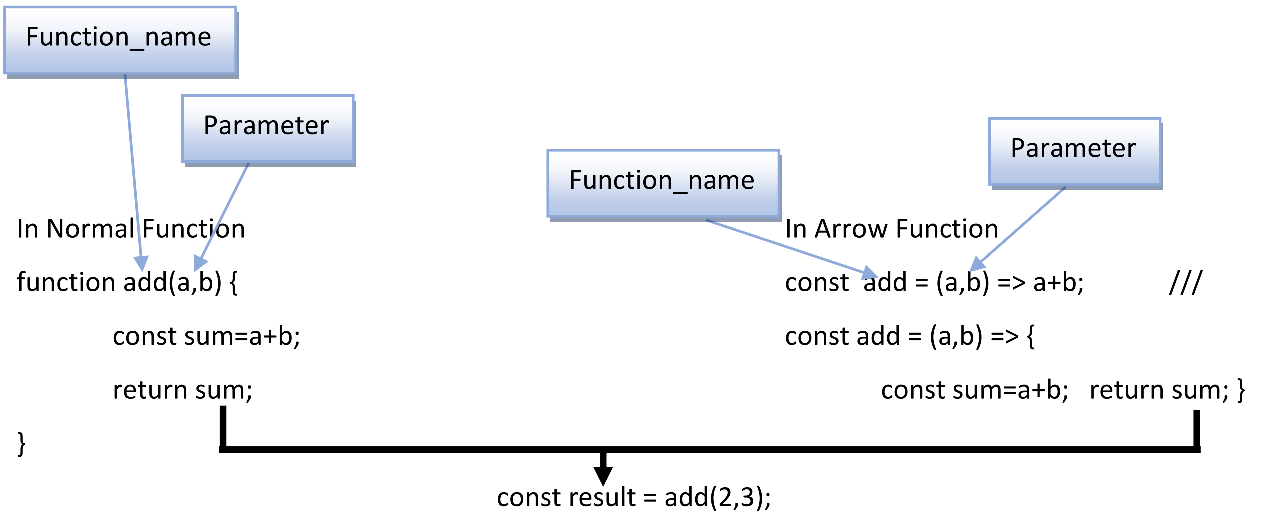
```
function  add(a,b){    ➔   const add = function add(a,b){ return a+b;}  ➔      const add= function (a,b) {return a+b;}

return a+b; }              console.log(add(1,2));                                console.log(add(1,2))

                                                                                          ⇩

                                                                const add = (a,b,c) =>  a+b+c;      console.log(add(1,2,3));
```

Arrow Function



In Normal Function

```
function add(a,b) {
        const sum=a+b;
        return sum;
}
```

In Arrow Function

```
const  add = (a,b) => a+b;          ///
const add = (a,b) => {
        const sum=a+b;   return sum; }
```

```
const result = add(2,3);
```

Explanation

- const add = (a,b) => { const sum=a+b;   return sum; }
    - Ekane add kunu variable na . eti ekta function name just like the normal function name.
    - Amora function call er somoy je argument dei ta parameter ney .
    - Onek somoy arrow function e confution lagte pare je result kar modde takbe . add er maje na sum er maje . kintu ekene add vatiable na, only function name ar value function name er maje takte pare na . Tai return korle je kayga theke kora hobe sekane hoyto console.log hobe na variable e value set hobe.

6. Distruct
   - Array Distruct ➔ [ ] = [ ]
   - Object Distruct ➔ { } = { }
      i. Object distruct er ketre bam pasher bracker er majer variable er name gula object er bitore je variable ase tader namer same hote hobe but array distruct er ketre lagbe na
      ii. ...variable_name → array/ object er maje obosisto ja ase tader niye arekta array/object create.

   EX..

      i. const obj={ name: ‘mahi’ , age:21}   const {name,age} = obj
      ii. const {name,age,...other} = {name: ‘mahi’, age:21, roll:13, reg:22}
      iii. const arr=[1,2,3,4,5]   const [a,b,c,...other]= arr;
      iv. const [a,b,c,...other]= [1,2,3,4,5,6,7,8,9];

7. Optional Chain ➔ object er ketre optional chain use korar fole compiler error na diye next e chole jay
   Meathod → ?.
   console.log(array1[0]?.namee);   ekane name name jodi kisu na thake tahole undefined diye porer line e chole jabe but error dekhabe na

8. MAP ➔
   Map er maddome ekta array r prottekta elements ken iye kub e sohoje kaj kora jay . for of loop chalanur dorkar pore na. It returns a new array but doesn’t customize the old array.
   EX.
      i. const arr=[1,2,3,4,5];
         console.log(arr);   console.log(arr.map(x=>x*2))
         console.log(arr.map(x=>x*4))   console.log(arr.map(x=> x*x));
      ii. const names=['mashudur', 'rahman','mahi'];
         console.log(names.map(firstAlphabet => firstAlphabet[0]));
      iii. const arrObj=[{name:'Mashudur', id:1, reg: 2020331513}, {name:'Rahman', id:2, reg: 2020331513}]
         console.log(arrObj.map(a => a.id))   console.log(arrObj.map(a => a.name))

9. forEach ➔ As same as map but it doesn’t return anything from provious array. It can be used to show something. Ex..
   const arr=[1,2,3,4,5] , names = [‘mashudur’, ‘rahman’, ‘mahi’]
   arr.forEach(a => console.log(a*2));   names.forEach(name=>console.log(name));

10. filter, find ➔ filter joto match khabe sob return kore dibe
   find joto match khabe tar sudu 1st element ta return kore dibe
   Syntex :: arr.filter/find(x => condition);
   EX.
   const arr1= [10,20,30,40,50,60,70,80,90,100];
   console.log(arr1.find(x => x>=50));
   console.log(arr1.filter(x => x>= 50));

11. reduce ➔ syntax :: arr.reduce(accumulator, initial_value);
      accumulator ➔ (first,last) => first+last;
   accumulator is a arrow type function which consists two variable
   Ex.
   const arr=[1,2,3,4];   console.log(arr.reduce((x,y)=>x+y,0);
   console.log(arr.reduce((x,y)=>x*y,1);

Class

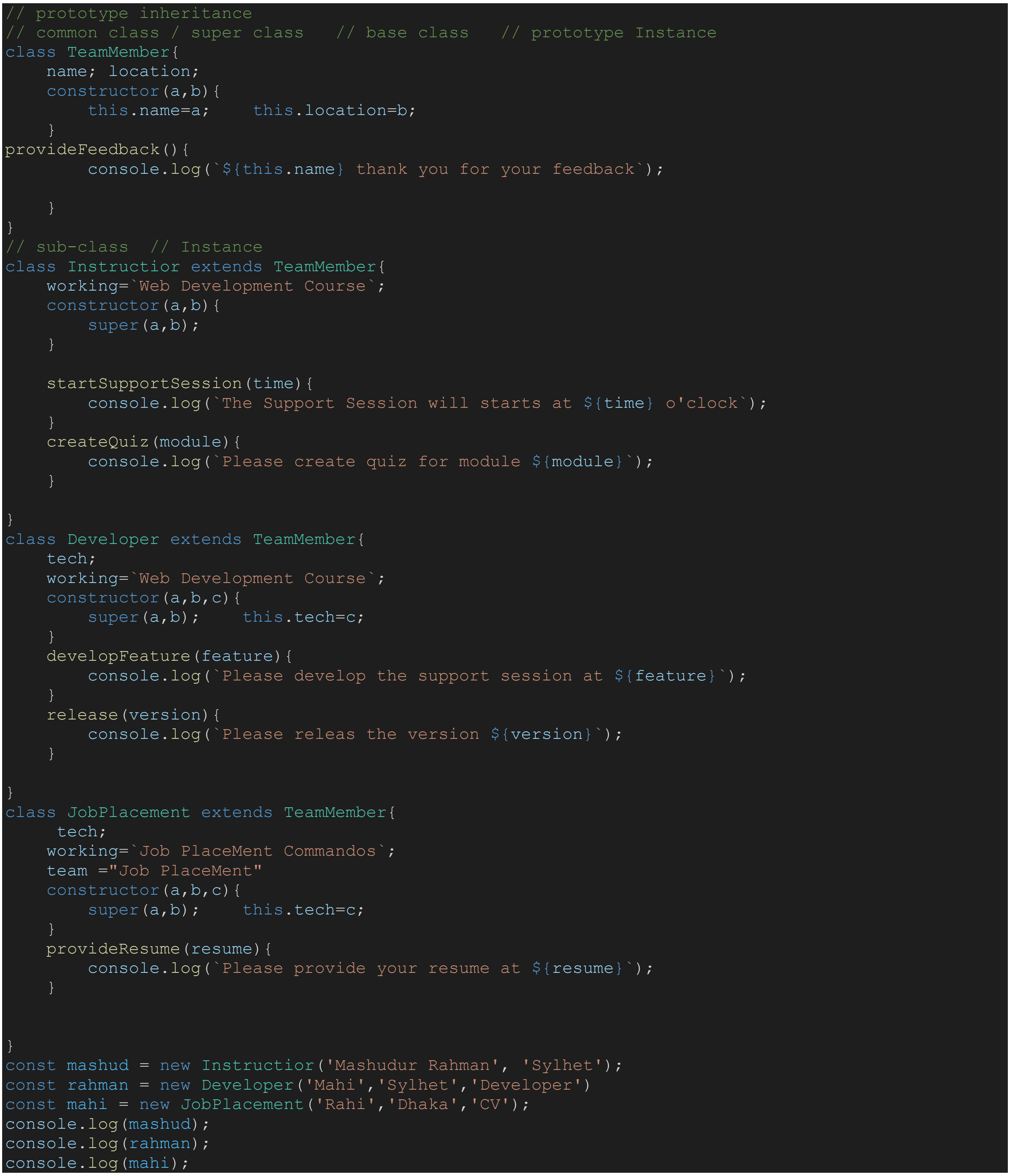
- Class name should be **pascal-case**. Ex: class Instructor{………}

  It works like function but not use can not use () for declaring a class
- Super class ar as same as normal class. Just all common properties of different classs are inside the super class. For Connection common/super/base class we should work in 3 process.
  1. Use **extends** in subclass. Ex.   class Instructor extends Teammember{………..}
  2. Use **super** to give property to the base class. Ex.   constructor(name){ super(name); }
- Function, Let, Const cannot be used inside the class
- Jesokol variable er value baire theke dewa hoy tader sudu name declare kora hoy. EX.   name;
- Use **constructor** to set value in class.   constructor(name)   { this.name = name; }

- Can use any function inside the class like this.   Function_Name(parameter){…………….}

  No need to write function before the function_name
- Declaring a variable of the function.
  1. Must use   new   property
  2. Must use   ( )   property
  3. Give the value inside the ( ) parenthesis property

     Ex.   const membersInfo = new Instructor("Mashudur Rahman Mahi");



```
// prototype inheritance
// common class / super class   // base class   // prototype Instance
class TeamMember{
    name; location;
    constructor(a,b){
        this.name=a;    this.location=b;
    }
provideFeedback(){
        console.log(`${this.name} thank you for your feedback`);

    }
}
// sub-class  // Instance
class Instructior extends TeamMember{
    working=`Web Development Course`;
    constructor(a,b){
        super(a,b);
    }

    startSupportSession(time){
        console.log(`The Support Session will starts at ${time} o'clock`);
    }
    createQuiz(module){
        console.log(`Please create quiz for module ${module}`);
    }

}
class Developer extends TeamMember{
    tech;
    working=`Web Development Course`;
    constructor(a,b,c){
        super(a,b);    this.tech=c;
    }
    developFeature(feature){
        console.log(`Please develop the support session at ${feature}`);
    }
    release(version){
        console.log(`Please releas the version ${version}`);
    }

}
class JobPlacement extends TeamMember{
     tech;
    working=`Job PlaceMent Commandos`;
    team ="Job PlaceMent"
    constructor(a,b,c){
        super(a,b);    this.tech=c;
    }
    provideResume(resume){
        console.log(`Please provide your resume at ${resume}`);
    }


}
const mashud = new Instructior('Mashudur Rahman', 'Sylhet');
const rahman = new Developer('Mahi','Sylhet','Developer')
const mahi = new JobPlacement('Rahi','Dhaka','CV');
console.log(mashud);
console.log(rahman);
console.log(mahi);
```

Object

1. JavaScript Object Declaration Meathod ➔ ei meathod gola diye object clone kora jay na karon tara ekoi Reference share kore thake tai ekta change korle onnota o auto change hoye jay

- General Object literal Meathod

```
const mahi = {
    name: "Mashudur Rahman Mahi",
    age: 20,
    college: "Sylhet Engineering College"
}
console.log(mahi);
```

- Object Constructor

  i. New object create kora

```
const mahi = Object();
mahi.name = "Mashudur Rahman Mahi";
mahi.age = 20;
mahi.college = "Sylhet Engineering College";
console.log(mahi);
```

  ii. Previous kunu object ke call kore ene tar sathe add kora. Eita Object.create er soman hole o create e object.property dite hoy . object ke console.log korle black dekay kintu ekane blank dekay na . object ke direct show koranu jay

```
const mahi1 = Object(mahi);
mahi1.college = "Sylhet Engineering College";
console.log(mahi1);
```

- ES6 Class

```
class Mahi {
    name = "Mashudur Rahman Mahi";
    constructor(age) {
        this.age = age;
    }
    college = "Sylhet Engineering College";
}
const mashud = new Mahi(20);
console.log(mashud);
```

- Object.create() ➔ Object chain type clone

```
const mashud = {
    name: 'Mashudur Rahman Mahi',
    age: 20,
    college: "Sylhet Engineering College"
}
const mahi = Object.create(mashud);
console.log(mahi.name);
```

ekane jodi amra direct object name jemon mahi console.log kori tahole blank object show korbe . but object.property jemon mahi.name jodi dei tahole er bitorer property show kore thakbe

2. Ovject value received meathod ➔ const obj ={name: ‘mahi’ , age :21} → 3 meathods

- obj.name / obj.age
- obj[‘name’] / obj[‘age’]
- By Using variable: const nm= ‘name’, ag= ‘age’; obj[nm] / obj[age]

3. JavaScript Object Clone ➔ ekane obj2 te obj1 clone jodi korte chai tahole 2 ta meathod

- Shallow Copy ➔ const obj2 = {…obj1};
- Json parsing ➔ const obj2 = JSON.parse(JSON.stringihy(obj1));

4. JavaScript Meathod in Object ➔ object er maje function use korle take meathod bola hoy

- Object er function er bitore theke bairer kunu variable ke call korte chaile this.variableName Use kora hoy.

```
Const person = {
      name: “Mashudur Rahman Mahi”,
      salary: 20000,
  treat: function(treat, tip) {
            availableMoney = this.salary – treat – tip;
            this.salary = availableMoney;
  } }
console.log(person);
person.treat(10000, 1000);
console.log(person);
```

5. JavaScript Object Propertirs ➔ const person = { name: ‘Mahi’ , age:21 }

Object.keys(person) → object er sob keys show
Object.values(person) → object er sob values show
Object.entries(person) → object er sob keys & values eksathe show kore
Object.seal(person) → object property add/delete kora jay na but value change kora jay
Object.freeze(person) → add/delete/value change kisui possible na

Object deleting 2 Meathods ➔

delete person.name → person namok object er name property delete kore
const {name, …other} =person → person object theke name bad diye baki sobkichu other e niyechi

6. Loop for Object

- Array for of → showing value for in→ showing index number
- Object for in → showing object keys for of → error

i. ```
for(const item in obj){
    console.log(item); }
```
```
for(const item of Object.keys(obj)){
    console.log(item); }   // keys showing
```

ii. ```
for(const item in obj){
    console.log(obj[item]); }
```
```
for(const item of Object.values(obj)){
    console.log(item); }          // values showing
```

iii. ```
for(const item in obj){
    console.log(item,obj[item]); }
```
```
for(const [key,value] of Object.entries(obj)){
  console.log(key,value); }   // keys,values showing
```

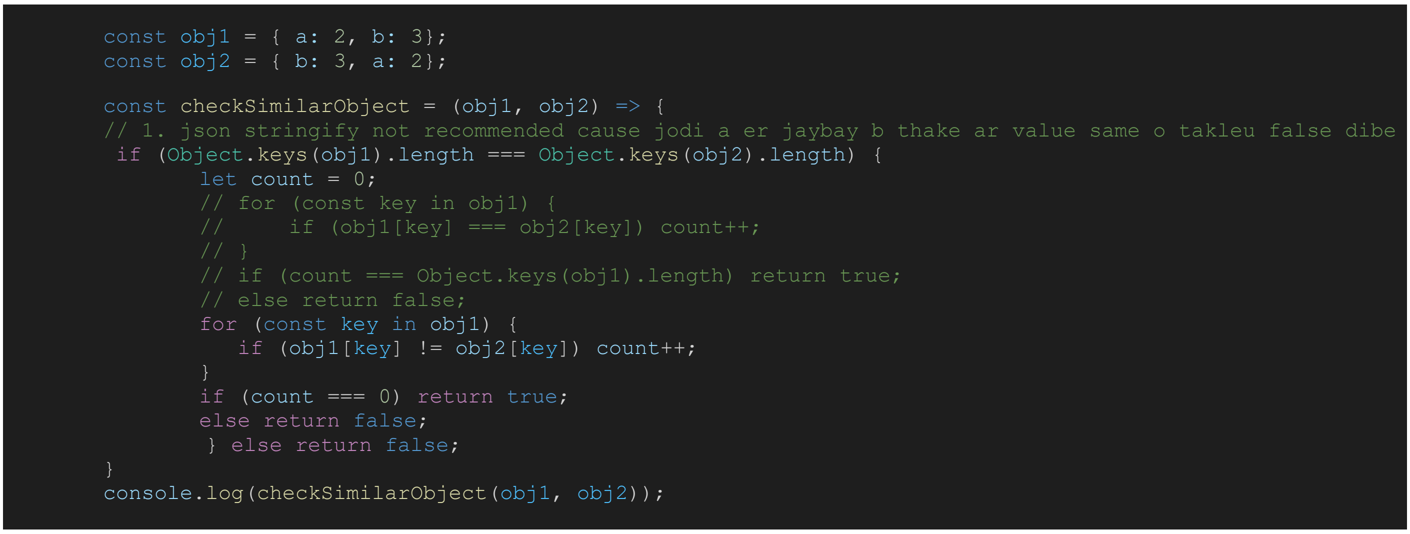
7. Object Similarity → const obj1={a:2} , obj2={a:2} , obj3={} , obj4={} , obj5=obj1;

False → obj1===obj2
obj3 === obj4
cause value same hobar sathe sathe reference o same hote hoy

True → obj5 === obj1
ekane 2 tar e value and reference same

8. Object Comparism → use different kind of snippets from google

- Json stringify is a meathod but it is not recommended to use because of if the keys of both object are not presents in serial by than it returns false



```
const obj1 = { a: 2, b: 3};
const obj2 = { b: 3, a: 2};

const checkSimilarObject = (obj1, obj2) => {
// 1. json stringify not recommended cause jodi a er jaybay b thake ar value same o takleu false dibe
 if (Object.keys(obj1).length === Object.keys(obj2).length) {
        let count = 0;
        // for (const key in obj1) {
        //   if (obj1[key] === obj2[key]) count++;
        // }
        // if (count === Object.keys(obj1).length) return true;
        // else return false;
        for (const key in obj1) {
           if (obj1[key] != obj2[key]) count++;
        }
        if (count === 0) return true;
        else return false;
         } else return false;
}
console.log(checkSimilarObject(obj1, obj2));
```
```

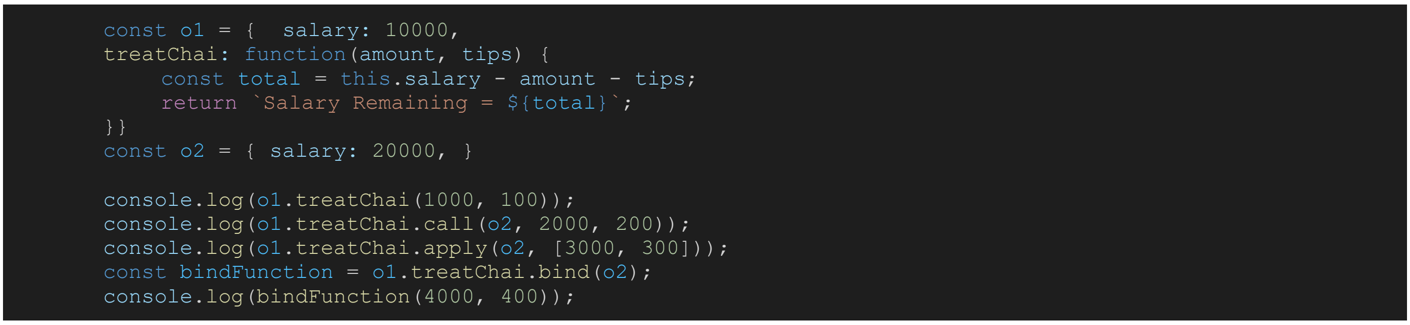
9. **this** in JavaScript usage

- In an object meathod it refers to the object
- In an eventHangler it refers to the element that received the event
- In object arrow meathod, onclick function or others it refers to the global object
- In a function it refers to the global object except new function ()

10. Borrow in Object ➔ ek object er this.value ta onno object er meathod function e shift kore thake

- **call** ➔ firstObject.functionName.call(secondObject, argument,argument,….)
- **apply** ➔ firstObject.functionName.apply(secondObject, [argument,argument,…])
- **bind** ➔ const binding = firstObject.functionName.bind(secondObject)
  binding(argument,argument,……..)

call comma, apply array, bind, separate function type



```
const o1 = {  salary: 10000,
treatChai: function(amount, tips) {
     const total = this.salary - amount - tips;
     return `Salary Remaining = ${total}`;
}}
const o2 = { salary: 20000, }

console.log(o1.treatChai(1000, 100));
console.log(o1.treatChai.call(o2, 2000, 200));
console.log(o1.treatChai.apply(o2, [3000, 300]));
const bindFunction = o1.treatChai.bind(o2);
console.log(bindFunction(4000, 400));
```

eder maddome sudu ek object er ekta value onno object er function er this.value r maje transfer kora hoy . normally o1.treatChai(1000,100) dile ta treatchai meathod function er maje je salary ase ta nibe this.salary hisebe . kintu jodi amra call/apply/bind er maddome onno function je include kore dei tahole treatChai meathod function tar uporer o1 er salary ne niye o2 er salary this.salary hisebe nibe

API

(Application programming interface)

1. JSON.stringify(object_name) → convert object into string type
2. JSON.parse(stringify_object_name) → convert previous stringify object into onject
3. fetch / async()

- **fetch** → fetch url , then arrow response json function , arrow data pass

```
 const url = '………………………'
 fetch(url)
.then(response => response.json())
.then(data => console.log(data))       /   .then(data => function_name(data))
.catch(error => console.log(error))
```

Process:::

→ Function e data pass korar somoy alltime array te convert kore patabo
→ .json() er maddome url theke data call
→ eivabe data show kora hoy / data function e pass kora hoy
→ er maddome url e error thakle ta catch kore console e error show kore
→ function theke data pass

- **async()** →

```
const arrow_function_name = async() => {
      const url = "…………..";
      try {
      const response = await fetch(url)
      const data = await response.json()
      console.log(data);       /       function_name(data)
      }
      catch(error){ console.log(error); }
```

→ .then er bodole await use kora hoy

4. Using fetch by event-hangler

   i. Onclick

```
 const loadQuote = () => {
 fetch('https://api.kanye.rest/')
.then(ref => ref.json())
.then(data => displayQuote(data)) }
```

→ arrow function use korar try korbo

   ii. Event

```
 document.getElementById('btn-add').addEventListener('click',function(){
 fetch('https://api.kanye.rest/')
.then(ref => ref.json())
.then(data => displayData(data))})
```

→ event e kunu kichu kaj na korte parle onclick e try korbo.

5. Fetch korar por html e kunu kicu array r maddome add korar ketre (for of)/ forEach use kora jay.

   Ex. for(const data of allData){……………} allData.forEach(data => {………})

6. innerHTML er maje dynamic button e onclick function use korle jodi function er bitore jodi

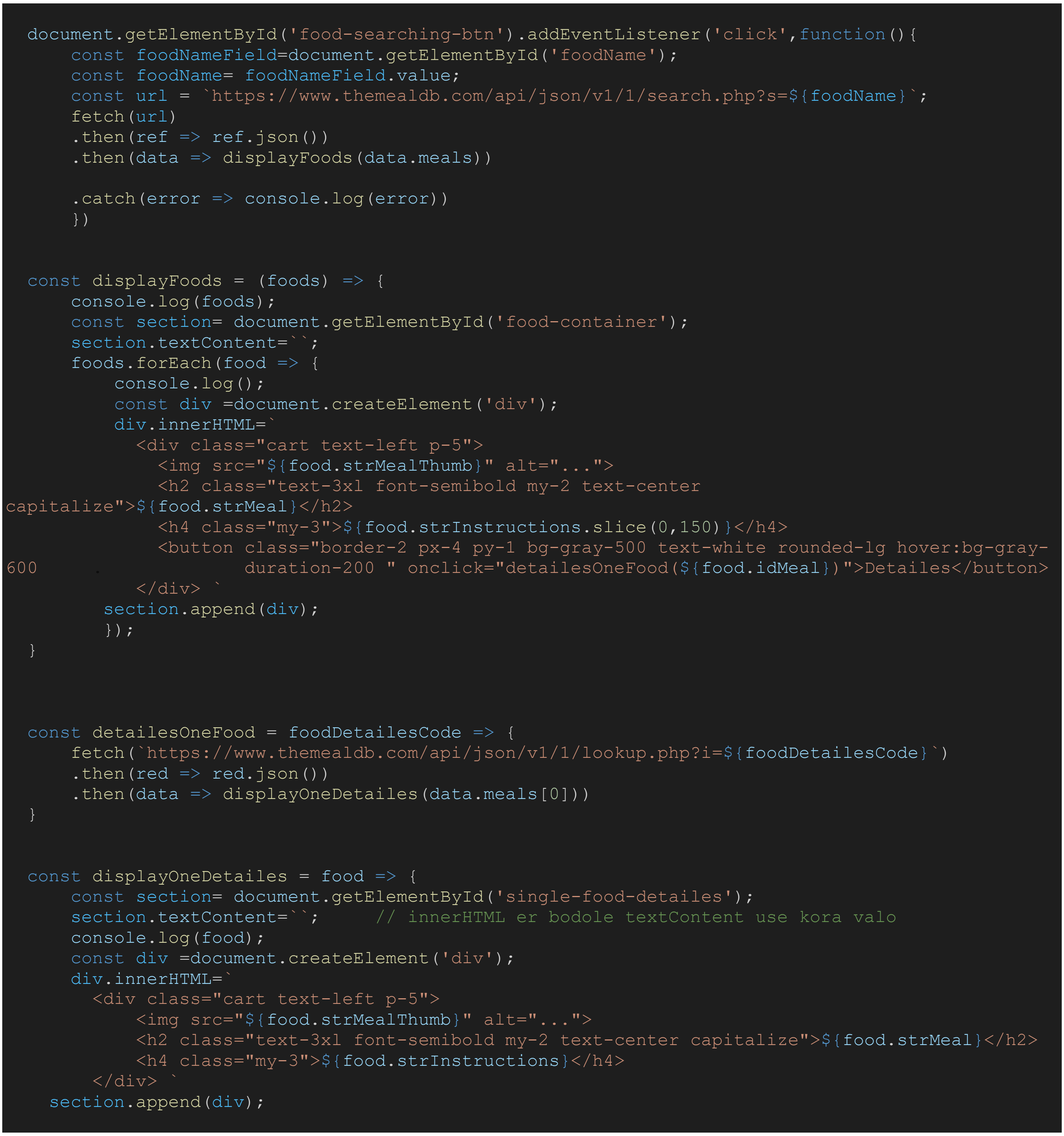
   String pass hoy tahole ""/'' quote er maje dite hobe.Ex. onclick="loadDetailes('${country.alpha3Code}')"

   baire single hole bitore double , baire double hole bitore single,

   jodi number pass hoy tahole quote na dileo hoy. Ex. onclick=" loadDetailes(${country.code})

7. .innerHTML= " " → .textContent= " " bitorer item gula remove korar ketre textContent use kora hoy

An Example ::   hint. try to use tailwind/bootstrap to make a webside before add the data and then copy the code



```
  document.getElementById('food-searching-btn').addEventListener('click',function(){
      const foodNameField=document.getElementById('foodName');
      const foodName= foodNameField.value;
      const url = `https://www.themealdb.com/api/json/v1/1/search.php?s=${foodName}`;
      fetch(url)
      .then(ref => ref.json())
      .then(data => displayFoods(data.meals))

      .catch(error => console.log(error))
      })


  const displayFoods = (foods) => {
      console.log(foods);
      const section= document.getElementById('food-container');
      section.textContent=``;
      foods.forEach(food => {
          console.log();
          const div =document.createElement('div');
          div.innerHTML=`
            <div class="cart text-left p-5">
              <img src="${food.strMealThumb}" alt="...">
              <h2 class="text-3xl font-semibold my-2 text-center
capitalize">${food.strMeal}</h2>
              <h4 class="my-3">${food.strInstructions.slice(0,150)}</h4>
              <button class="border-2 px-4 py-1 bg-gray-500 text-white rounded-lg hover:bg-gray-
600                    duration-200 " onclick="detailesOneFood(${food.idMeal})">Detailes</button>
            </div> `
         section.append(div);
         });
  }



  const detailesOneFood = foodDetailesCode => {
      fetch(`https://www.themealdb.com/api/json/v1/1/lookup.php?i=${foodDetailesCode}`)
      .then(red => red.json())
      .then(data => displayOneDetailes(data.meals[0]))
  }


  const displayOneDetailes = food => {
      const section= document.getElementById('single-food-detailes');
      section.textContent=``;     // innerHTML er bodole textContent use kora valo
      console.log(food);
      const div =document.createElement('div');
      div.innerHTML=`
        <div class="cart text-left p-5">
            <img src="${food.strMealThumb}" alt="...">
            <h2 class="text-3xl font-semibold my-2 text-center capitalize">${food.strMeal}</h2>
            <h4 class="my-3">${food.strInstructions}</h4>
        </div> `
    section.append(div);
```

Debugging

1. JavaScript is a single threaded, non-blocking, asynchronous, concurrent language.
2. JavaScript runs on V8 engine with makes by c++ because of c++ is close to the matchine language.
3. JavaScript have a concept that named (event loop) that manage the synchronous and asynchronous and run them.
4. Event Loop have 3 parts. Heap, Stack, Queue(LIFO)

   Must watch the Video : What the heck is the event loop anyway ? Plilip Roberts
5. setTimeout, fetch, async,await are the asynchronous function.
6. setTimeout/setInterval(function,miliseconds) , clearTimeout/clearInterval(variable_name)
   - setTimeout → a function that show a specific delay of time and exicude just one time is setTimeout function.
   - setInterval → It also delay like setTimeout but execute infinite times. User should stop the repeating execute by condition.
   - clearTimeout / clearInterval → clearTimeout stops the works of setTimeout and clearInterval stops the works of setInterval. But they give to the variable name working on setTimeout or setInterval.

Ex. setTimeout & clearTimeout

```
const showThird = () => {
console.log('third');}
console.log('first')
console.log('secion')
setTimeout(showThird, 2000);
const timing = setTimeout(() => {
console.log('forth');
}, 1000);
clearTimeout(timing);
console.log('fifth');
console.log("six");
```

setInterval & clearInterval

```
let count = 0;
        console.log(1);
console.log(2);
        const inter = setInterval( () => {
        console.log(++count);
        if(count===0) clearInterval(inter); } , 1000);
//clearInterval(inter);
console.log(3);
```

7. document.body.contentEditable = true → By using this code on any webside console we can edit anything on the website
8. In Search Bar → chrome://settings , chrome://version , chrome://newtab , chrome://restart
9. **location** → opens many option if we write on console in any webside
   - location.reload() → reload the webpage
   - location.assign('url') / location.href('url') / window.location.href('url') → opens a new webside
   - location.replace('url') → also opens a new website.

   assign ad replace do same works but assign/href keeps previous page on memory . And the location.replace function replace the previous page into new page . We cannot find previous page on the memory. That's the main different both of the system.

10. location === window.location   document === window.document

    location.href === window.location.href

11. **history** ➔

- history.length → show the hength of the browser history
- history.back() → back 1 page from the current page
- history.forward() → go 1 page forwards from the current page
- history.go(0)/.go() → it reload the webpage like window.reload()
- history.go(-2) → back 2 page earlier from current if 2 pages earlier is available
- history.go(-n/n) → shift n page back or front if those page ar available on history. Otherwise I will give a undefine error.

12. **alert , confirm and prompt** ➔

- Alert → It will give a alert on website. Ex. `alert("Fuck off!!");`
- confirm → It can returns **true or false**. Ex.

```
const decission = confirm("Are you coming to the picnic?")
if (decission == true) alert("Dosto 500 tk bkash kor");
else console.log('DGM!');
```

- prompt → It can take **some information.** Ex.

```
const name = prompt('Tell Us Your Name');
if (!!name)/if(name) {                //both are same. Both indicates if anythings present or not
   console.log("WelCome Here " + name);
}
```

13. Cookies ➔ Module 38-5.   We can find it on application.

    It is not safe alltime to use but http cookie is safe for doing some work.

- document.cookie → it can show all the cookie of the webside
- document.cookie.split(‘; ’) → there are (;) presents between 2 cookies . Using this cookies split and convert array type
- document.cookie.split(‘; ’).forEach(x=>console.log(x)); → showing the cookies seperately after it to array.

14. https://codomain.www.domain.com:80/page/content.pho?id=123#top

    protocol+co-domain+domain+port+pathname/address+search+hash

    Here: protocol (http://)   codomain   domain (www.domain.com)

    port (:80)   pathname (/page/content.pho)

    search (?id=123)   hash (#top)

15. Storage In Javascript ➔ There are two types of storage in JS. Sessional and Local Storage.

But the main difference is : Data in Local Storage → doesn't expire

Data in Sessional Sro. → cleared when the page session ends

There are every information on both storage in String type.

16. Local Storage ➔ **localStorage.getitem/setItem/removeItem/clear**

- localStorage.getItem('item_name') → 2 types information. String or Object
  - string → same above meathod
  - object → have to use JSON.parse()

    JSON.parse(localStorage.getItem('item_name'));
- localStorage.setItem('item_name',value) → if string as same as meathod
  - object → have to use JSON.stringify()

    JSON.stringify(localStorage('item-name',value));
- localStorage.removeItem('item_name') → removed just 1 perticular item
- localStorage.clear() → removed all the local Storage item and information

Ex.

1) setItem as string and removeItem and clearItem

```
localStorage.setItem('name', name);
localStorage.clear();
localStorage.removeItem(itemName);
```

2) setItem as Object ➔ **5 Steps**

I. take value from input

II. make cart ={} or rollback cart if it present on localStorage

III. add element and value in cart like this. cart[product] = quantity;

IV. make cart function into JSON.stringify(). Info = JSON.stringify(cart)

V. push the cart by using localStorage.setItem('cart',info)

Example Process:

- Take input
- Check cart available or not in localStorage. If available turn it back by json parse. If not create a object
- Set value into object type cart. Ex: cart[product] = quantity;
- Check cart available or not in localStorage. If available turn it back by json parse. If not create a object
- Update cart into localStorage by using json stringiify
- Display newly added item
- Display previous added item at start of the sebsite.
    - Use json parse to to bring information from localStorage
    - Then use loop on object and pass product and quantity to displaying unit

```
    // reveiving the value
    const valueReveived = (id) => {
            const field = document.getElementById(id);
            const value = field.value;
            field.value = ``;
            return value;}


    const addProduct = () => {
    const product = valueReveived('product-name-field');
    const quantity = valueReveived('product-quantity-field');
    setToLocalStorageAsObject(product, quantity);}


 //checking previous cart available or not
    const previousCartAvailability = () => {
            const cartValue = localStorage.getItem('cart');
            let cart = {};
            if (cartValue) {
            cart = JSON.parse(cartValue);
            }
            return cart;}


    // add extra information on cart
    const setToLocalStorageAsObject = (product, quantity) => {
    const information = previousCartAvailability();
    information[product] = quantity;      // information.product dewa jabe na cause product is a
                                  variable . er bitore string ase

    displayCartInformation(product, quantity);
    updateCart(information);}


// updating cart to local storage
    const updateCart = (information) => {
    const converInformationIntoStringify = JSON.stringify(information);
    localStorage.setItem('cart', converInformationIntoStringify);}


    // display cart as ul
    const displayCartInformation = (product, quantity) => {
            const ul = document.getElementById('product-container');
            const li = document.createElement('li');
            li.innerText = `${product} = ${quantity}`;
            ul.appendChild(li);}


    // displaying previous saved products
    const displayPreviousSavedProducts = () => {
            const information = JSON.parse(localStorage.getItem('cart'));
            for (let info in information) {
    displayCartInformation(info, information[info]);
            }}
    displayPreviousSavedProducts();
```

17. JavaScript is a Dynamic typed programming language

18. 3 Datatypes. Premitive (Number,String,Boolean) Non-Primitive(Array, Object)

Trivial(Undefiined,Null)

- In Primitive → let a=20,b=a; b=10; value: a=20, b=10. Both doesn't changed

  Because primitive datatype share only value, not the reference.

- In Non-Primitive → It share not only the value but also the reference .

  Let a=[1,2,3],b=a; b[1]=50; result: a=b=[1,50,3]. Both value changed.
  For object it is similar because it share the reference. But if we want to
  Change full array/object than the value will be different . But for partial
  Change the value will be same

19. isNaN → not a number check

!isNaN → is Number check

20. 8 ways to get undefined

i.	let/var variable that is not initialized.	let a; console.log(a);
ii.	function with no return.	function sum(a,b){console.log(a+b);}
iii.	parameter that is not passed will be undefined.	function sum(a,b){}; sum(1);
iv.	if return right side is empty on a function	function sum(a,b){……. return }
v.	if property that doesn't exist on an object	const obj={a:"b"}; console.log(obj.c);
vi.	access array elements outside of the index range	const arr=[1,2]; console.log(arr[10]);
vii.	deleting an element inside an array and trying to search it by it's index number	const arr=[1,2,3,4]; delete arr[2]; console.log(arr[2]);
viii.	set undefined directly in an variable.	const a= undefined. It should not used

21. It we want to show undefined to a variable we should avoid undefined and use to null

const a=null , b=[] ; typeof null=object & typeof undefined=undefined

22.

Truethy	Falsy
true	false
any number except 0	0
any string except empty string('')	empty string ('') abar ' ' maje space thakle true
'0' , 'false'	undefined, null, Nan
Object/empty Object{} / function	
empty function / array / empty array []	

23. Check Truthy → if(a) , if(!!a)

Check Falsy → if(!a), else

24. == vs === Always Recommended to use === in code

(===) → i. It check the value and the datatype of the value. If both are same then true.

(==) → It check just the value . In it: 1==true , 0==false , 3== '3' correct because it just checked the value. If the datatype are same for both value then it is not an issue , if not then it tries to convert one value datatype and check it.
The changing datatype process called (**type coercion/casting/conversion**)

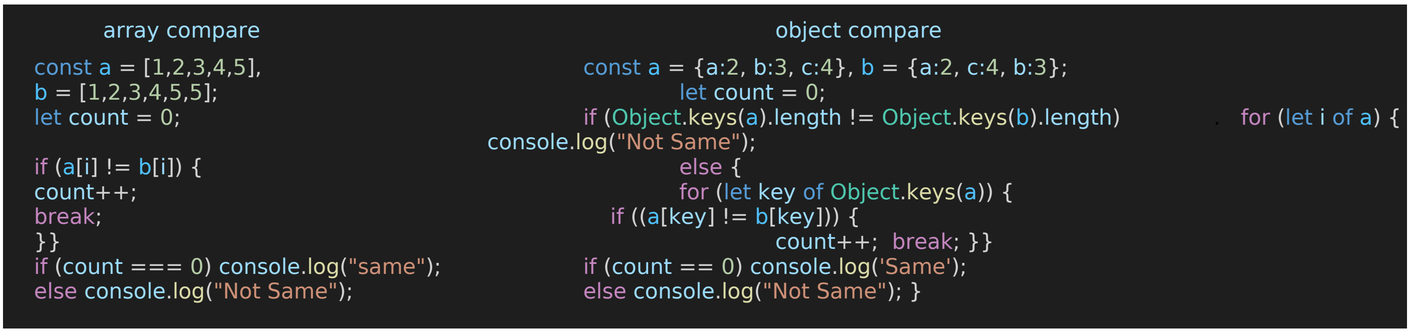
== / === → If we try to check primitive datatype then it only check the value.

But if we try to check non-primitive datatype it will check (value+reference) both.

Const a=[ ] , b=a; a===b (true) const a=[ ],b= [ ]; a===b (false)

25. Scope → Global , Function & Block scope

26. Array & Object Comparism



```
        array compare                                    object compare

const a = [1,2,3,4,5],                        const a = {a:2, b:3, c:4}, b = {a:2, c:4, b:3};
b = [1,2,3,4,5,5];                                    let count = 0;
let count = 0;                                if (Object.keys(a).length != Object.keys(b).length)          . for (let i of a) {
                                       console.log("Not Same");
if (a[i] != b[i]) {                                   else {
count++;                                              for (let key of Object.keys(a)) {
break;                                           if ((a[key] != b[key])) {
}}                                                           count++;  break; }}
if (count === 0) console.log("same");         if (count == 0) console.log('Same');
else console.log("Not Same");                 else console.log("Not Same"); }
```

27. Hoisting means function ot variable comes to top and make themselves as global

Let , Const → follow the blockquote rules that the cannot get back from the block quote

Var → Whereever we use var it goes to the top and make it to a global variable

function () {…..} → this type of function goes to the top using hoisting. We can call the function from anywhere in the JS file. (top / bottom)

const mahi = function() {…} → this type of function doesn't fullow the hoinsting . we can call

const mahi = a => {…} → then only from the below of the function

28. Closer → ফাংশন যদি অন্য একটা ফাংশনকে return করে এবং ২য় ফাংশনের মাঝে যদি ১ম ফাংশনের ক◌োন◌ো variable থাকে তবেই তা closer হিসেবে কাজ করবে।
যদি ক◌োন◌ো variable না থাকে তাহলে closer ফাংশন হিসেবে কাজ করবে না।

```
function counting() {
     let count=0;
     return function(){
         return ++count;}
const a= counting(),b=counting();
console.log(a());      // 1
console.log(a());      // 2
console.log(b());      // 1
console.log(b());      // 2
```

এখানে a,b উভয়ই ফাংশন। কারণ তাদের মাঝে ফাংশন রিটার্ন করা হয়েছে। তাদের প্রত্যেকের মানে ++count=1 থেকে আলাদা আলাদা ভাবে শুরু হবে। একটার মান 1 থেকে শুরু হলে তাকে n বার কল করা হলে তাহলে মান ও n সংখ্যক হবে তাই বলে এক variable এর মানের শেষ হতে অন্য variable এর মান শুরু হবে না। আমার নতুন থেকে শুরু হবে।

29. CallBack Function ➔ ekta function ke call korar somoy argument hisebe jodi arekta function diye dei ar 1st function theke pe parameter neya hoy ta diye jodi abar 2nd function call kora hoy tahole.

```
function greeting(anotherFunc,name) { anotherFunc(name); }
function greetMorning(name) { console.log("Good Morning",name);
greeting(greetMorning, 'Mashud');
```

Practical Use: Use in EventHandler

```
function submitHandler(){console.log(1);)
document.getElementById('id_name").addEventListener('click',submitHandler);
```

30. Function er bitore **arguments** use kore deka jay jototi parameter patanu hoyeche ta sob.

Arguments in function → object type array → **[...arguments]** → convertin it into an array

```
Function sum() {
    const args = [...arguments];
    console.log(args);
    const sum = args.reduce(
        (previous, newValue) => previous + newValue, 0
    );
    return sum;}
console.log(sum(1, 2, 3, 4, 5, 6));
console.log(sum.length)
```

31. Not-Primitive Datatype ➔

- Equal korle eder reference same theke jay . Tokon jodi eder 1 item ke partially change korle both item change hoy but full Object/array change korle kebol 1 ta change hoy.
- Primitive datatype er ketre function er bitore variable er value change korle baire change hoy na but not ptimitive er ketre function er bitore change korle baire change hoye jay

```
let a = 2,  b = 5;

const sum = (a, b) => { a = 20;}
console.log(a);      // a=2
sum(a, b);
console.log(a);  // a=2                    function er baire value change hoyni



const arr1 = { name: true }, arr2 = { name: false };
const arraycustomize = (a1, a2) => {
    a1.name = false; a2.name = true; }
console.log(arr1, arr2);      // true , false
arraycustomize(arr1, arr2);
console.log(arr1, arr2);      //false , true                    //Non-primitive e function er por value change
                                                                        hoye jay jodi partially changed hoy
```

DevTools

1. Elements ➔ Html code, id class search kora jay
    - Event Listener → oi particular jinis e eventHandler ase kina check kora jay. Thakle click Kore direct sources er code e jaua jay.
    - Accessibility → Ei part e special type people(color blond,blind,leaf,..) eder jonno kaj kora hoy
2. Sources → Ekane full folder je js,pic,css ja sob ase sob kisu deka jay
    - Add Folder to woks → sources e folder add korar maddome vsCode er moto direct Eikane code like save korle save hoye jay
    - Snippets → kisu code age thekei save kore raka jay jate (Ctrl+Enter) marle run hoy. Previous time e besi run hole o ekon temon use hoy na.
    - Call Stack → Je line/function er callStack dekbo oi function jekan theke asbe tar link show
    - Breakpoints → joto breakPoints add kora ase sob show korbe
    - Conditional BreakPoint → condition deye dewa hoy. Jodi condition true hoy tahole breakpoint kaj kore jemon ekta loop 0-100 porjonto . er maje 50 te jodi chai breakpoint dite Tahole conditional breakpoint e condition dile jokhon 50 asbe tokhon e kebol breakpoint kaj korbe.
    - Event Listener Breakpoints → ei part e huge type ase. Jemon jodi amra click kori tahole html er jodi click kora hoy tahole breakpoints kaj korbe.

3. Network → API related work.
4. Application → Ei part e Storage, cookies , catch niye kaj kora hoye thake

5. Stortcut for Devtool:
    - Ctrl+Shift+C → Open Elements in Devtool
    - Ctrl+F → Find Anything in specific page. Class kunje (.class_nm) , same id (#id_nm)
    - Ctrl+Shift+O → Find function, class, id
    - Ctrl+L → clear console
6. ShortCut for VsCode:
    - Ctrl+f → Find Anything in specific page
    - Ctrl+Shift+f → Find Anything in all page
    - Ctrl+G → For go to a specific line

Console API

1. console.assert(condition,string) === jodi condition false hoy tahole assertation failed show kore next string show korbe . true hole nothing
2. console.clear() / (Ctrl+L) === used for clearing the console
3. console.context() ===
4. Counting →
   - console.count() / counsole.count('mahi') === 1st ta diye independently counting start hobe. Ar maje jodi name dile name diye counting start hobe.
   - console.countReset() / console.countReset('mahi') === reseting the count and come back into 0.
5. Error / Warm →
   - console.error('It will show a error message with red background color');
   - console.warn('It will show a warning message that is not as restricted as error but the html file is warning you with a yollow color type background color');
6. Grouping →
   - console.group() / console.group(label) === grouping start kore
   - console.groupCollapsed() / console.groupCollapsed(label) === collapsed akare group start kore
   - console.gropEnd() / console.groupEnd(label) === grouping remove kore dey

   group er bitore kisu jodi amora likhe thaki tahole ta group er bitore ache denote korar jonno halka dan dike chepe thake. Jodi group end korar por abar likhe tahole ta normal style e thake . Jodi abar groupCollapsed tahole collapsed hoye thakbe & chaile expand kora jabe.

Ex.

```
const timeline1 = 'New York 2012';
const timeline2 = 'Camp Lehigh 1970';
console.group(timeline1);
console.info('Mind');
console.info('Time');
console.group(timeline2);
console.info('Space');
console.info('Extra Pym Particles');
console.groupEnd(timeline2);
console.groupEnd(timeline1);
```

Result:

```
New York
    Mind
    Time
    Camp Lehigh 1970
        Space
        Extra Pym Parsicles
```

7. console.info() / console.dir() === As same as console.log() but some minor difference.
8. console.memory() === show the memory status
9. console.trace() === Basically it's a tracking meathod. Trace use korar maddome last theke function Prothom e ki ki step e eseche ta show kore link soho. Link e click korle sources e niye jay.
10. TIMING →
    - console.time() / cl.time(lebel) === starts ta stopwatch in miliseconds
    - console.timeLog()/cl.timeLog(lebel) === we can see the miliseconds value after any given period by using this
    - console.timeEnd() / cl.log(lebel) === stopped the stopwatch and returns the final time

Times In JS

const time = new Date();

1. console.log(time); = = showing current time
2. console.log(time.getDay/…….) = particular;y day hour minute…… show kore
3. time.setDay(21)/.setFullYear(2032) = new day/year ta set kore thake
4. new Date(years,months,days,hours,minutes,seconds,miliseconds) eivabe man diye custom date run
   Ex. const timing1=new Date(2020,10,30,23,59,10,1000); console.log(timing1);
5. new Date('year-months-days') eivabe o custom date set kora jay
   Ex. const timing2 = new Date('2020-12-31") console.log(timing2);

Debugging Tips & Tricks

Debug Steps:

1. Check all things on webside and find errors and note it.
2. Check Console for error. Console e giye check korbo je kunu error show kore kina.
3. click on the link of the error (it will take you to the code)
4. EventHandler jesob jaygay thaker kotha sesob jaygat eventHandler ase kina check korbo Elements > Event Listener e giye.
5. BreakPoint add kore breck forward,function er bitore in hoye check korbo kaj kore kita . Kunu line Ses hoye gele oi line er value console.log e chole ase . console.log() kore dekbo value ase kina ba kun format e ase eisob.
6. Sources e breakpoint add korar pashapashi dorkar porle kunu function kutha theke asche ta jante chaile Call-Stack e giye check korbo je function kutha theke asche.
7. Kisu search diye kujte hole shortcut use kore kujbo.
8. look around for typo.