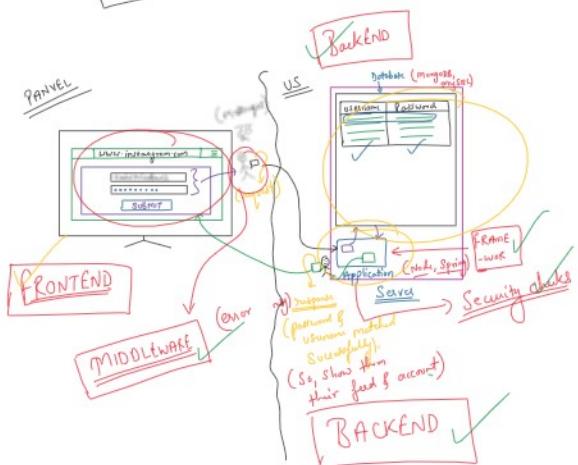
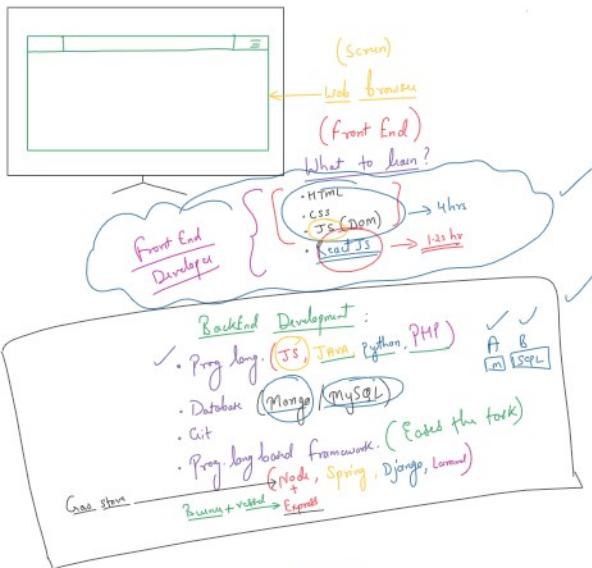


What is a prog. language?

A language used to communicate with the Computer.

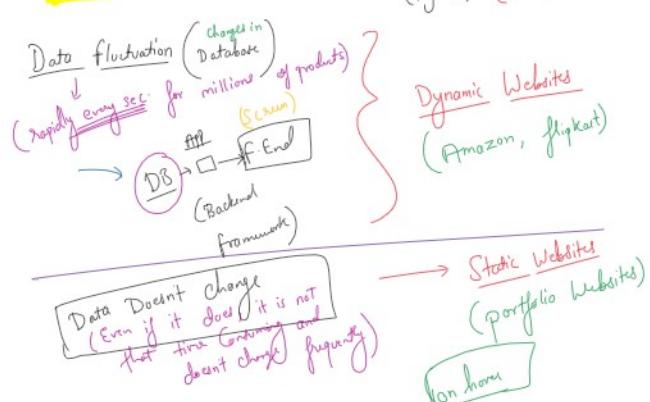


QUESTION (iii) → Recording
(Duration)

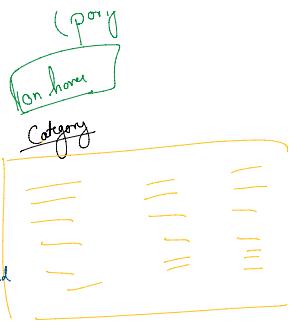
What after full Stack Development?

- ↳ 3 LPA to 24 LPA
- ↳ you can do Mobile App Dev ON YOUR OWN
- ↳ DSA (Matrix + PAMA) ✓ (Data Structures & Algo) ✓ (Security) (Ethical) ✓ (Artificial Intelligence) ✓ (Database)
- ↳ Create your final year projects
- ↳ Teach International & National Students
- ↳ FREELANCE

Practical → Theory

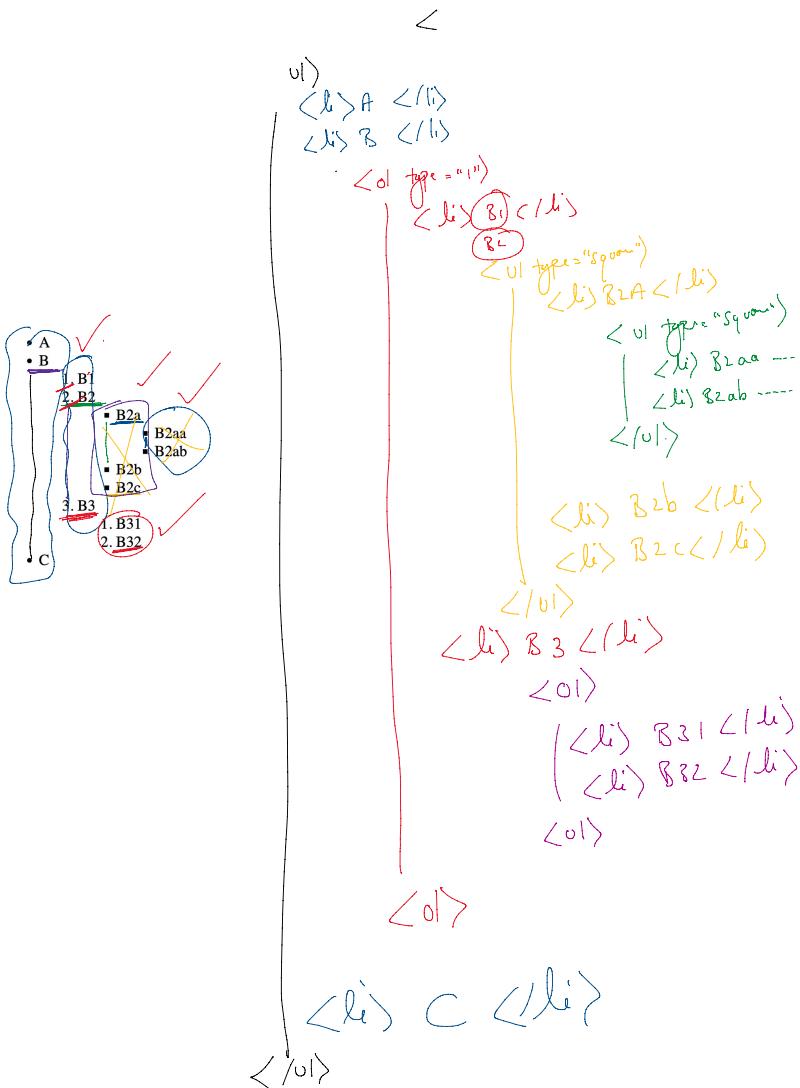


Data
(Even if it does "the same" and
doesn't change frequently)



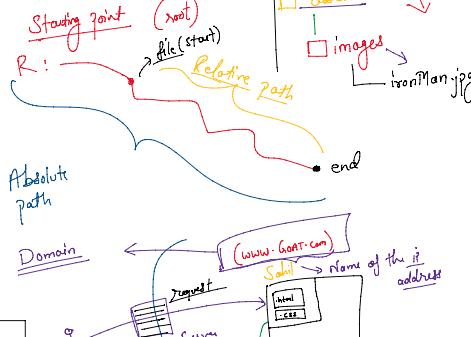
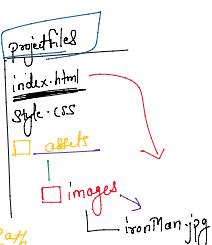
Frontend Technologies:

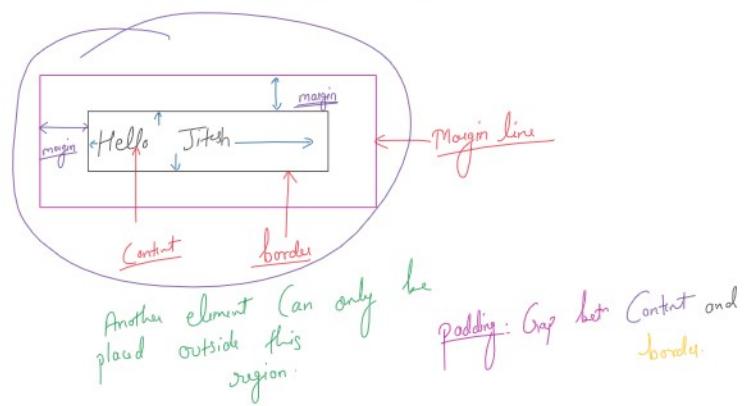
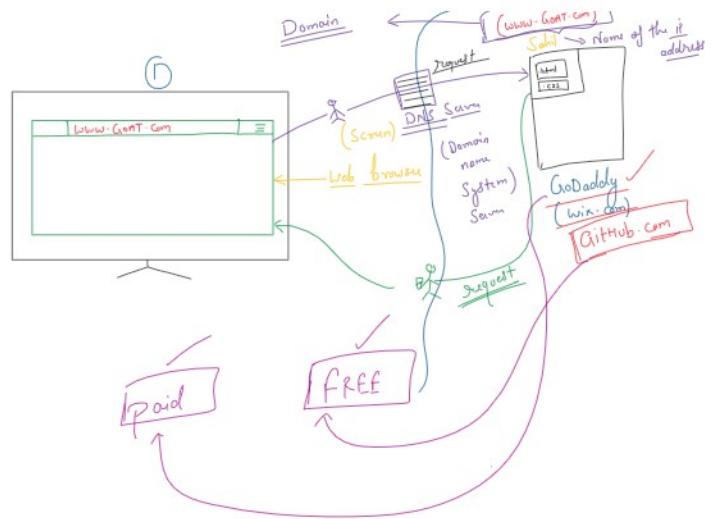
- HTML → Skeleton (โครงสร้าง)
- CSS → muscles & skin (กล้ามเนื้อและผิว)
- JS → Brain
- React JS → Eases the Work of frontend Development.



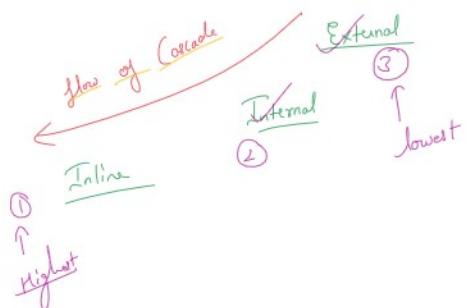
* ~~Image tag~~ *

 → self-closing tag

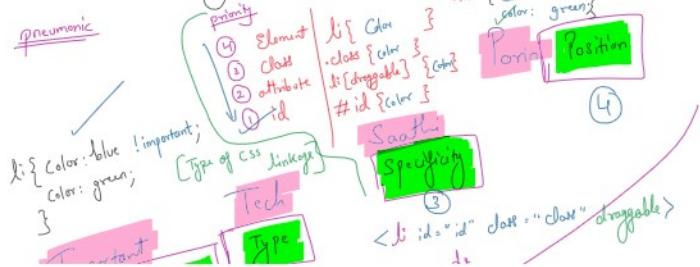


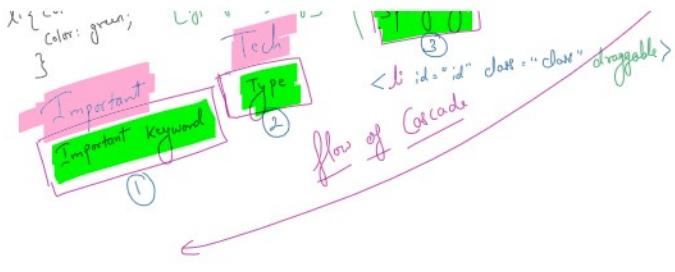


* Rules Applicable to Elements in CSS.

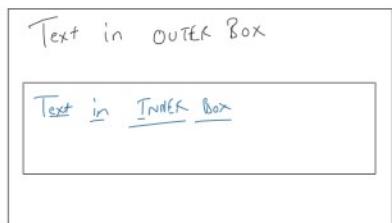


* Factors Determining the priority of Accompanied Flow





★ Combining Selectors



<div>

<h1 id="heading" class="head" draggable="true"> Hello </h1>

<h1 id="heading" class="head" draggable="false"> Hello Guys </h1>

Selector: (Chaining of Selectors)

h1#heading.head[draggable = "false"]

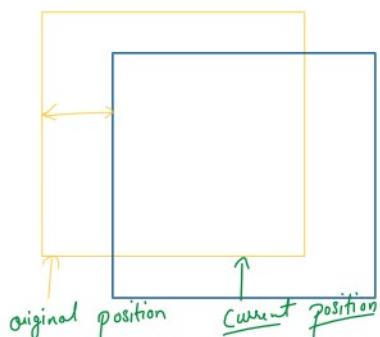
~~h1#head h1#~~

</div>

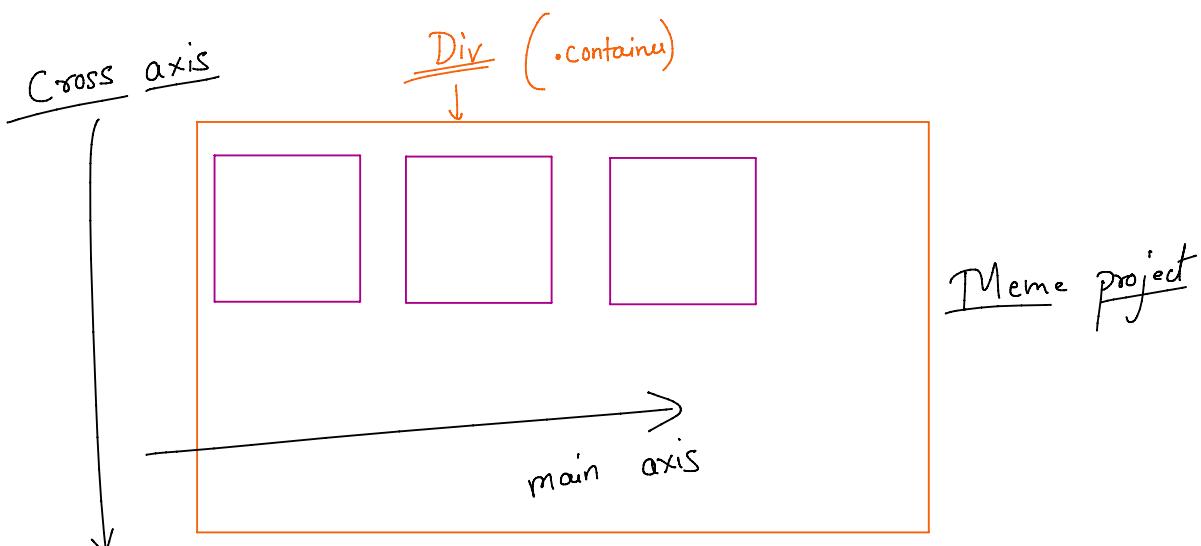
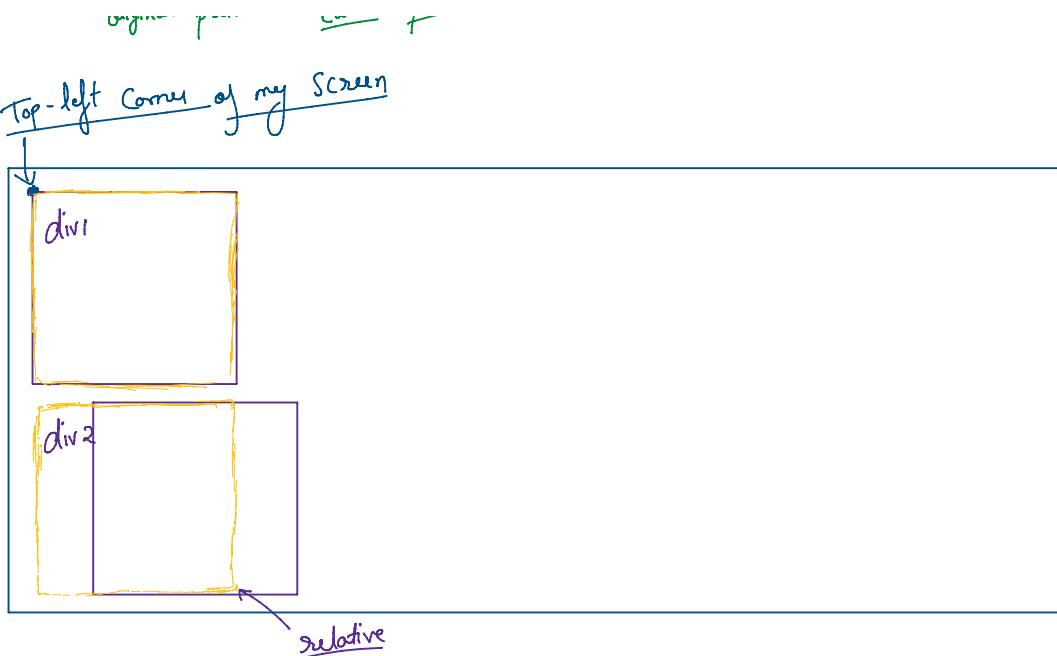
div h1#heading.head [draggable = "false"]

{

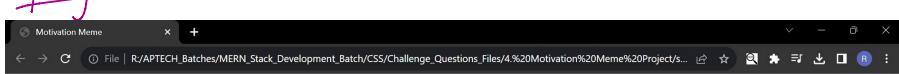
}

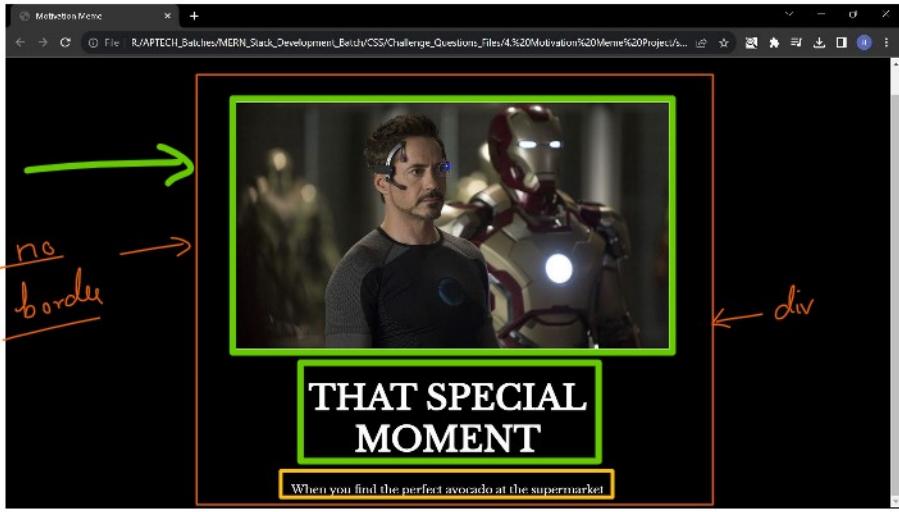


Top-left corner of my screen



★ Meme Project

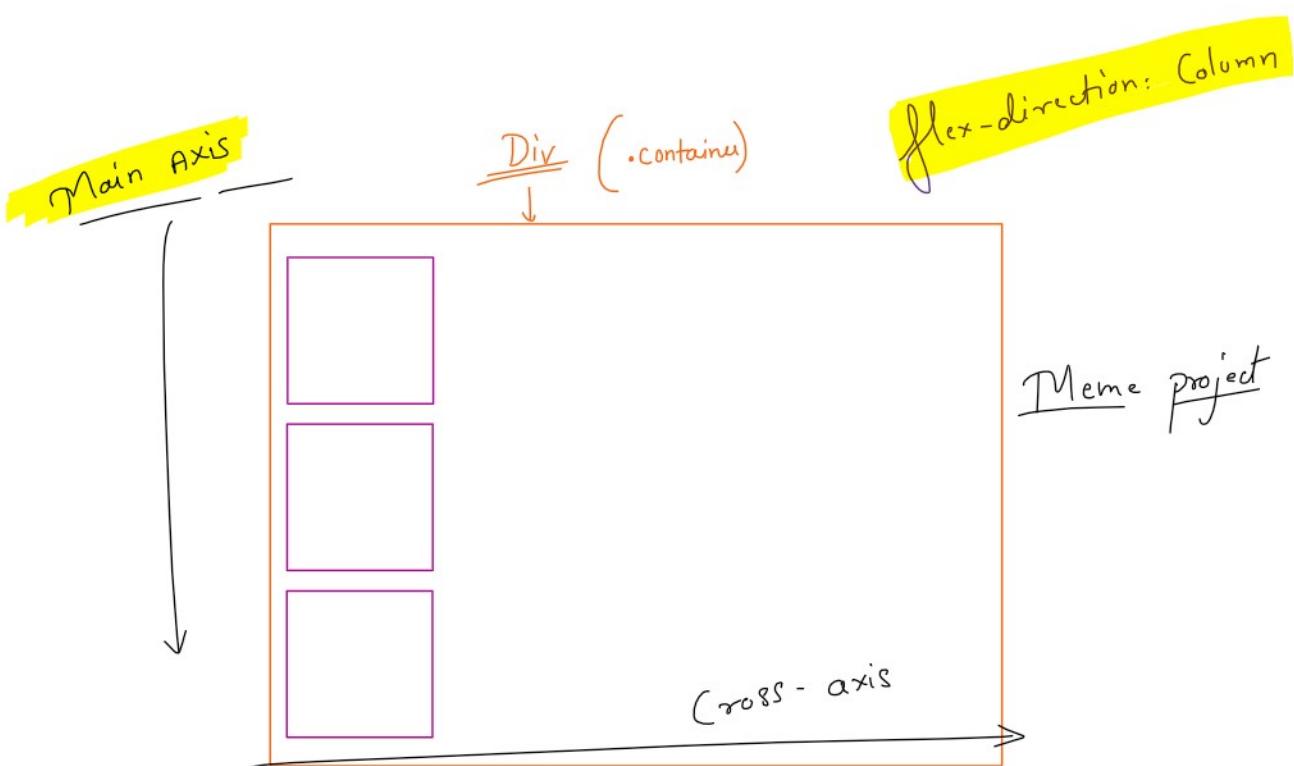
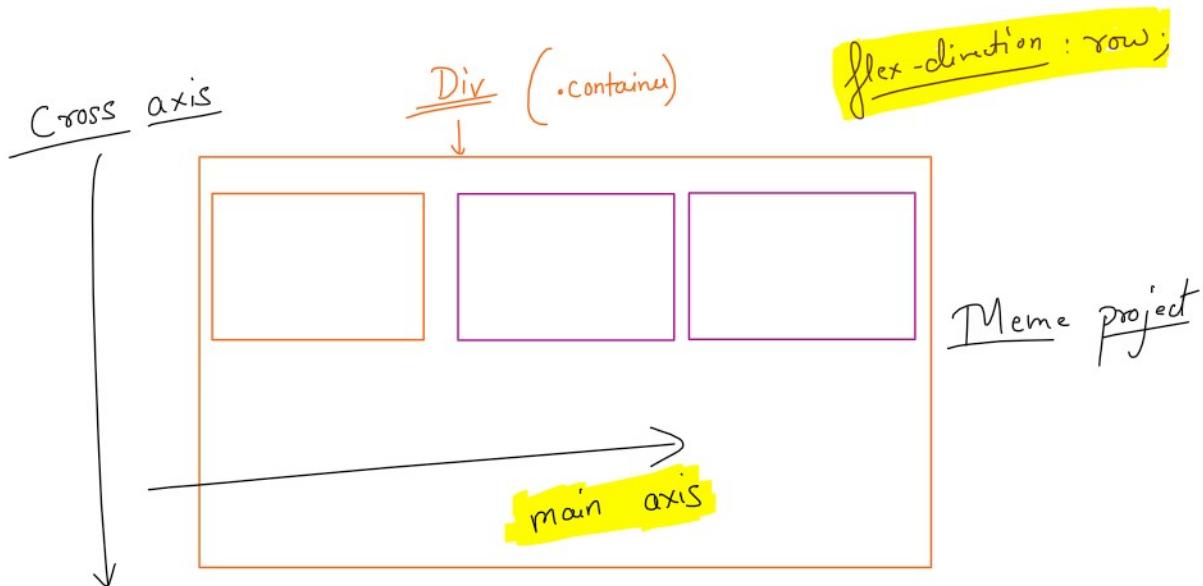




```

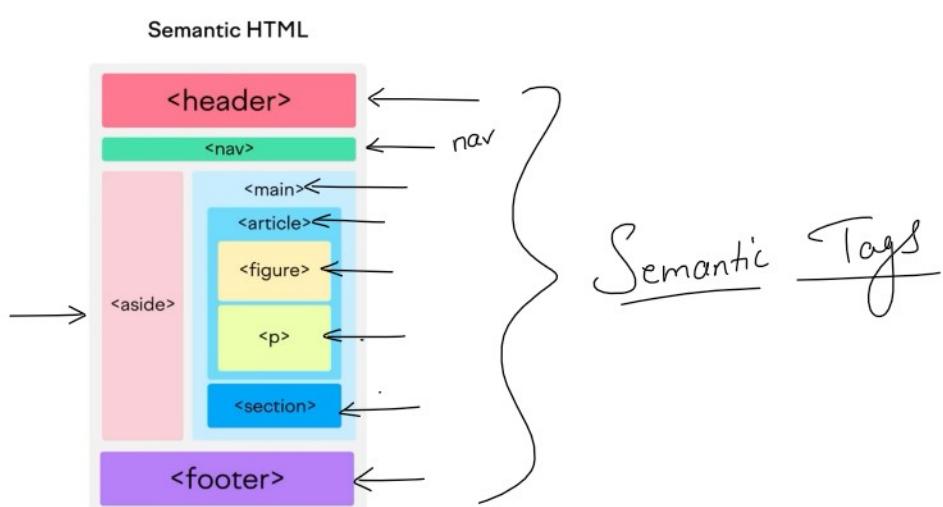
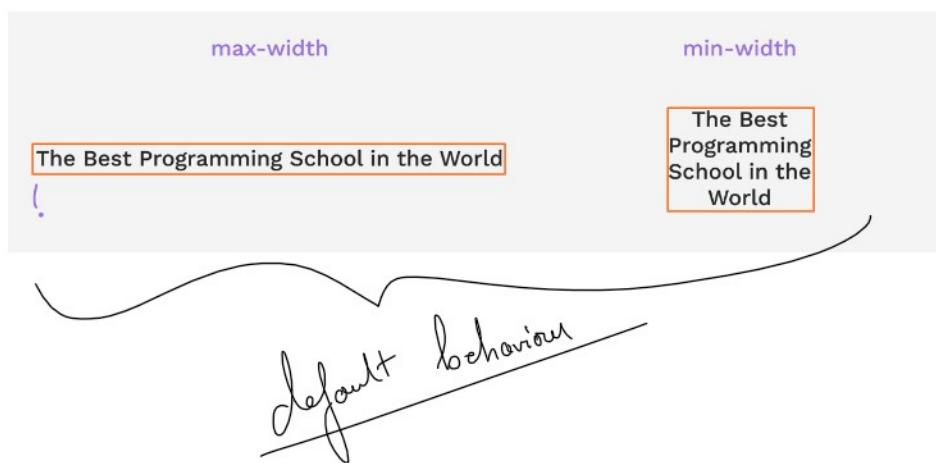
<div class = "container">
  <div class = "image">
    <img src = "" />
  </div>
  <div class = "name">
    <h1> _____ </h1>
  </div>
  <div class = "statement">
    <p> _____ </p>
  </div>
</div>

```



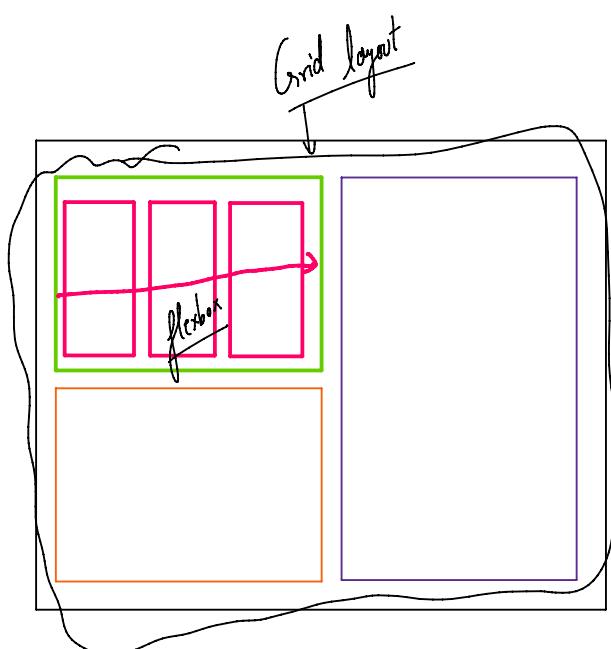
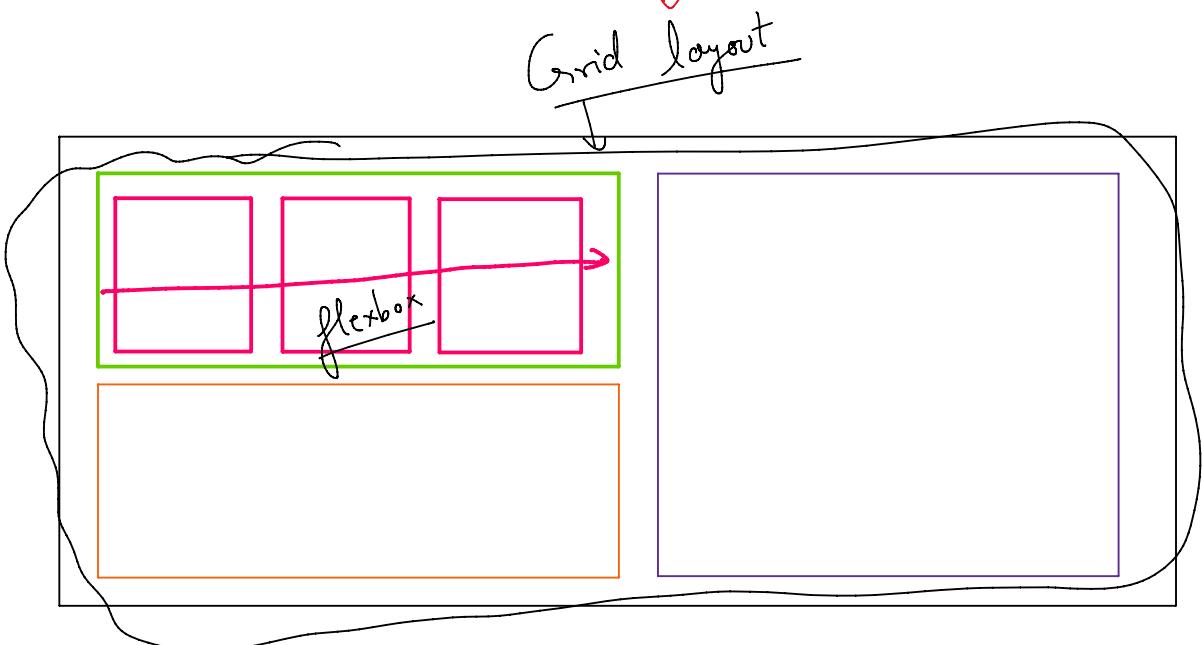
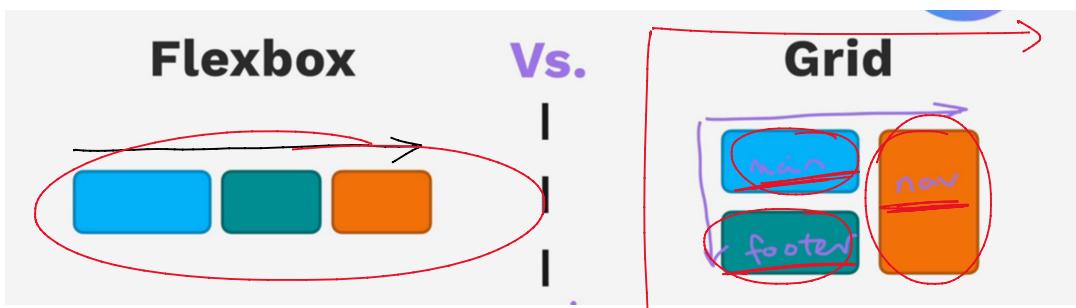


flex-basis: Possible expansion of flex items along the **Main axis**.
 (this property shall be applied to flex-items (child elements)).



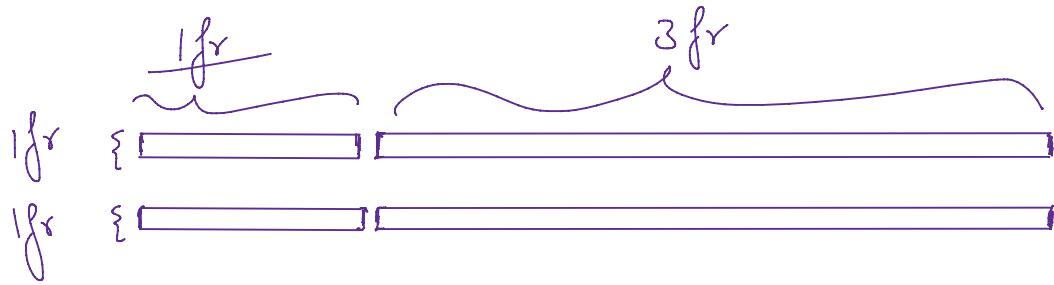
Flexbox

Grid



1fr 1fr

1fr 1fr



* Why 3 files?

HTML

- Skeleton

CSS

- Skin & muscles (Beautification)
(make-up)

JS

- Brain (Senses enable)

Hyper-text

Markup

language

(HTML)

(Every browser)

(a text which
can do
the work of
linking)

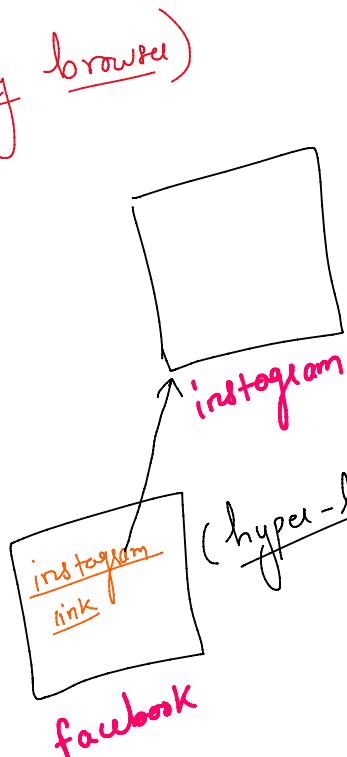
Sam Rahul

Rahul saw a tiger in the wild forest.

<p>



image tag
paragraph

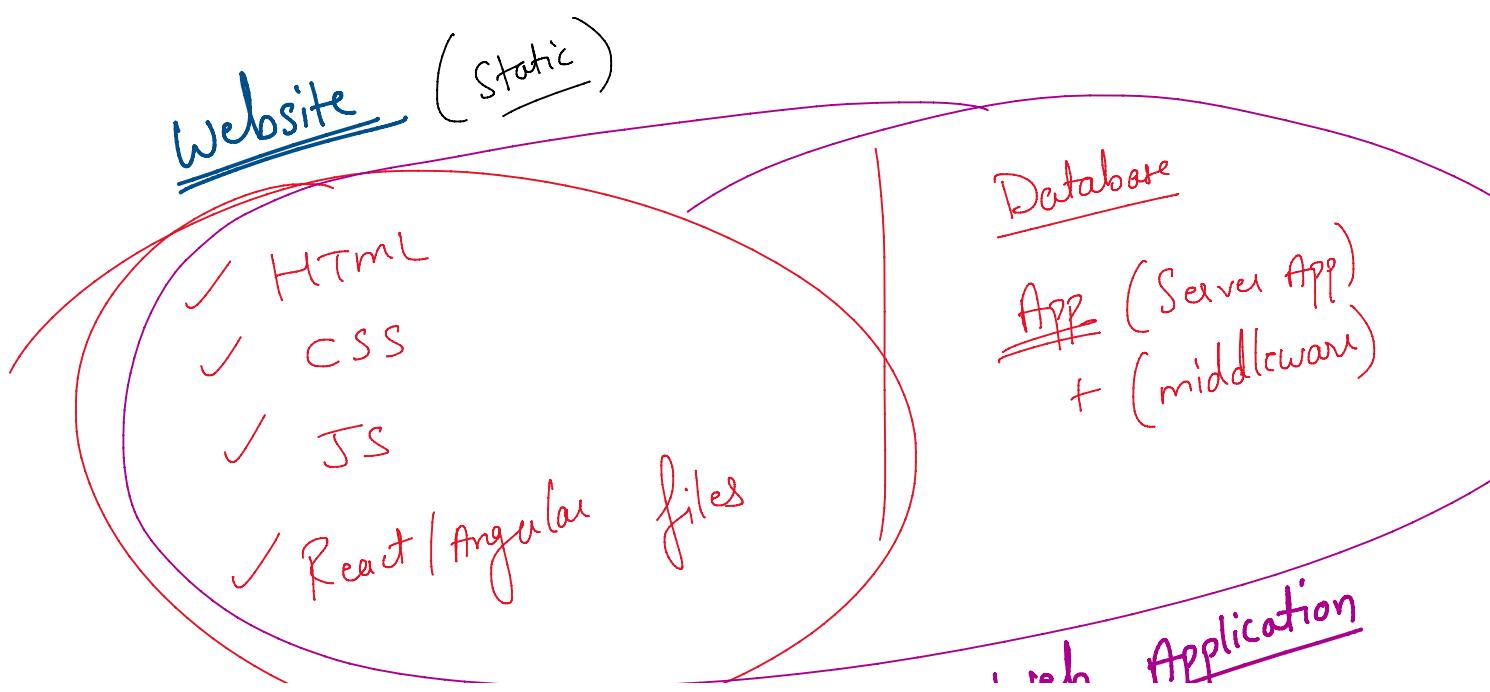
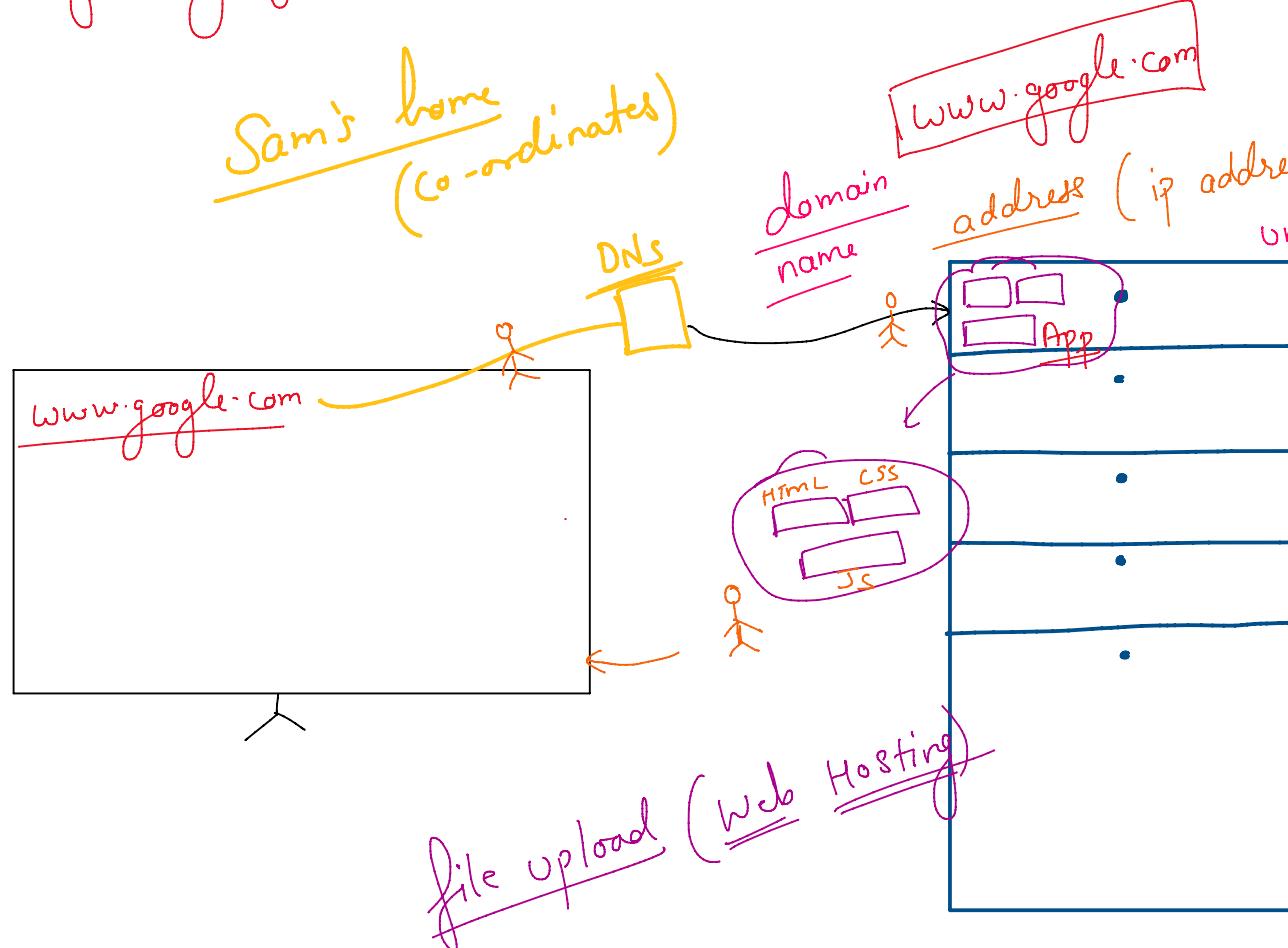


* HTML Code Structure *

<tag> Content </tag>

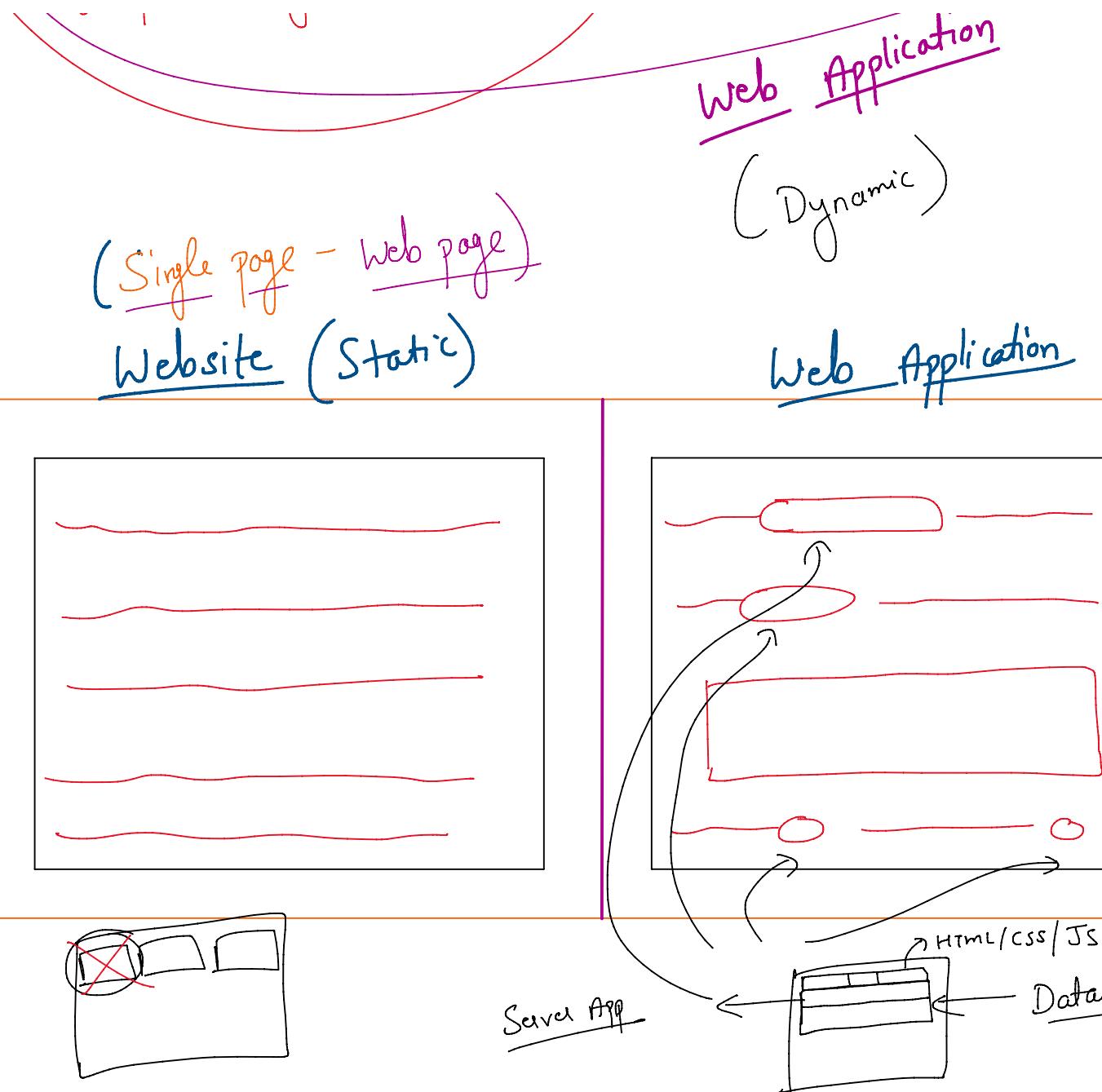
link)

<tag> Content </tag>
<tag> </tag>
<Self Closing tags> />



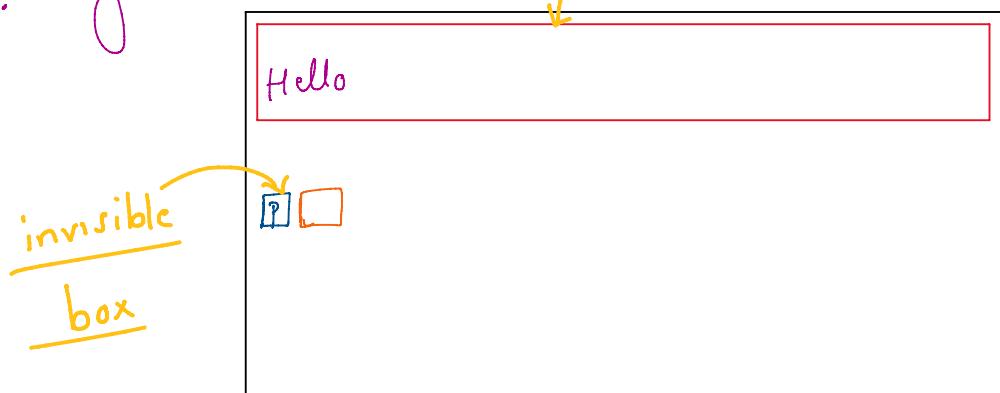
88)
nique



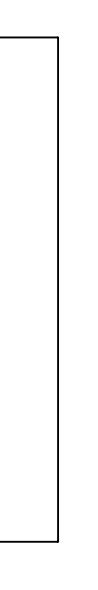


Box Model

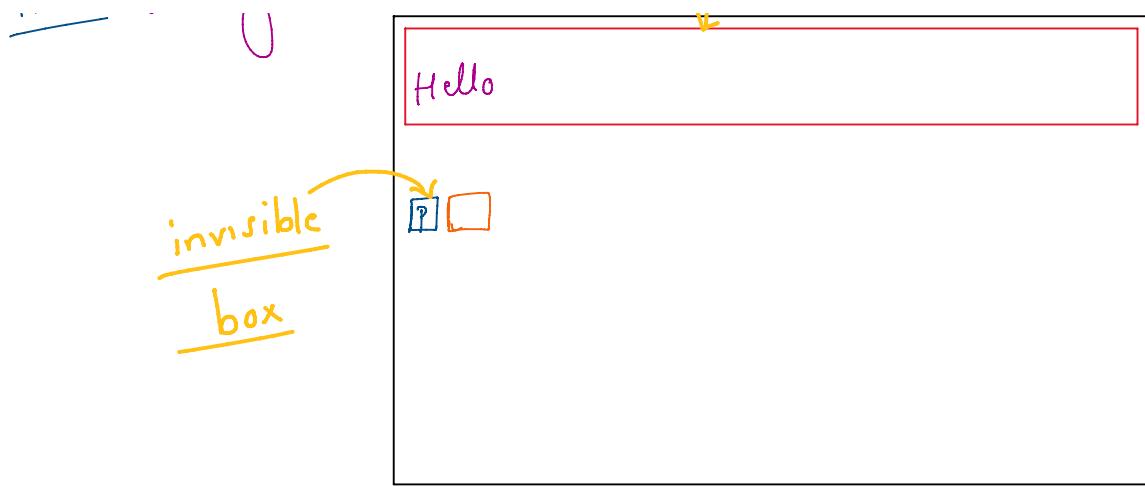
- Block Level Elements: Entire width
- inline: only as much width as needed



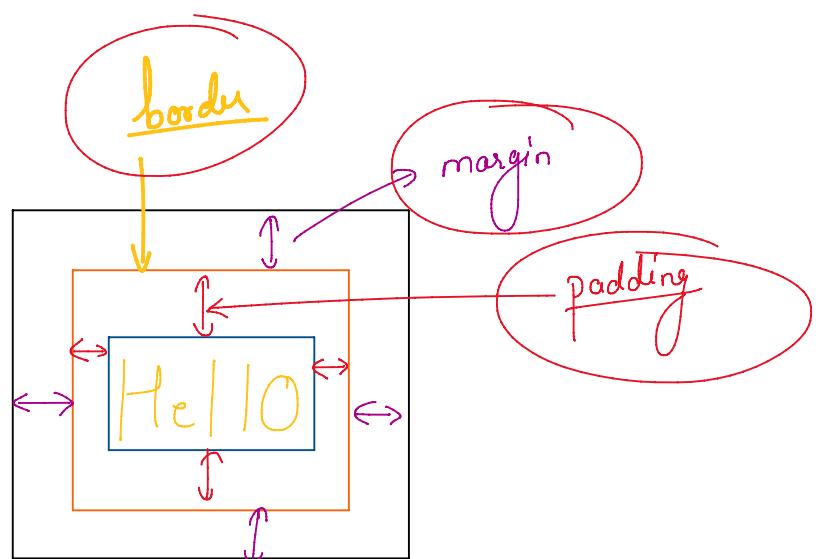
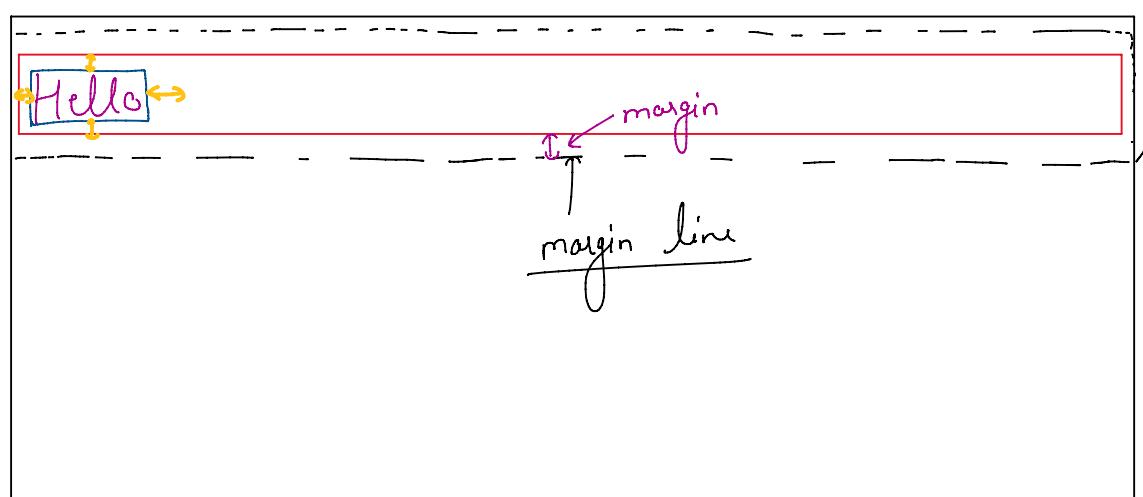
(Dynamic)

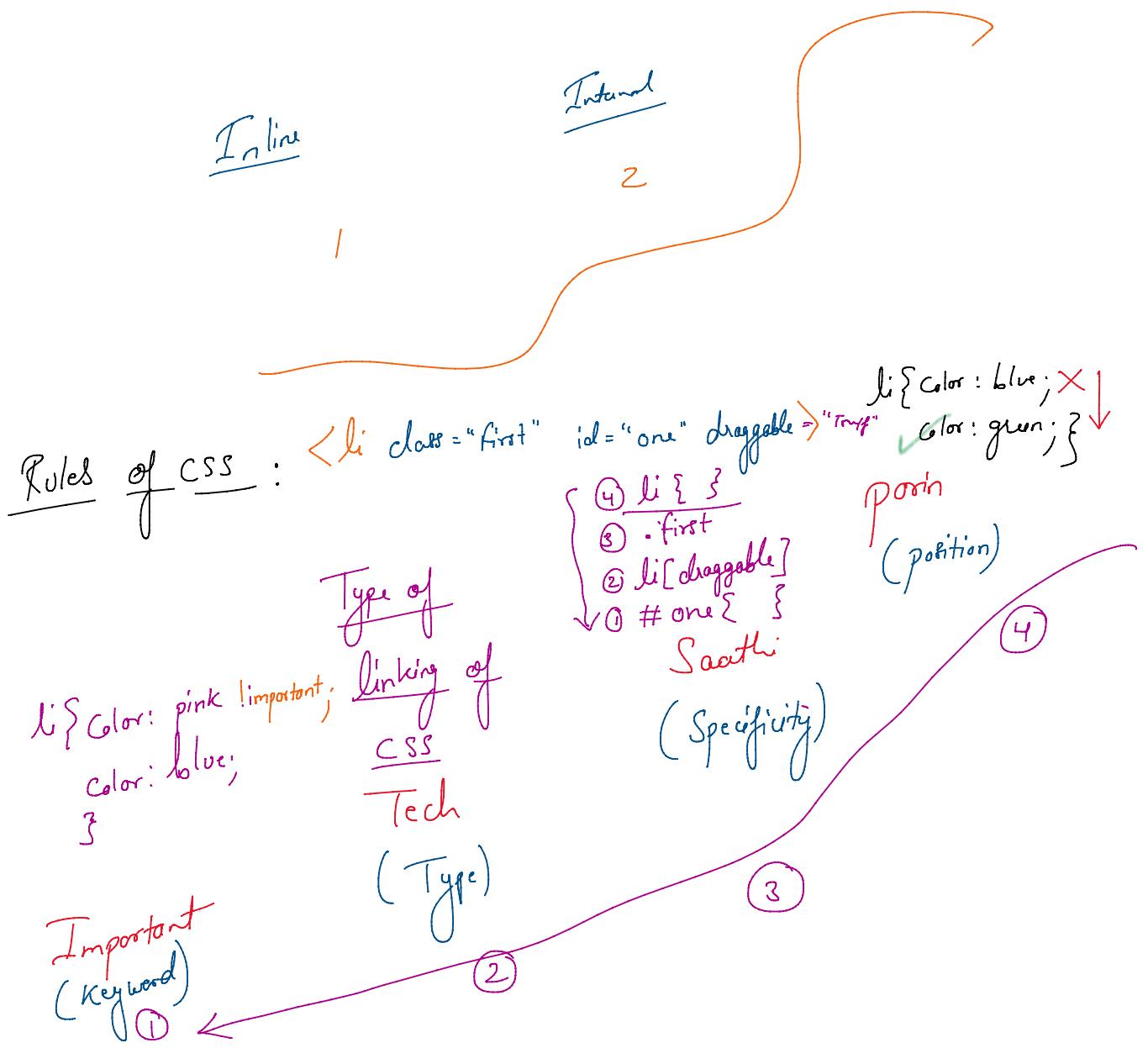
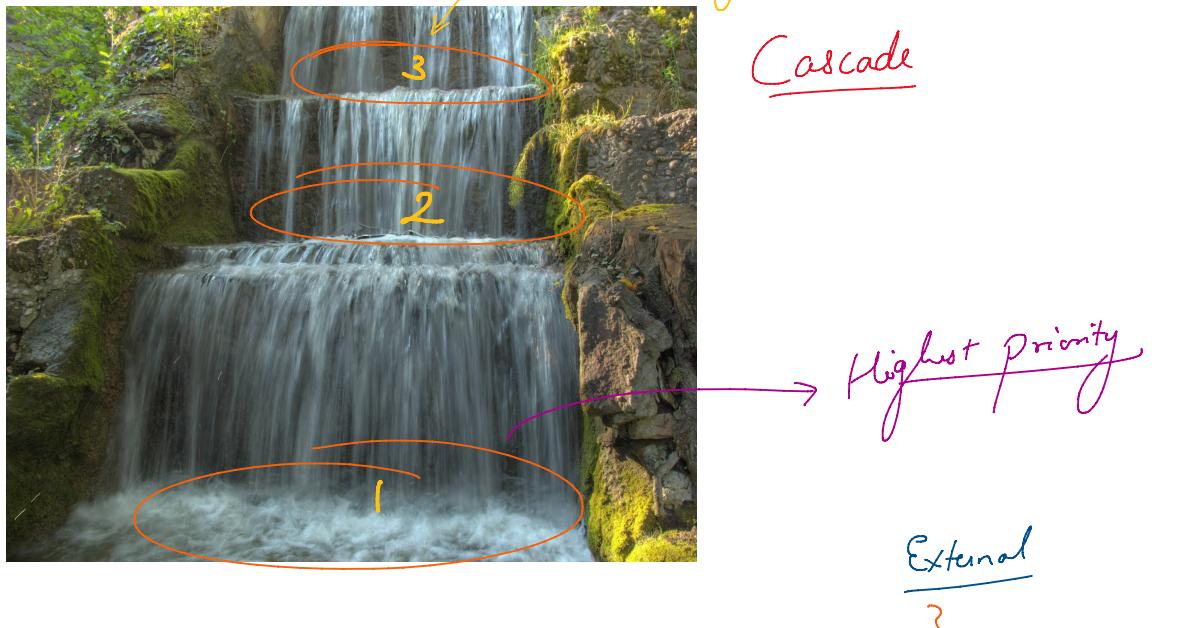


bote



→ - padding






```

index.html
<p>Yellow Text</p>
<div class="box inner-box">
  <p>White Text</p>
</div>
<h1>
  <div class="inner-box">
    <h1>Yellow Text</h1>
    <p>White Text</p>
  </div>
</h1>

```

```

styles.css
.inner-box h1 {
  color: yellow;
}
.inner-box p {
  color: white;
}

```

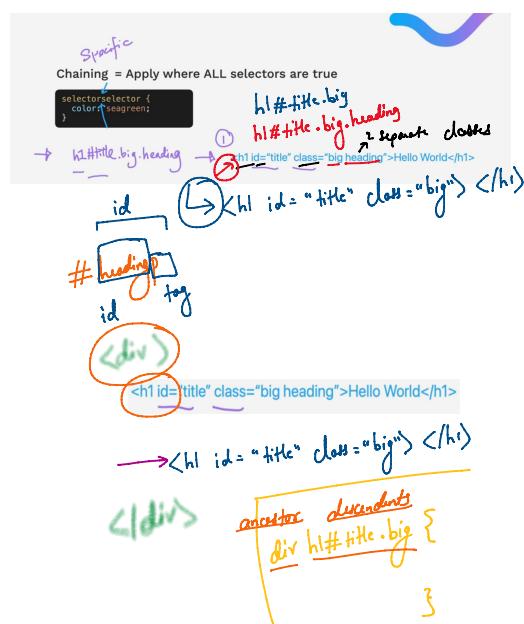
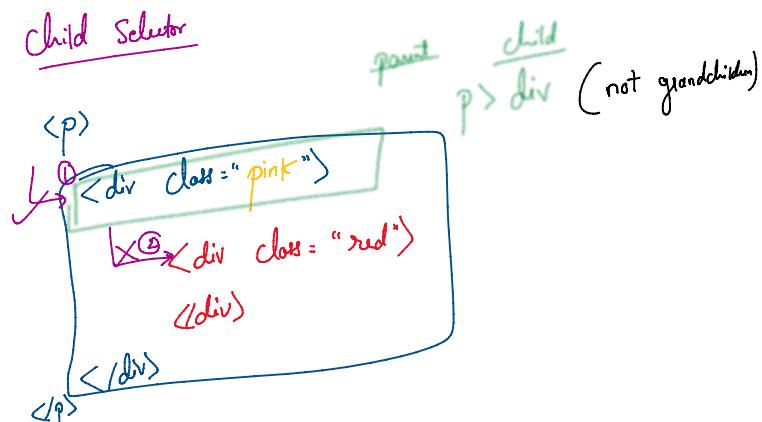
The App Brewery

www.appbrewery.com

Apply to both

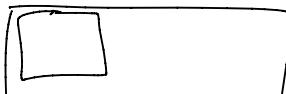
Selector₁, Selector₂ {

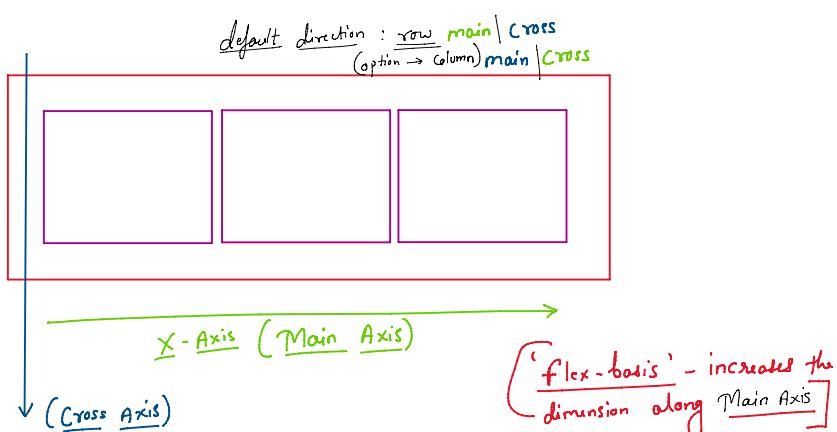
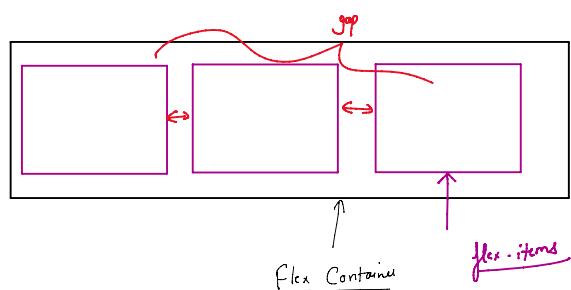
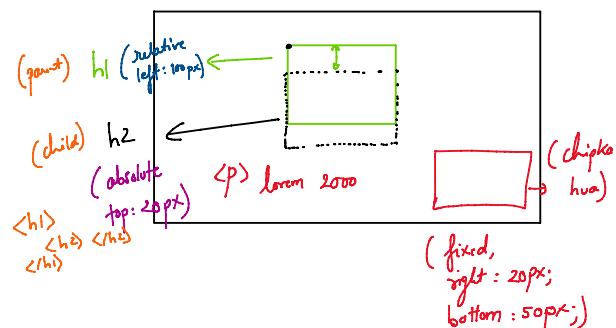
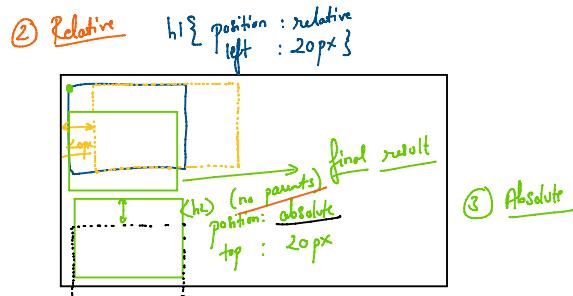
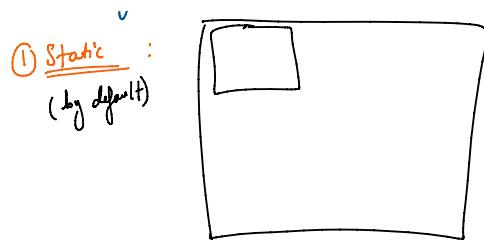
}



★ positioning (position)

① Static :
, by default)



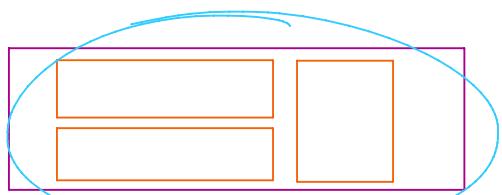
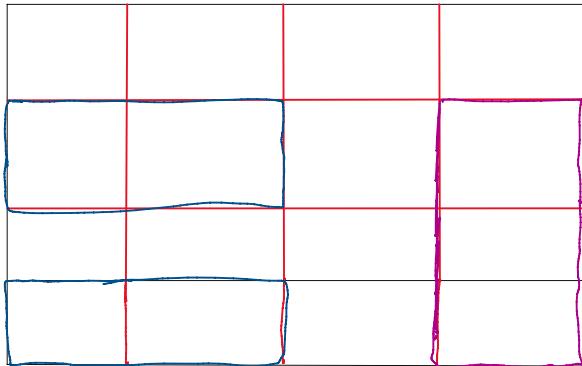


flex Sizing (Inference)

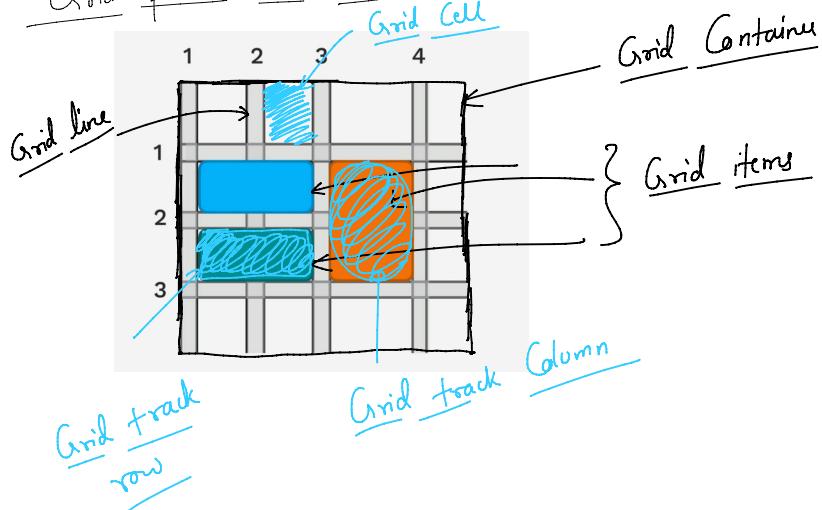
$\text{initial width} < \text{width} < \text{flex-basis} < \text{max-width/min-width}$

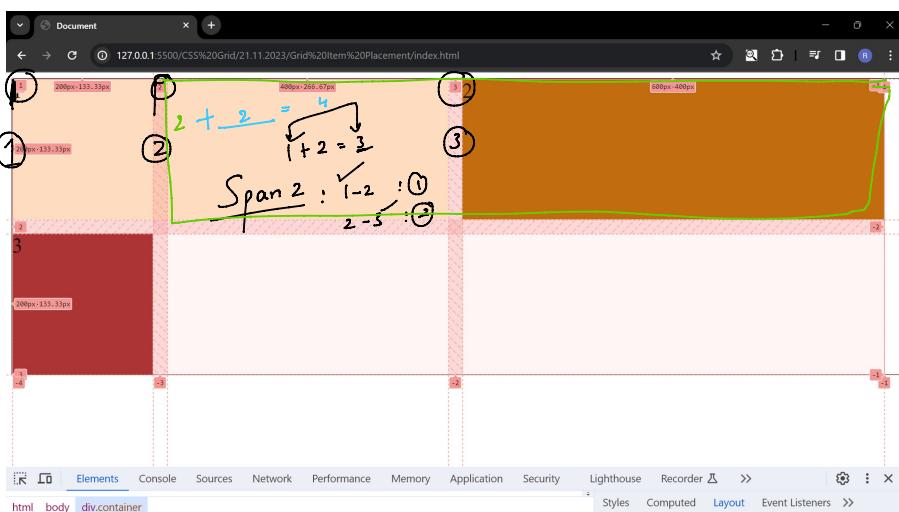
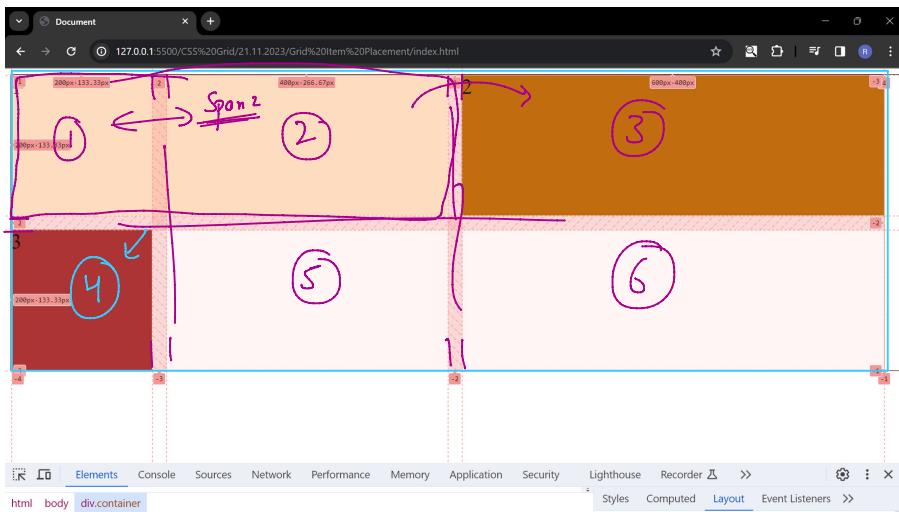
$\frac{\text{Content-width}}{\text{(min/max)}} < \text{width} < \text{flex-basis} < \frac{\text{max-width}}{\text{min-width}}$

Grid Item placement :



Grid provided us with this :





* Bootstrap

- css framework (Grid) (Pre-built Elements) (Copy-paste)

- Little bit Customizable (Not as much as raw CSS)

- Quick Way of Creating Simple UI

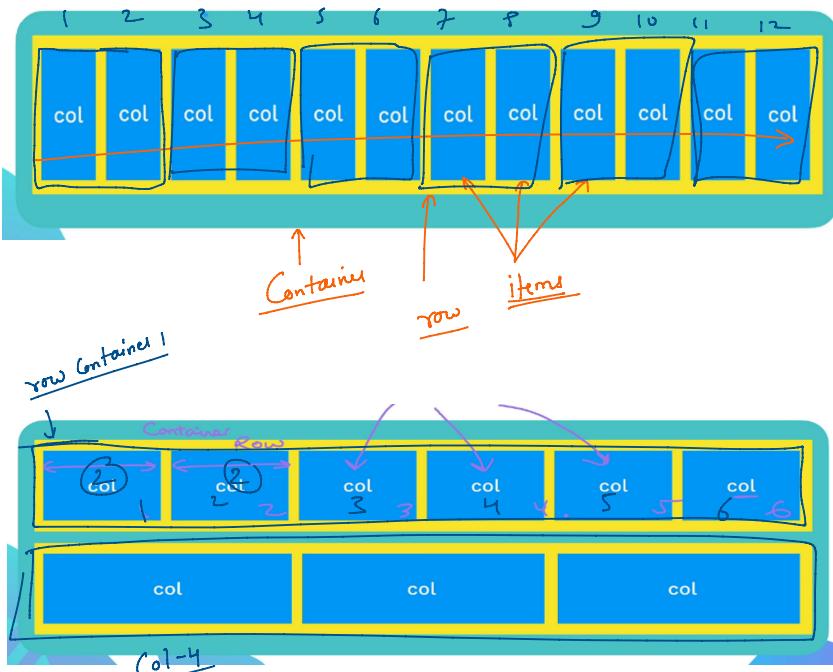
- Quick Way of Creating Simple UI
- Complex Layouts Can't be Created Easily

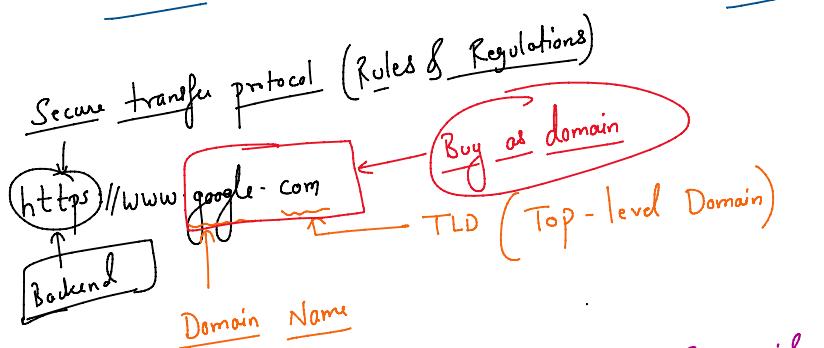
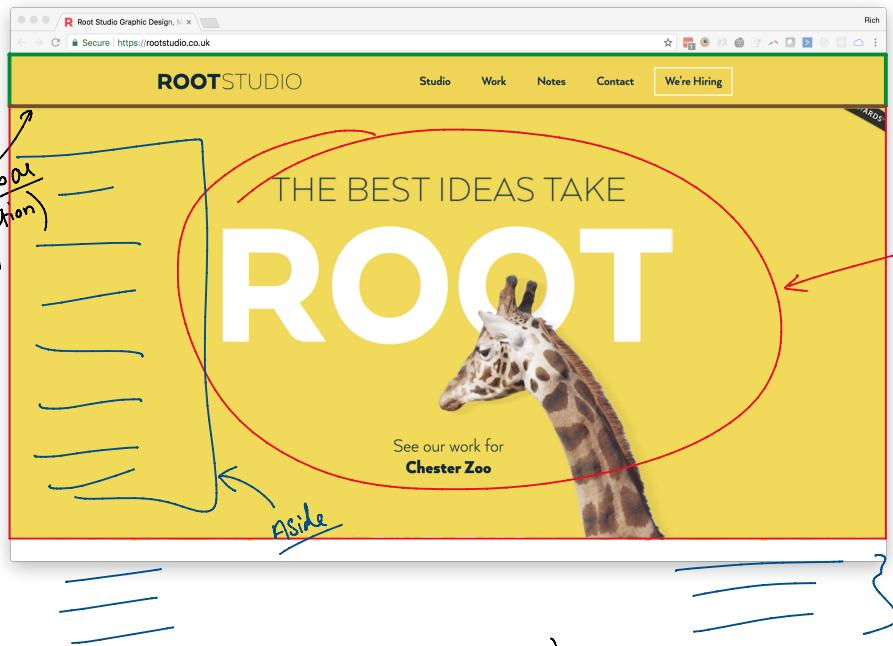
- How does Bootstrap Work?

CDN (Content Delivery Network) Npm package (install)



* Contains :

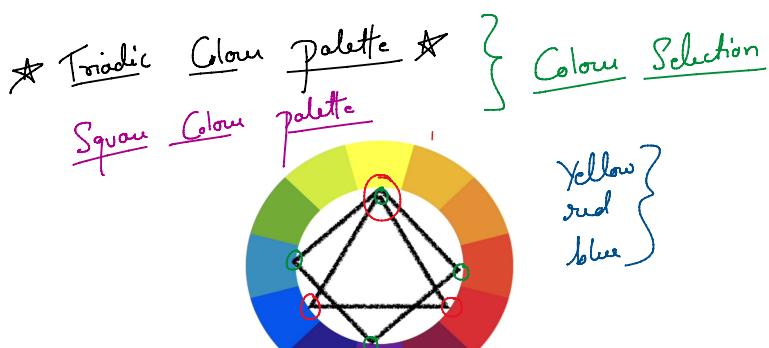


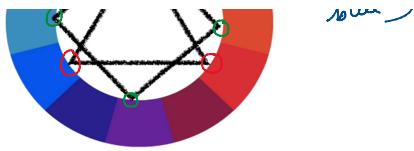


E.g. http://www.akasa.com → Commercial
 https://akasa.in → India

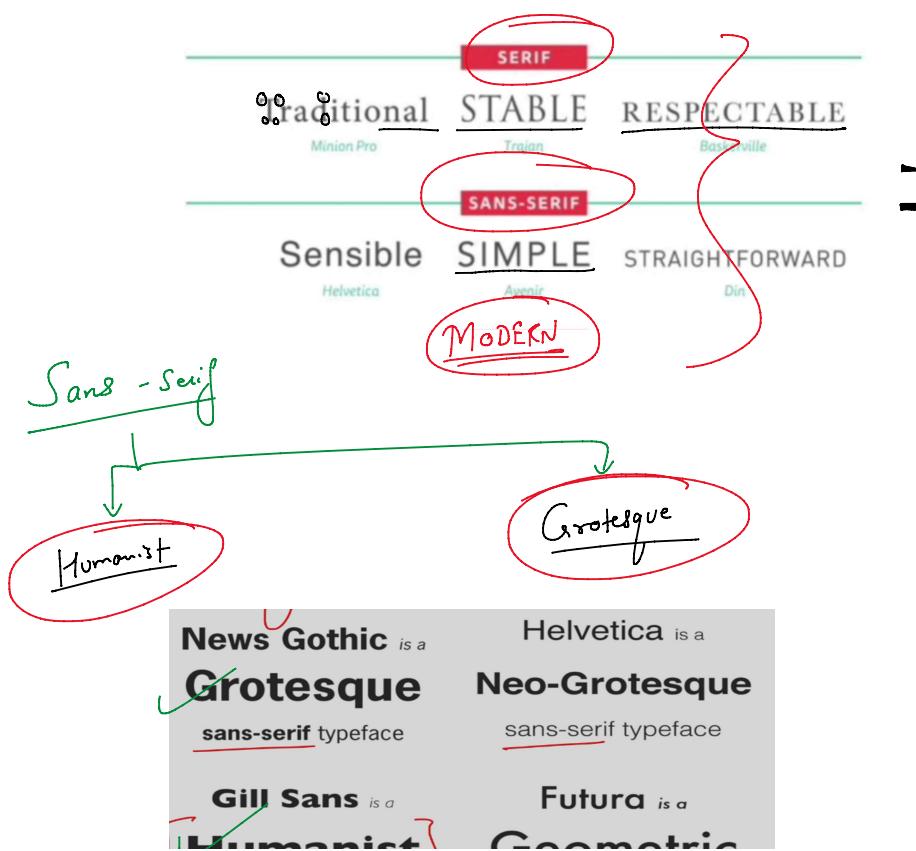
paid
 Security (certificate)

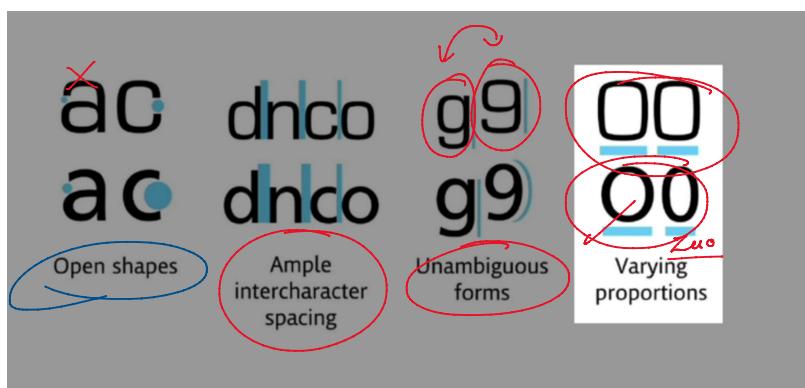
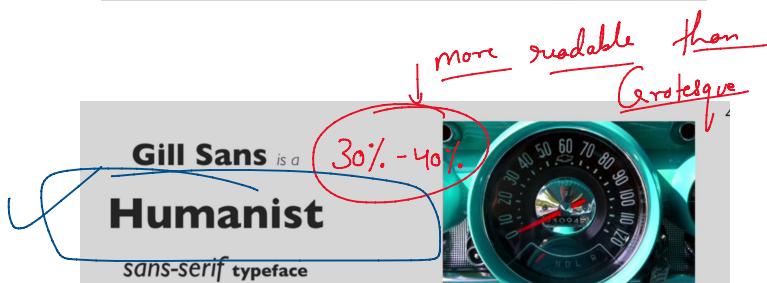
- .uk
- .au
- .gov
- .org





★ Typography ★





Using lots of fonts can make for a design that is cluttered, overcomplicated, AND JUST NOT VERY NICE

Don't use more than 2 font types
 But if you just use a small selection, you can keep your design cleaner, clearer and just much easier to digest

Never use similar font styles for Heading & Sub-Heading

Similar



Mood



Time Fra

Contrast



Serif-ness



Weights

Similar



Mood



Time Era

Contrast



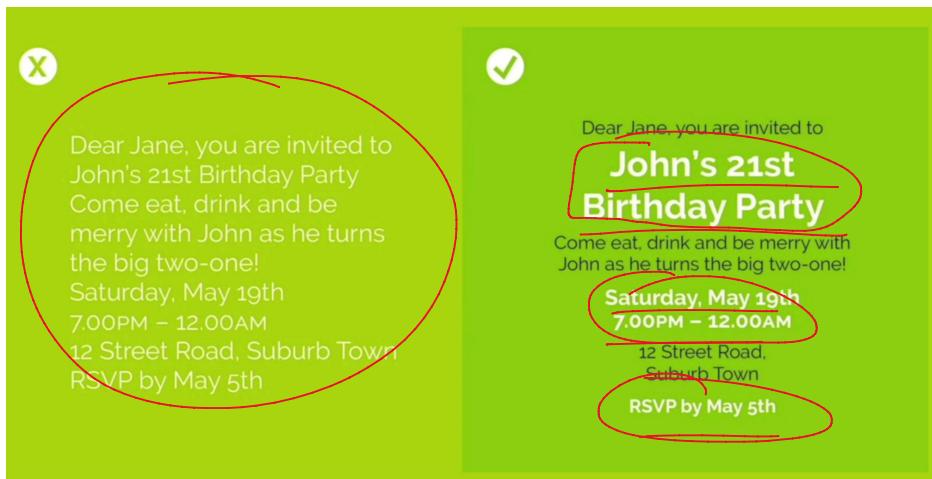
Serif-ness



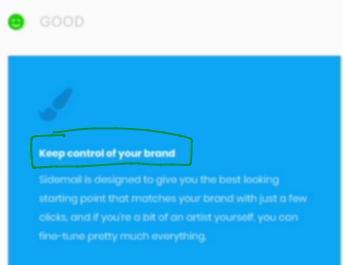
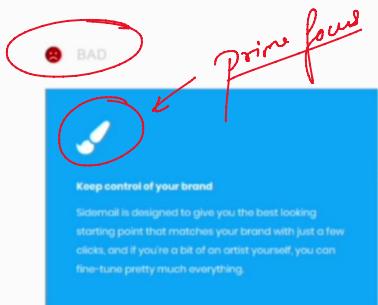
Weights

* Attention Grabbing Techniques by learning about Human Psychology

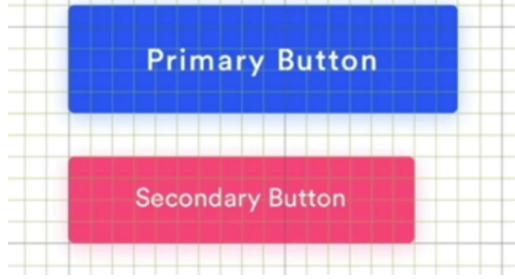
(Z path traced by Humans)



Colour

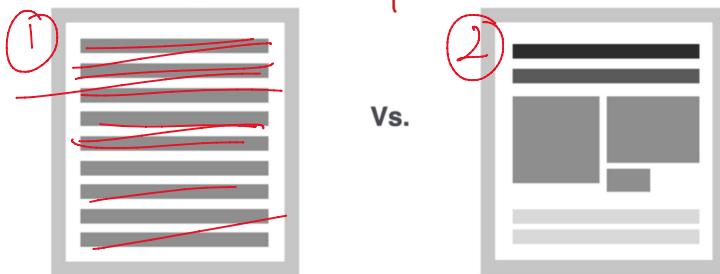


Size Matters!! (of buttons)



2. Layout

[Text per line]

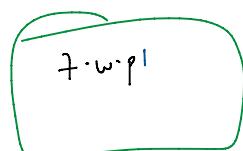
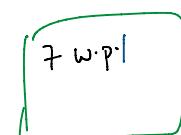


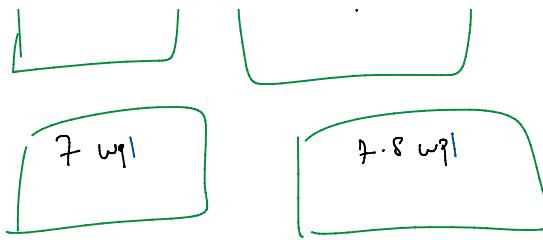
Vs.

Matters a lot.

The image shows a Wikipedia page titled "Typography". The sidebar contains links like "Etymology", "History", "Scope", "Text techniques", "Display graphics", "See also", "References", and "External links". The main content area has a large heading "TYPOGRAPHY" and several paragraphs of text. To the right, there's an image of a Trajan's Column inscription and another image of movable type being assembled. Handwritten red text on the left side of the image says "too long of a line to keep reading" and "BAD DESIGN".

Tinta portas



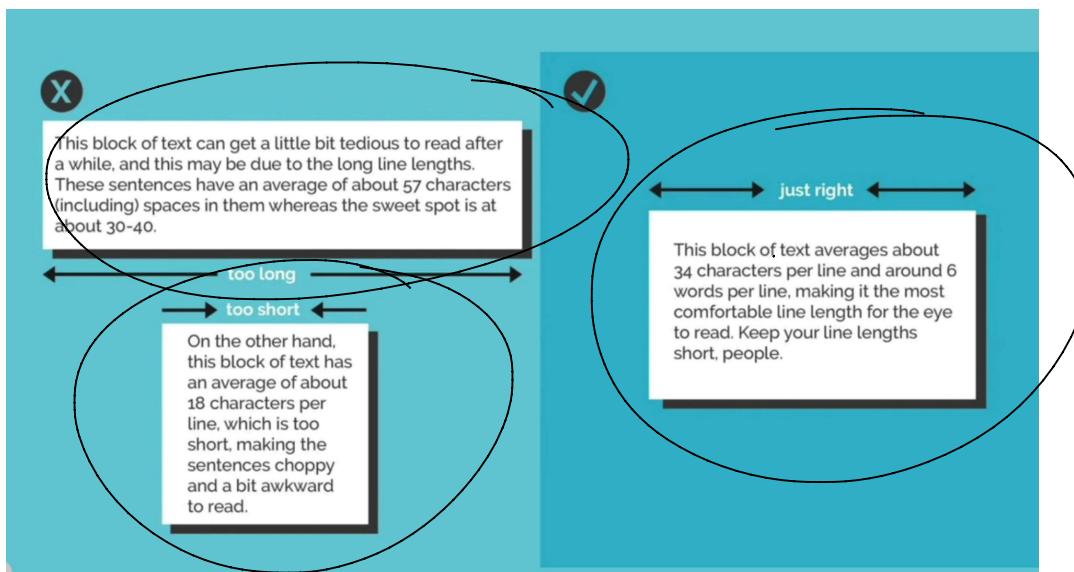


Words per line :

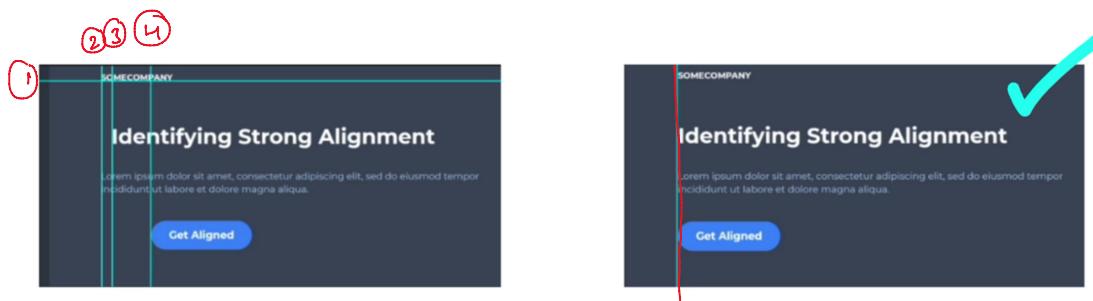
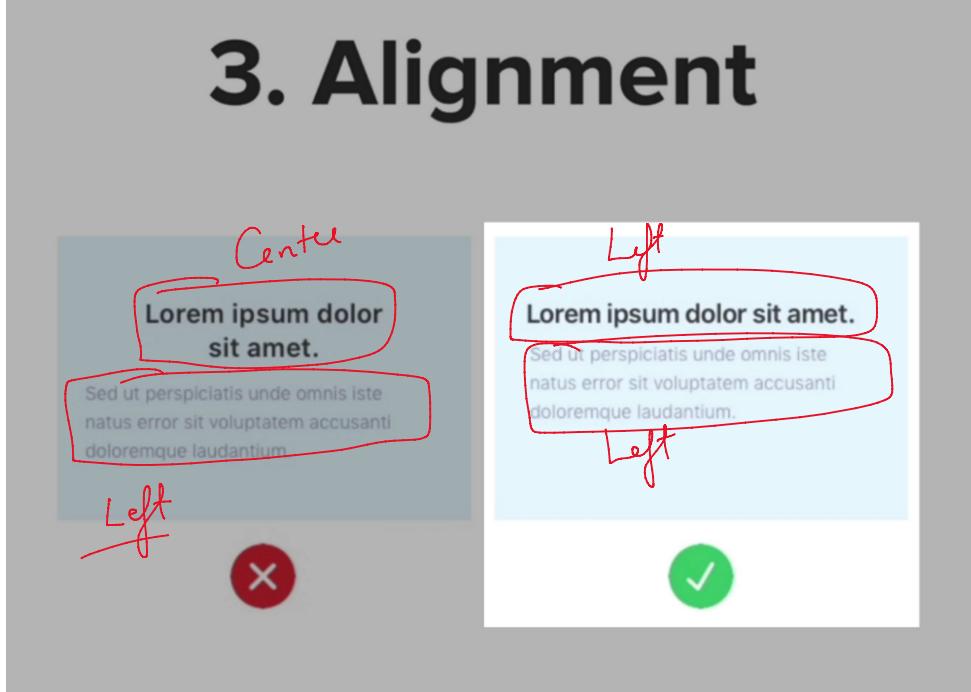
Mobile phone: 3-5

Tablet: 5-7

Laptop : 8-10



3. Alignment



The image compares two versions of a website for 'Design Solutions'.

Left Version:

- Navigation:** Shows a sidebar with navigation items like 'Navigation', 'Fusce porta suscipit eros', 'Duis vitae urna', 'Fusce non ligula', 'Donec hendrerit magna', 'Vivamus sodales', and 'egestas. Fusce in libero id laus auctor dignissim. Sed et arcu. Curabitur ac magna eu elit rhoncus ultrices. Proin quis metus nec mauris dictum convallis. In auctor mauris quis neque.'
- Content:** Includes sections for 'Fusce porta suscipit eros', 'Duis vitae urna', 'Fusce non ligula', 'Donec hendrerit magna', 'Vivamus sodales', and 'egestas. Fusce in libero id laus auctor dignissim. Sed et arcu. Curabitur ac magna eu elit rhoncus ultrices. Proin quis metus nec mauris dictum convallis. In auctor mauris quis neque.'
- Footer:** Shows '1234 Main St Springfield, IL 60528 123-456-7890' and a search bar.

Right Version:

- Navigation:** Shows a sidebar with navigation items like 'Navigation', 'Fusce porta suscipit eros', 'Duis vitae urna', 'Fusce non ligula', 'Donec hendrerit magna', 'Vivamus sodales', and 'egestas. Fusce in libero id laus auctor dignissim. Sed et arcu. Curabitur ac magna eu elit rhoncus ultrices. Proin quis metus nec mauris dictum convallis. In auctor mauris quis neque.'
- Content:** Includes sections for 'Fusce porta suscipit eros', 'Duis vitae urna', 'Fusce non ligula', 'Donec hendrerit magna', 'Vivamus sodales', and 'egestas. Fusce in libero id laus auctor dignissim. Sed et arcu. Curabitur ac magna eu elit rhoncus ultrices. Proin quis metus nec mauris dictum convallis. In auctor mauris quis neque.'
- Footer:** Shows '1234 Main St Springfield, IL 60528 123-456-7890' and a search bar.

Handwritten annotations 'Left' with a red circle is placed below the left version, and 'Right' with a green checkmark is placed below the right version.

4. White Space

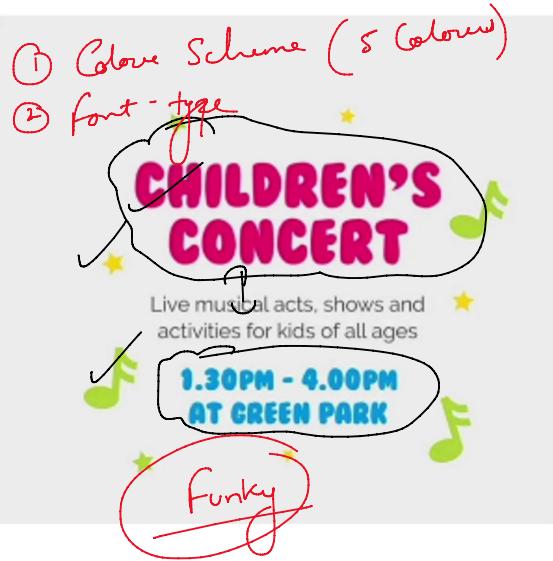


Premium

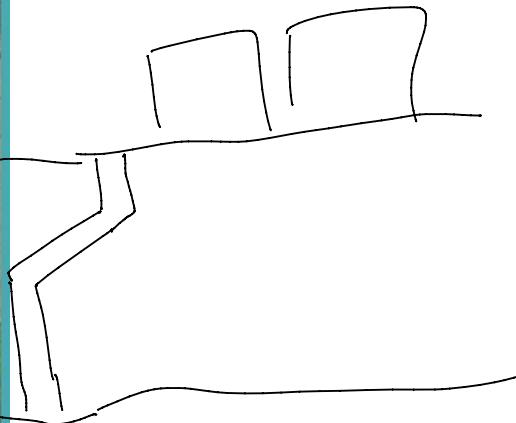
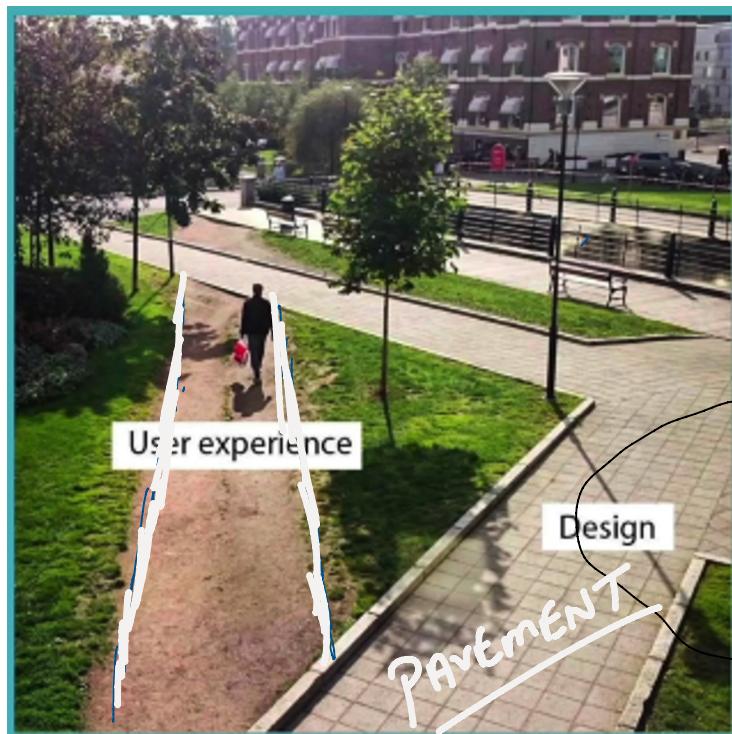


Discounts Shop (Sale)

5. Design According to your Audience



★ UX (User Experience)



Creating An Element in a way that user
Can easily figure out how to use that particular element

1. Simplicity

MONOCLE

ABOUT US LISTEN LIVE

Subscribe Shop M24 Radio Magazine Travel More

MONOCLE TODAY. Our top stories

In Italy they're investing in young craftspeople and tailors. But here in the US people don't think it's a good enough job.

Emily Adams Bode, Founder, Bode

The Monocle Minute

RECIPE EDITION, NICOLE PISANI

新闻 大家正在搜：宁津生院士逝世

北京 4°C

ABOUT US LISTEN LIVE

Subscribe Shop M24 Radio Magazine Travel More

MONOCLE

MONOCLE TODAY. Our top stories

In Italy they're investing in young craftspeople and tailors. But here in the US people don't think it's a good enough job.

Emily Adams Bode, Founder, Bode

The Monocle Minute

RECIPE EDITION, NICOLE PISANI





- 武汉部分医院恢复正常门诊 舟山音乐会无观众
- 直明餐饮堂食恢复 医护人员为新冠肺炎患者理发
- 墨大利新冠确诊患者近2.5万 征集全民抗疫
- 侃侃绿色毛衣复古摩登 王俊凯拉链西装个性十足

视频 纪录片 综艺



新浪产品



新浪众测



OPPO Watch智能手表



价格: ¥1999

- 入境航班三个风险等级 境外归来三种情形可居家观察
- 良好的公民素养和国民心态是“硬核”力量 逐渐复工
- 物价上涨楼市怎么走? 经济放缓疫情冲击物价会高?
- 送医心肺健康培训为何疫区新庄 为医护人员心理解压
- 专家谈中医药在疫症中的重要作用 出租车内如何防护?
- 澳北地区面对困难谁选择是班主“门苏队”可歌
- 多地生活秩序逐步恢复 是治愈困难卫生给役违法犯罪
- 该中国实情频频在外热火 疫政署将名化中国进抨击
- 大幅下放用地审批权降低提高审批质量 保障农民利益
- 精准救治 及时奔跑 同机魔较量 疫情新热点
- 关键的越要有的放矢 每位英雄中国人民都将被历史铭记
- 如何有效防范 会否二次暴发?一线专家解答 疫情地图
- 钟南山: 目前无证据说明新冠病毒会成季节性疾病

RECIPE EDITION, NICOLE PISANI

Find neighbourhoods on Monocle 24. Containing our series of recipes from some of the world's best chefs, we hear from Nicole Pisani, co-founder of the Chefs in Schools initiative.

Monocle 24 Radio
THE MENU - 2 min listen

More on food and drink:

MAGAZINE

MONOCLE 24

Lessons from leaders

A parcel of masks I sent from Hong Kong – the city is currently overflowing with personal hygiene products – arrived in the UK last week. My family is now all fit for the school term – so I’m forced to include instructions on how to wear them. Don’t be fooled: there’s etiquette and fashion to consider. And while I’m a late adopter myself, I have seen these...

2. Consistency

xfinity Shop My Account Support My XFINITY TV Email Sign In Comcast Business

Find the best deals in your area Enter address to view offers

Just Recorded Empire

XFINITY TV Go Watch Online Watch On TV Saved

Monday Night Shows See All TV

Sunday Night Shows See All TV

Top News Sports Finance TV Entertainment Life Games



Keep the repeating elements which are repeating on every

of similar design:

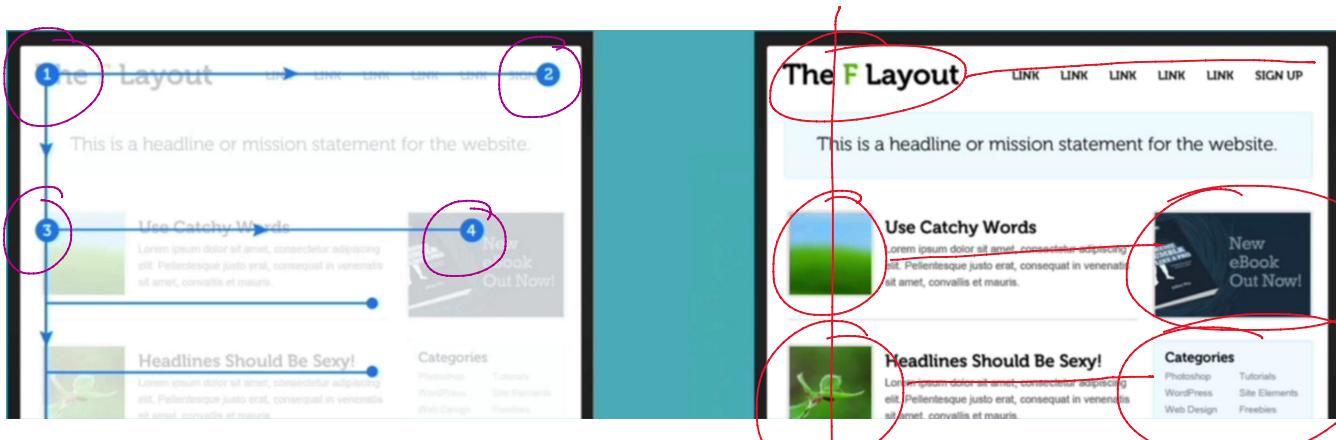
E.g. : ① Footer }
 ② Navbar }

3. Reading Pattern



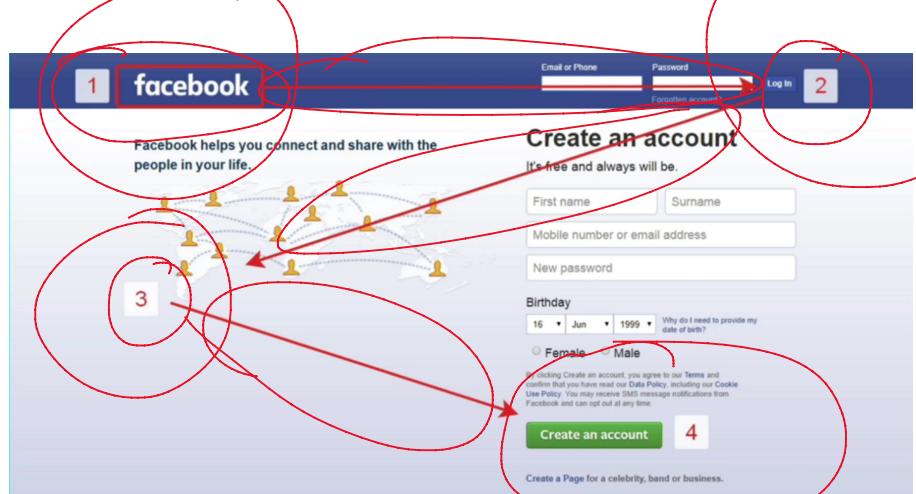
Poge

3. Reading Pattern



Use this for Sparse amount of Content.

Or else Z or Zig-Zag Pattern if you more Content.



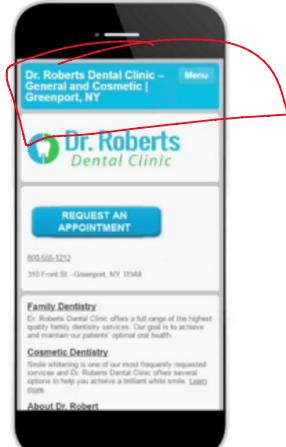
5. All responsive Design

B

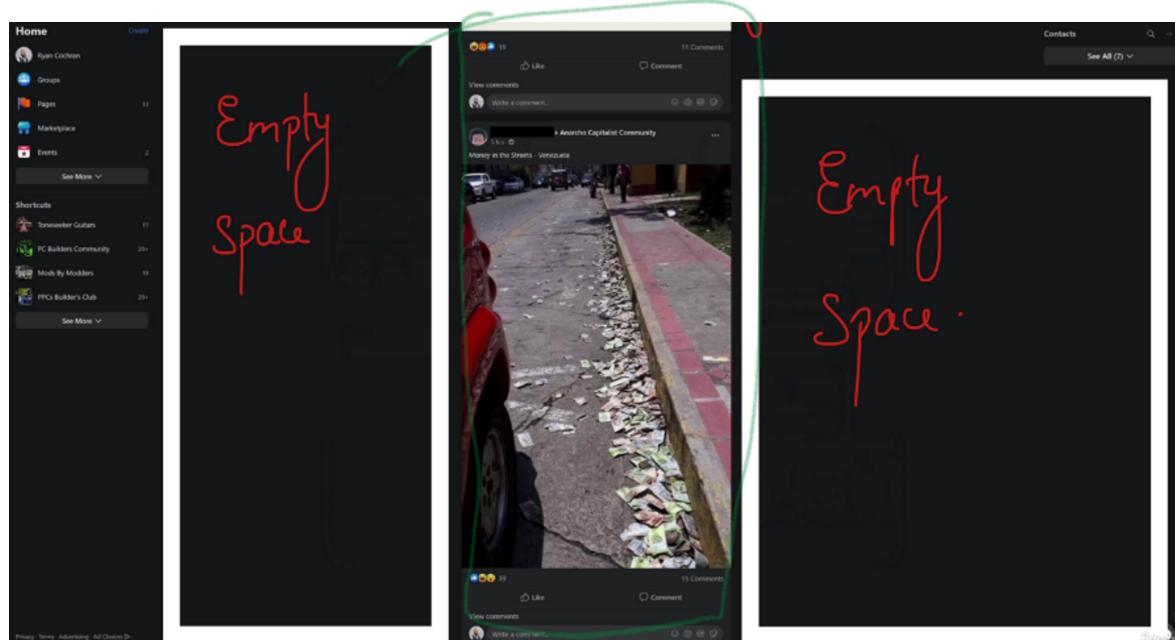
have

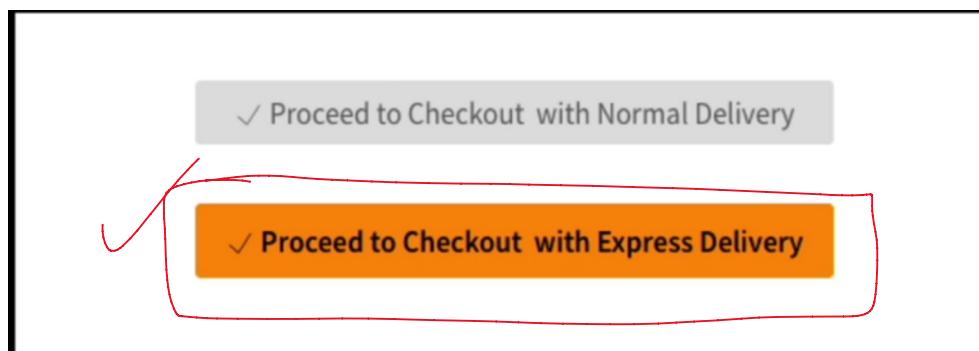
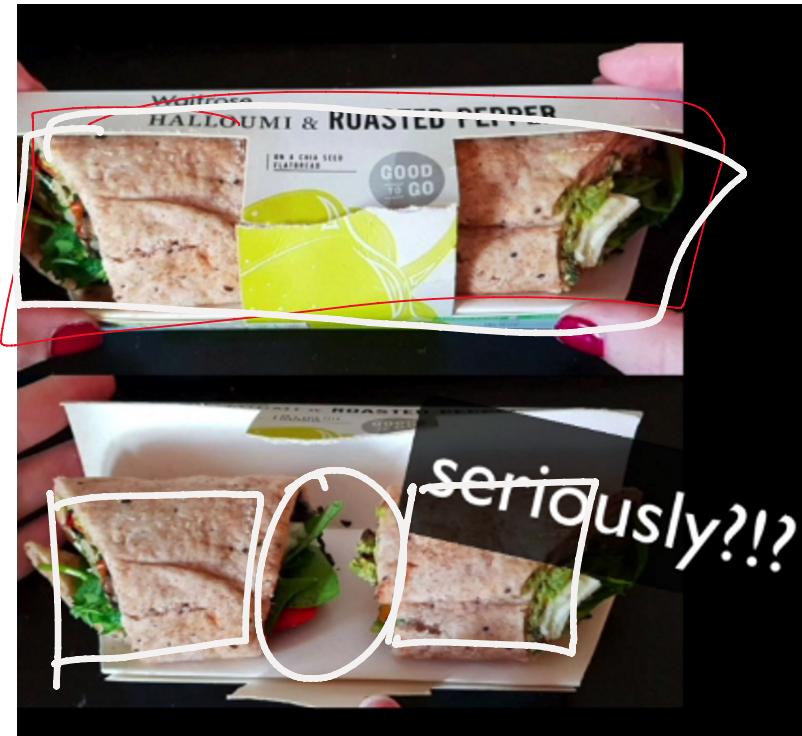
5. All Platform Design

✓ Good



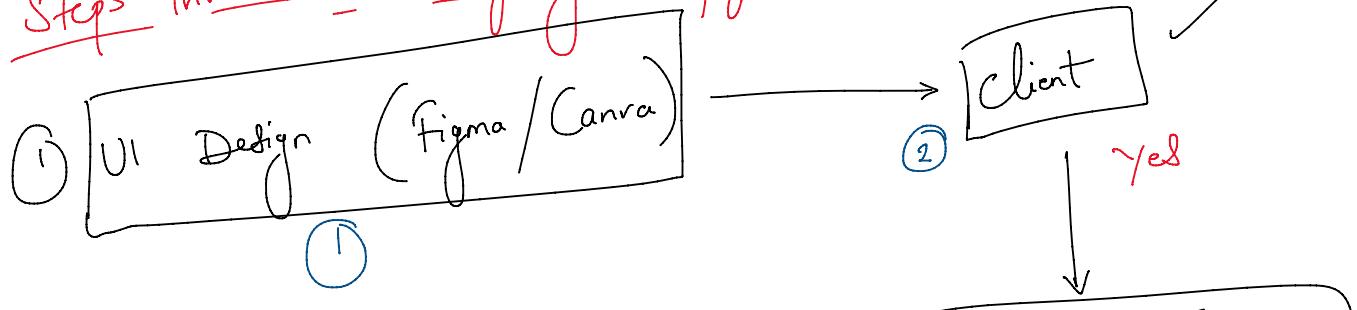
✗ Bad



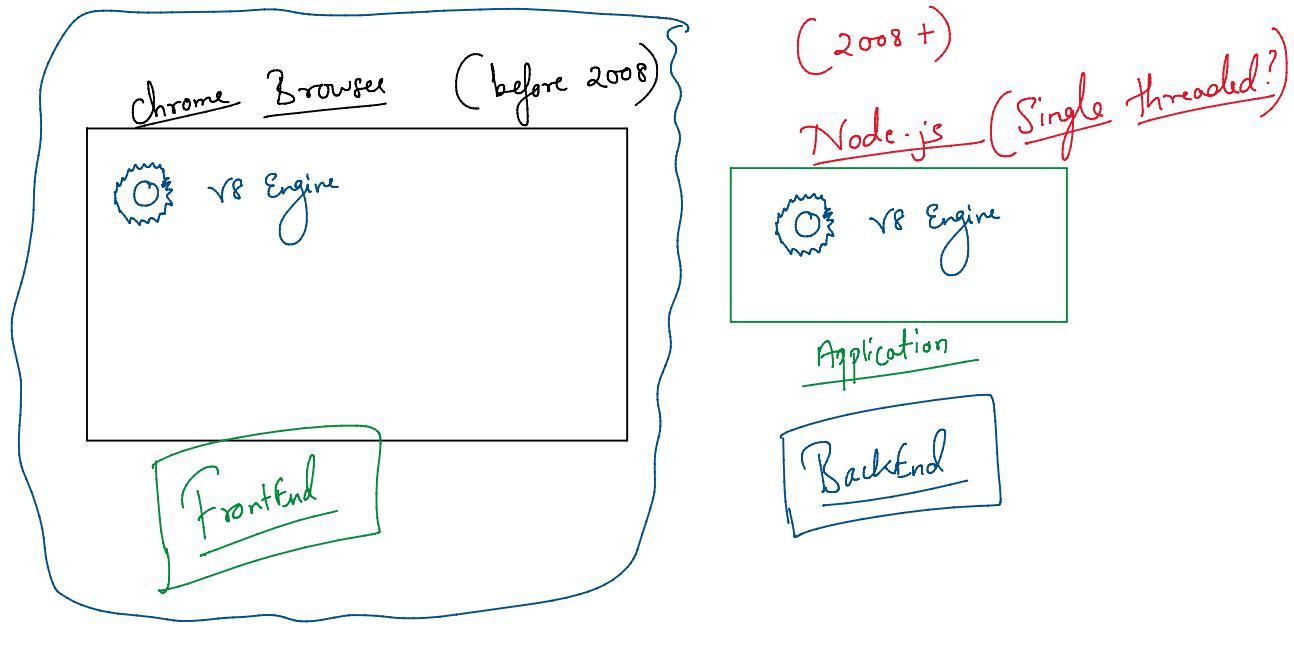
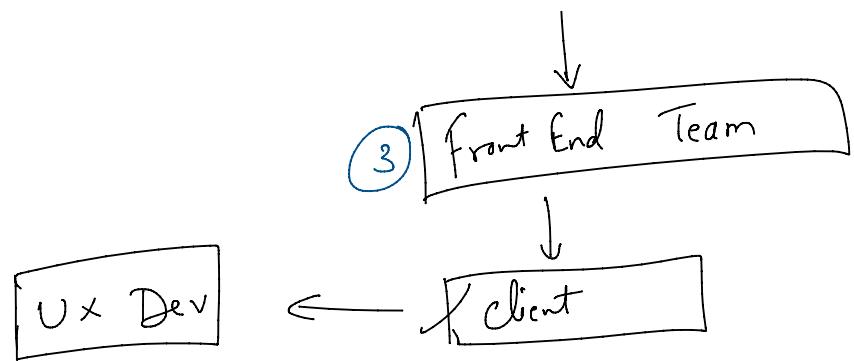


If you would like us to no longer continue to stop not sending you special deals and offers every week, please indicate you are inclined to yes by not checking the box.

Steps involved in Designing Webpage :



①



You name

"Rahul"

name

* Data types in JS *

[You can put any type of data in any variable
No need to specify the type of data that goes into a variable]

① Number : int, float -----

② String : Group of characters

Single : Number
Symbol
Letter
Space

[Dynamically typed language]
[Python, JS]

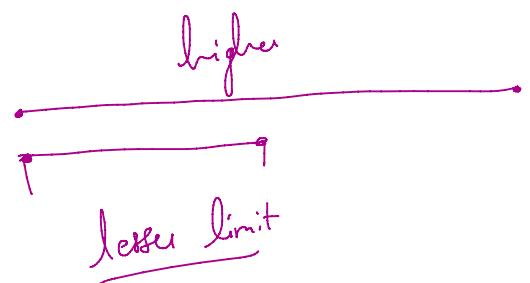
③ Boolean : true/false

able
of
J

- 0

③ Boolean : true/false.

④ BigInt : BigInt(" — ");



⑤ Symbol : Symbol(" — ");

⑥ null : let a = null;

⑦ undefined : let a;
[undefined]

PRIMITIVE

Non-Primitive [Complex/
Reference]

① Object

② Array

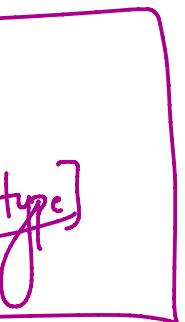
③ Function

④ Map

⑤ Class

num 7
clg(num%2);

3
2 7
-



$$\begin{aligned}
 & \text{clg}(\text{num}(\underline{\text{1}}/\underline{\text{2}})) ; \\
 & \text{clg}(\text{num}(\underline{\text{1}})) ; \\
 & 7/2 = \underline{3.5}
 \end{aligned}$$

2
 +
 - 6
 1

nestingOfLoops.js

```

1  for(let a = 1 ; a <= 5 ; a++)
2  {
3    for(let b = 1; b <= 3 ; b++)
4    {
5      console.log("a : " + a + " b : " + b);
6    }
7  }
  
```

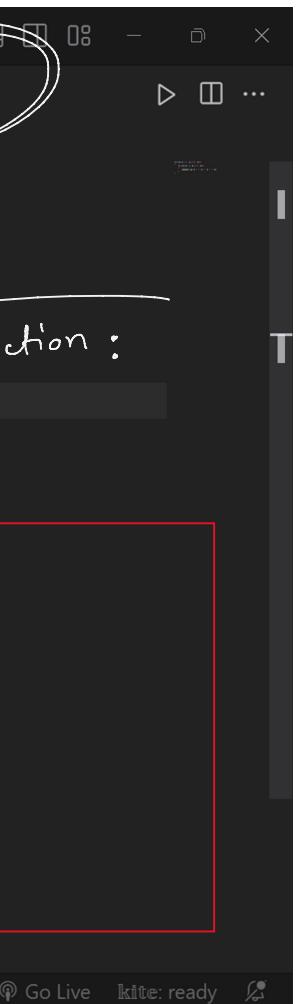
Output of clg fun

Output Screen :

```

a:1, b:1
a:1, b:2
a:1, b:3
a:2, b:1
a:2, b:2
a:2, b:3
a:3, b:1
  
```

a:3, b:3



The diagram illustrates the generation of a right-angled triangle pattern. On the left, a series of horizontal lines of increasing length are shown, labeled 1 through 5. Each line is composed of asterisks (*). To the right, these lines are mapped to a sequence of numbers and parentheses, labeled 'stars'. The mapping is as follows:

- Line 1 maps to stars (1)
- Line 2 maps to stars (1, 2)
- Line 3 maps to stars (1, 2, 3)
- Line 4 maps to stars (1, 2, 3, 4)
- Line 5 maps to stars (1, 2, 3, 4, 5)

Below the diagram is a screenshot of a code editor showing the following JavaScript code:

```
/*  
TARGET :  
*  
* *  
* * *  
* * * *  
*/  
  
Let horizontalPattern = "";  
  
for(let i = 1; i < 6 ; i++)  
{  
    horizontalPattern = horizontalPattern + "* "  
}  
  
console.log(horizontalPattern);
```

The screenshot shows a code editor with a file named `RightAngledTrianglePattern.js`. The code prints two patterns: a horizontal pattern of asterisks and a star pattern of asterisks arranged in a triangle.

```
// File: RightAngledTrianglePattern.js
// Date: 8.12.2023

12 // Let horizontalPattern = "";
13
14 // for(let i = 1; i < 6 ; i++)
15 //{
16 //     horizontalPattern = horizontalPattern + "* ";
17 //}
18
19 // console.log(horizontalPattern);
20
21 let starPattern = "";
22
23 for (let line = 1; line <= 5; line++)
24 {
25     for (let stars = 1; stars <= line; stars++)
26     {
27         starPattern = starPattern + "* ";
28     }
29 }
30
31
32 console.log(starPattern);
```

Handwritten annotations explain the logic:

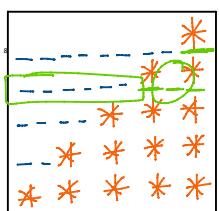
- A box labeled "line" contains the value "3".
- A box labeled "Stars" contains the value "4".
- An arrow points from the variable `starPattern` to a diagram of a right-angled triangle made of asterisks. The triangle has 5 rows. The first row has 1 asterisk, the second has 2, the third has 3, the fourth has 4, and the fifth has 5. A bracket under the triangle indicates the range of the loop, from 1 to 5.
- Below the triangle, the value "4" is written next to a row of four asterisks.
- At the bottom, the code `console.log(starPattern);` is annotated with a large bracket underneath it, indicating the output of the program.

Hw :

①	5	4	3	2
	5	4	3	2
	5	4	3	
	5	4		
	5			

	1	2	3	4	5
	1	2	3	4	
	1	2	3		
	1	2			
	1				

Step 1: Complete the Square

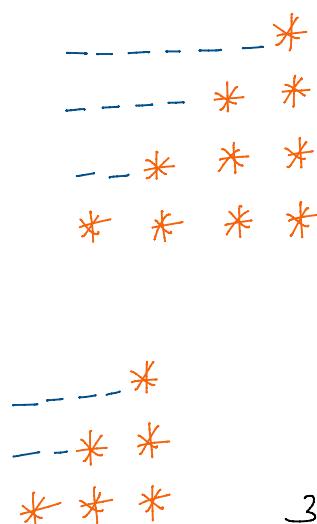


Blank Spaces

8
6
4
2
0

Stars Line No

1	→ 1
2	→ 2
3	→ 3
4	→ 4
5	→ 5



4 lines
6 initial blank spaces

Lines: 5
initial blank spaces: 8

$$[(\text{TotalLines} * 2) - 2]$$
initial bspaces

$3 \rightarrow 4$

```

RightAngledTrianglePattern.js
InvertedRightAngledTriangle.js
8.12.2023 > InvertedRightAngledTriangle.js ...
5 // console.log(initialBlankSpaces);
6
7 let starPattern = "";
8
9 for (let line = 1; line <= totalLines; line++) {
10   for (let spaces = initialBlankSpaces; spaces > 0; spaces--) {
11     starPattern += " ";
12   }
13   for (let stars = 1; stars <= line; stars++) {
14     starPattern += "* ";
15   }
16   starPattern += "\n";
17   initialBlankSpaces -= 2;
18 }
19 console.log(starPattern);
20
21
22
23
24
25
  
```

Annotations:

- Line 10: $\text{initialBlankSpaces}$ is crossed out.
- Line 11: $\text{spaces} > 0$ is crossed out.
- Line 12: $\text{spaces}--$ is crossed out.
- Line 13: $\text{stars} <= line$ is crossed out.
- Line 14: $\text{stars}++$ is crossed out.
- Line 16: $\text{starPattern} += "\n"$ is crossed out.
- Line 17: $\text{initialBlankSpaces} = \text{initialBlankSpaces} - 2;$ is crossed out.
- Handwritten labels:
 - 'line' is labeled above line 10 with a box around 2.
 - 'Spaces' is labeled above line 11 with a box around 0.
 - 'Stars' is labeled above line 13 with a box around 3.
 - 'initial Blank Spaces' is labeled above line 17 with a box around 4.
 - 'StarPattern' is labeled above line 19 with a box around the resulting string of dashes and stars.
- A red box highlights the calculation $(\text{TotalLines} * 2) - 2$ and the value 4 (initial blank spaces).
- The output window shows the resulting inverted right-angled triangle pattern.

★ Increment & Decrement Operators Imp ($++, --$)

$D \sim -T$

param

{

Post - I

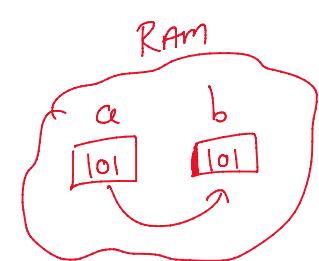
* Increment \rightarrow ----- + - .

Pre - I

let a;
let b;

$a = 100$

$dg(a)$
 $dg(b)$



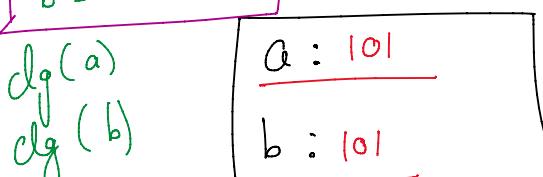
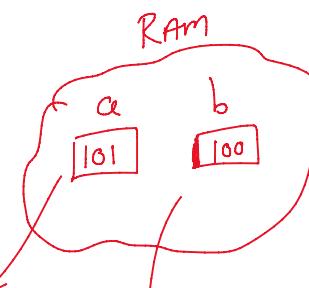
Post - I

let a;

let b;

$a = 100$;

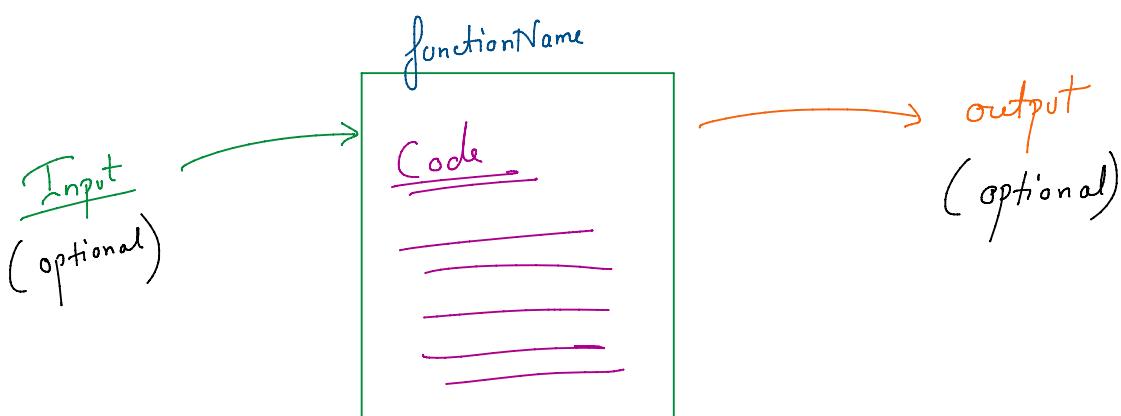
$a++$;



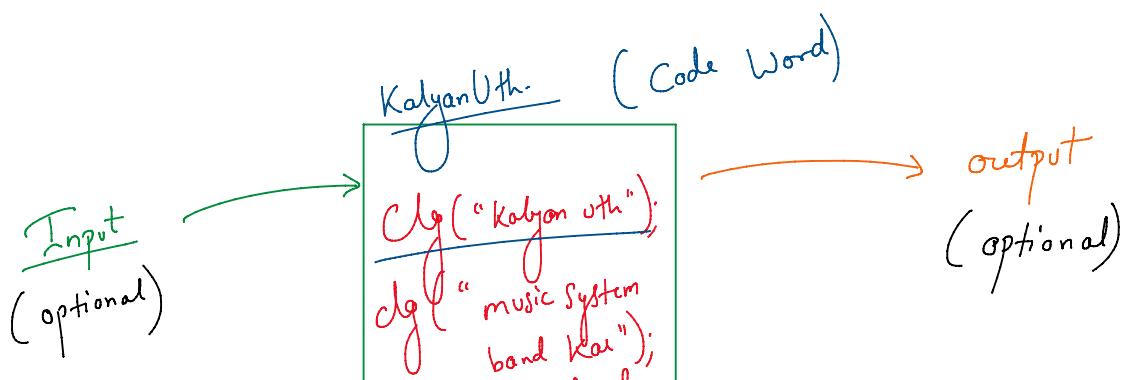
① Look at the immediate left side woala operator of a variable.

Bcoz that is the operator which gets executed first.

* Functions (Just like a Machine)

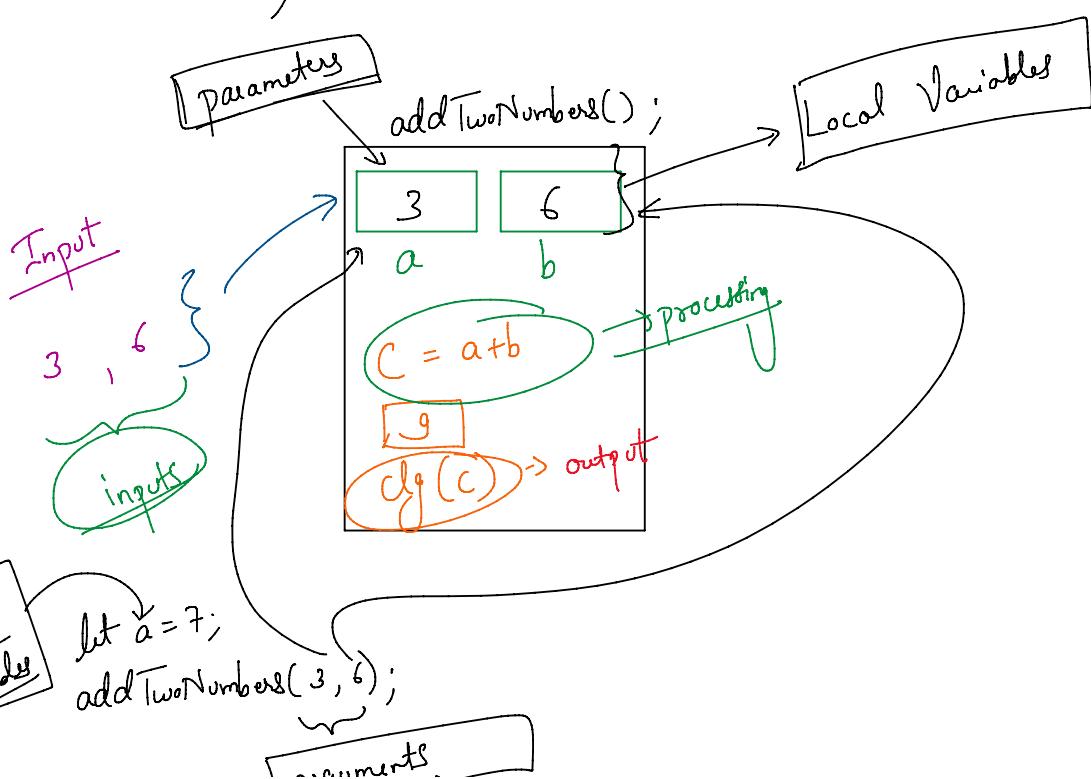
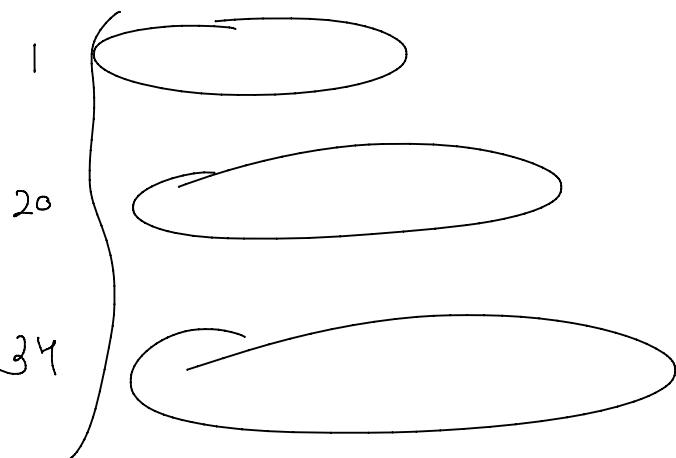
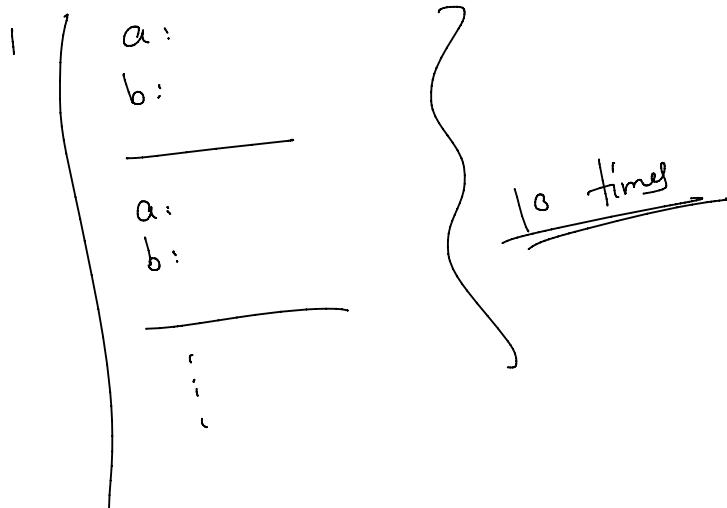


Turn the machine on by taking its name (Calling a function)



(optional)

`dg("music system"
 "band Kai");`
`dg("open closed
 doors & windows");`



Function addTwoNumbers(a, b),
~~~~~  
arguments

## Creating A Function

function functionName (Parameters)  
{     // Code  
}

}

function addTwoNum(a, b)  
{     let c = a + b;  
    clg(c);  
}

The diagram illustrates the scope of variables in a JavaScript function. It shows a code editor window with a file named 'Functions.js' containing the following code:

```
function addTwoNumbers(a, b){  
    let c = a + b;  
    console.log("The addition is : " + c);  
}  
  
let a = 100;  
let a = 103;  
  
addTwoNumbers(3, 6);
```

Annotations explain the scope of variable 'a':

- A red circle highlights 'a' in the parameter list, with an arrow pointing to a callout bubble that says "not able to see".
- A red circle highlights 'a' in the assignment 'let a = 103;', with an arrow pointing to a callout bubble that says "main scope".
- A red circle highlights 'a' in the function call 'addTwoNumbers(3, 6)', with an arrow pointing to a callout bubble that says "fn scope".



A screenshot of a code editor window titled "Functions.js". The code defines a function `addTwoNumbers` that adds two numbers and returns their sum. Below it, a variable `returnedVal` is assigned the value returned by the function. Handwritten annotations include a red box around the function definition, a red arrow pointing from the function body to the assignment line, and a red circle highlighting the value `9` in the assignment line.

```
function addTwoNumbers(a, b)
{
    let c = a + b;
    return c;
}

let returnedVal;
returnedVal = addTwoNumbers(3, 6);
// returnedVal = 9;
```

## \* Prime Nos Program \*

No divisible by 1 & itself

1 2, 3, 4, 5/6, 7, 8, 9      10  
  2                          5

No decimal pt.  
•, 0 remainder  
1 — NOT PRIME  
nor COMPOSITE

1 X 11 ✓

13 (n)  
1                          13  
2 — 12



$$2 - (n^{-1})$$

A screenshot of a code editor showing a JavaScript file named `primeNumbers.js`. The code is a prime number checker. Handwritten annotations explain the state of variables during execution:

- `num` is labeled with the value `7`.
- `checker` is labeled with the value `6`.
- `primeFlag` is labeled with the value `T`.

The code in `primeNumbers.js` is as follows:

```
11.12.2023 > primeNumbers.js > ...
9 |     console.log(num + " is a PRIME NUMBER");
10 |
11 | }
12 | else if (num > 2)
13 | {
14 |     for (let checker = 2; checker < num; checker++)
15 |     {
16 |         if(num % checker == 0)
17 |         {
18 |             primeFlag = false
19 |             break;
20 |         }
21 |         else
22 |         {
23 |             primeFlag = true;
24 |         }
25 |     }
26 |
27 |     if (primeFlag == true)
28 |     {
29 |         console.log(num + " is a PRIME NUMBER");
30 |     }
31 |     else
32 |     {
33 |         console.log(num + " is NOT a PRIME NUMBER");
34 |     }
35 | }
```

Annotations highlight the `checker` loop, the `primeFlag` assignment, and the final `if` statement that logs the result.



A screenshot of a code editor with handwritten annotations explaining type conversion in JavaScript.

The code in `TypeConversion.js` is as follows:

```
1 let num = "123";
2 console.log(typeof(num));
3
4 Let numberInINT = Number(num);
5 console.log(typeof(numberInINT) + " : " + numberInINT);
6
7 // Let num = 123;
8 // Let stringedNum = JSON.stringify(num);
9 // console.log(typeof(stringedNum) + " : " + stringedNum);
```

Annotations:

- An arrow points from the string "String" to the output box containing "123".
- A bracket underlines the line `console.log(typeof(numberInINT) + " : " + numberInINT);` with the handwritten note "Shows the type of variable".
- A blue box labeled "Number" surrounds the variable `numberInINT`.
- Two boxes labeled "num" contain the value "123". One is highlighted with a red border.
- A green box contains the output "String String : 123".



A screenshot of the Visual Studio Code interface. The top bar shows the file path: 12.12.2023 > FunctionExpression.js > ... . The code editor displays the following JavaScript code:

```
1  let addFunction = function(num1, num2)
2  {
3      return (num1 + num2);
4  }
5
6 // Let result = addFunction(12,2);
7
8 let func = addFunction;
9
10 let result = func(12,2); 15
11
12 console.log(result);
```

The code is annotated with several hand-drawn elements:

- A red box highlights the line `let func = addFunction;`.
- A red box highlights the line `let result = func(12,2);`, with the number "15" written next to it.
- A red box highlights the line `console.log(result);`.
- A large curly brace on the right side groups the `addFunction` declaration, the assignment to `func`, and the assignment to `result`.
- A small red circle highlights the number "15" at the end of the `result` assignment line.

The terminal tab is active, showing the command `$ node FunctionExpression.js` and its output "15". The status bar at the bottom indicates the file is saved and shows the current line and column: Ln 12, Col 21.

A screenshot of a code editor window showing a closure diagram. The code in the editor is:

```
function add(a) {
    return function(b) {
        return a + b;
    };
}
```

The variable `a` is highlighted with a green oval and labeled `num1`. The variable `b` is highlighted with a green rectangle and labeled `num2`. The returned function is highlighted with a red box and labeled `(num1 + num2);`. Handwritten annotations include a red box around `num1` with the text `lol`, a red box around the returned function with the text `lol`, and a red box around the parameter `b` with the text `func`. A handwritten note `(Sakshi)` is written next to the `func` annotation. The status bar at the bottom shows: RLF, { JavaScript, Go Live, kite: ready,

/JavaScript ES-6/CodeFiles/12

RLF { JavaScript Go Live kite: ready



File Edit Selection View Go ...

← →

CodeFiles



44



✖ ⊗ 0 ⌛ 0 { } : 0 ✓ javascript | ✓ Strings.js ⚡

Ln 1, Col 19 Spaces: 4 UT

JS Strings.js X

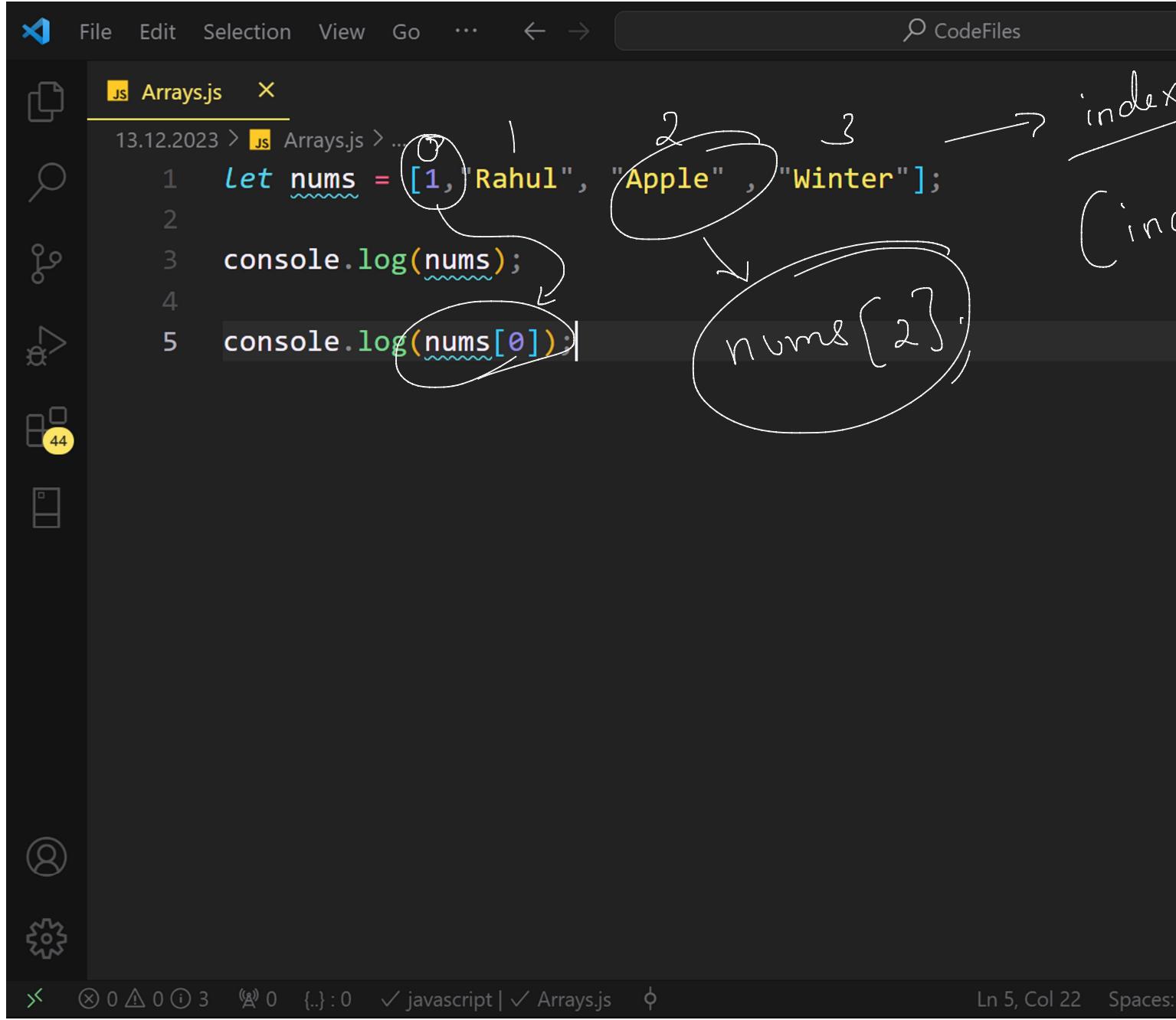
12.12.2023 > JS Strings.js > ...

1 Let str = "RAHUL";

0 1 2 3 4  
| | | |  
-5 -4 -3 -2 -1  
↙

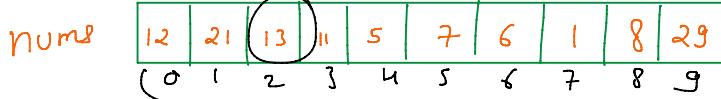


F-8 CRLF { JavaScript Go Live kite: ready



## Steps to Solve this problem

- ① Create an array of 15 elements :



- ② Create an empty array

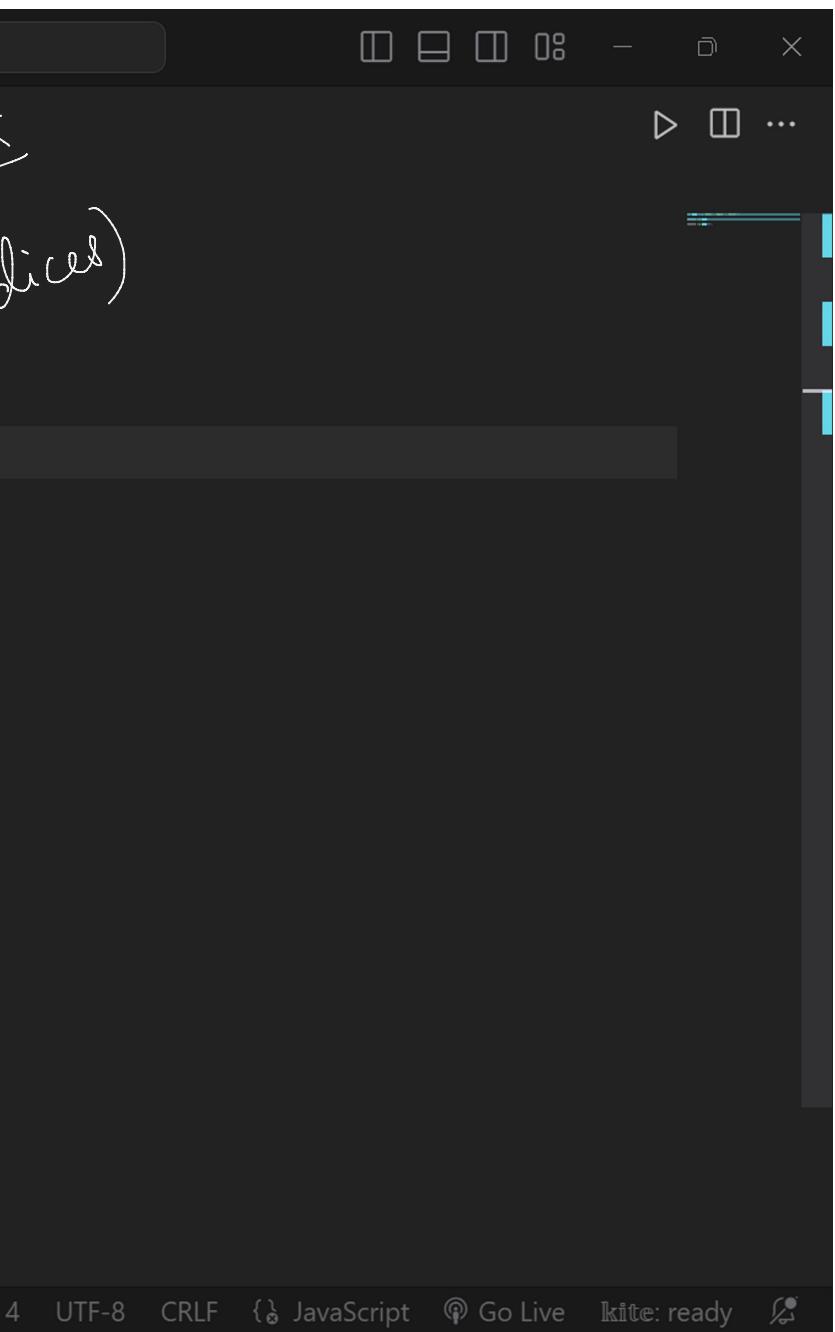
primos.push(num);

- ③ pick each element from nums

The diagram illustrates the execution flow of the following JavaScript code:

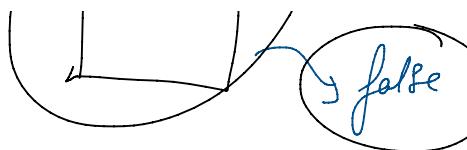
```
let number;  
for(let index=0; i < nums.length; index++) {  
    number = nums[index];  
}
```

The code starts with a declaration of the variable `number`. The loop begins with `index` set to 0. Inside the loop, the value at index 0 of the array `nums` is assigned to `number`. A decision point follows, checking if `index` is less than the length of `nums`. If true, the loop continues with `index++` and the next iteration. If false, the loop exits.



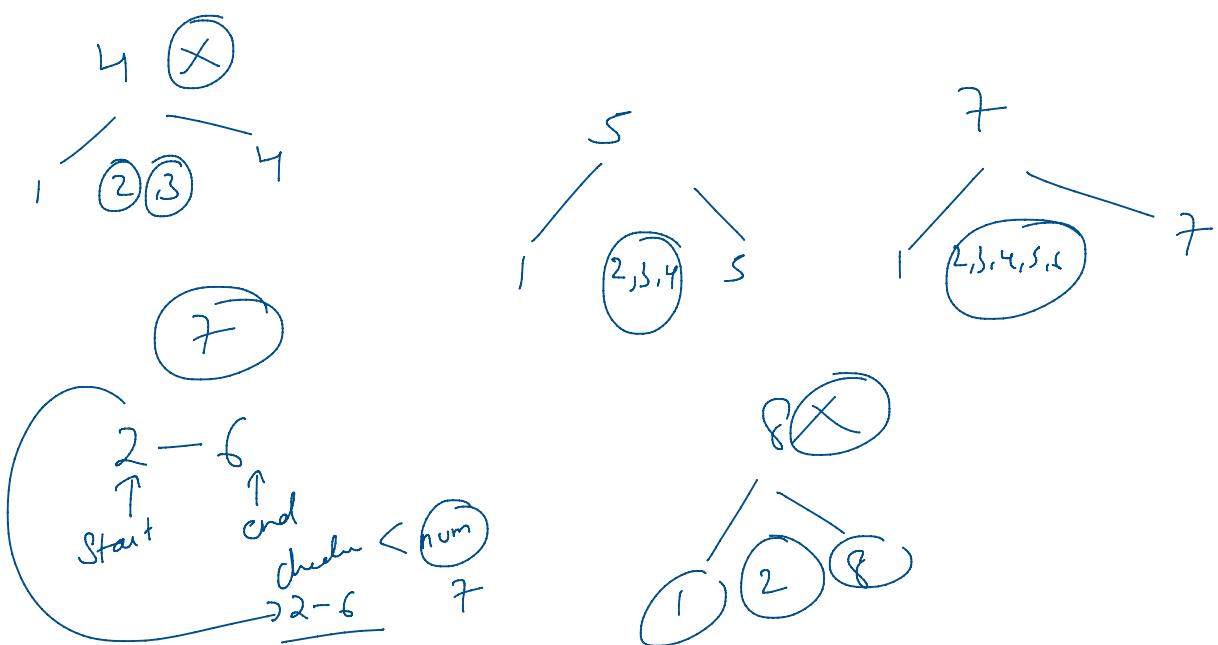
③ Pick each element from nums

number



④ Check if that picked Element (number) is Prime

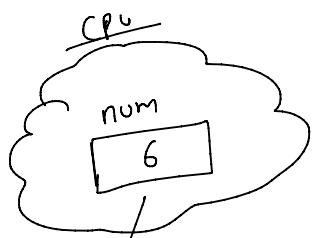
⑤ If it is prime, then push it into the empty Array.



for Loop:

Code:

```
1 for (let num=1; num < 6; num++)  
2 {  
3   console.log(num);  
4 }  
5 in/decr.
```



Output target

per phase  
incr /  
decr

1  
2  
3  
4  
5

Output:

1  
2  
3  
4  
5

Prime Nos

Array:

Prime Nos Using Arrays



A screenshot of the Visual Studio Code interface. The top bar shows tabs for 'ComplexObjects.js', 'ForInLoop.js', and 'ForInWithArrays.js' (which is the active tab). A handwritten note is overlaid on the code editor area, reading: 'In arrays, 'index number' is the property.' The code editor displays the following JavaScript code:

```
1  let numbers = [1,2,3,4,5];
2      0 1 2 3 4
3  for (num in numbers)
4  {
5      console.log(num);
6  }
```

The bottom panel shows a terminal window with the following output:

```
rahu@Romeo-Alpha09 MINGW64 /r/APTECH_Batches/3. MERN Stack Pillai's Batch/JavaScript ES-6/CodeFiles/15.12.2023
$ node ForInWithArrays.js
0
1
2
3
4
```



Every element  $\rightarrow$  page

```

File Edit Selection View Go Run ...
1 Let nums = [233, 56, 23, 56, 88, 3, 4, 788, 12, 11];
2 // arrName.length -----> gives the total number of elements in the array
3
4 Let evenNos = [];
5
6 Let number;
7
8
  
```

(Any number by 2 is even)  $\rightarrow$  num[0]

number = num[0];

number = 233;

number = 233 / 2 = 116.5

remainder = number % 2;

remainder = 1

gives us the remainder of the division.

Quotient = 116

Dividend = 233

Divisor = 2

Name of 1st Element

nums[indexNo]

indexNo

0 1 2 3 4 5 6 7 8 9

```

3 if (remainder == 0)
{
    console.log(number + " is an Even Number");
}
else
    console.log(number + " is Not an Even Number");
  
```

## ★ for Loops - Revisit ★

```

1 for( let number = 1; number < 11; number++)
2 {
3     console.log(number);
4 }
5
  
```

number जोपहल 11  
Peksha दोता असेल  
number++

number

11

RAM

Value print  
outpt

Output 21 आत हैचीत

1 - 1 7 1 10 21 21 आत हैचीत



5 out



Loop  $\Sigma T$   $\Sigma T$  time  
Content  $T$   
There many times execute.  
So slow cache

clg ("Hello");

nums

|   |   |   |   |   |    |
|---|---|---|---|---|----|
| 0 | 7 | 2 | 6 | 3 | 11 |
| 0 | 1 | 2 | 3 | 4 | 5  |

elements : 6  
last index : 5  
length : 6

index numbers

Name of the 1st element.

nums[0];

clg (nums[0]);

clg (nums[1]);

clg (nums[2]);

clg (nums[3]);

clg (nums[4]);

clg (nums[5]);

for (let count = 0; count <= 5;)

{  
clg (nums[count]);  
clg (nums[2]);

0  
7  
2

6  
3  
1

array

6  
count ++)

) ;

# ★ Billing System Project ★

① Prompt → How many items do you want in your bill?

↓  
7 (for loop)

Menu Mapper

1. "IDLI"

2. "Medu Vada"

3 \_\_\_\_\_

4 \_\_\_\_\_

5 \_\_\_\_\_

6 \_\_\_\_\_

7 \_\_\_\_\_

8 \_\_\_\_\_

9 \_\_\_\_\_

a \_\_\_\_\_

b \_\_\_\_\_

c \_\_\_\_\_

d \_\_\_\_\_

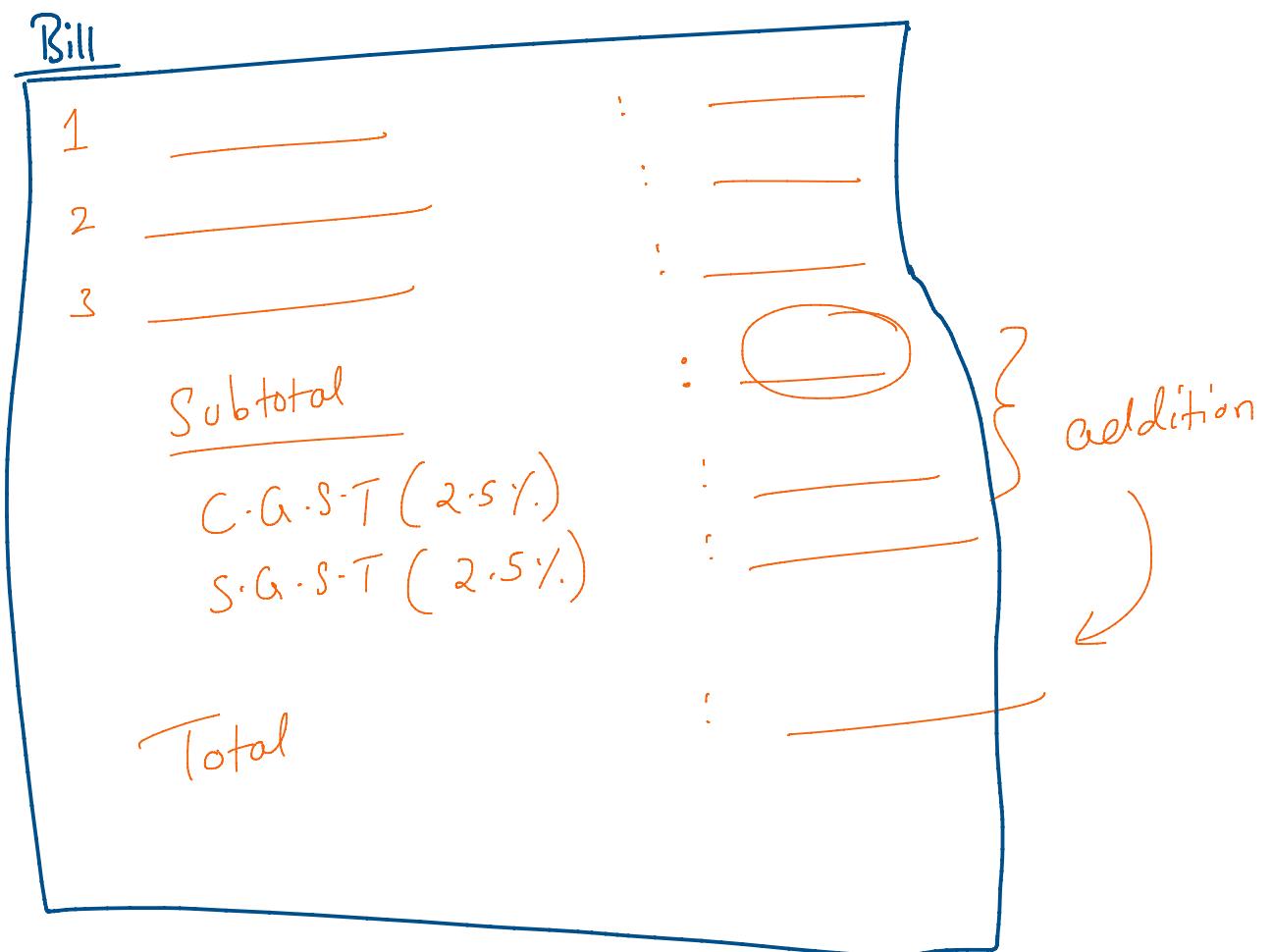
e \_\_\_\_\_

f \_\_\_\_\_

Successfully added Medu Vada, enter next dish:

Menu Mapper

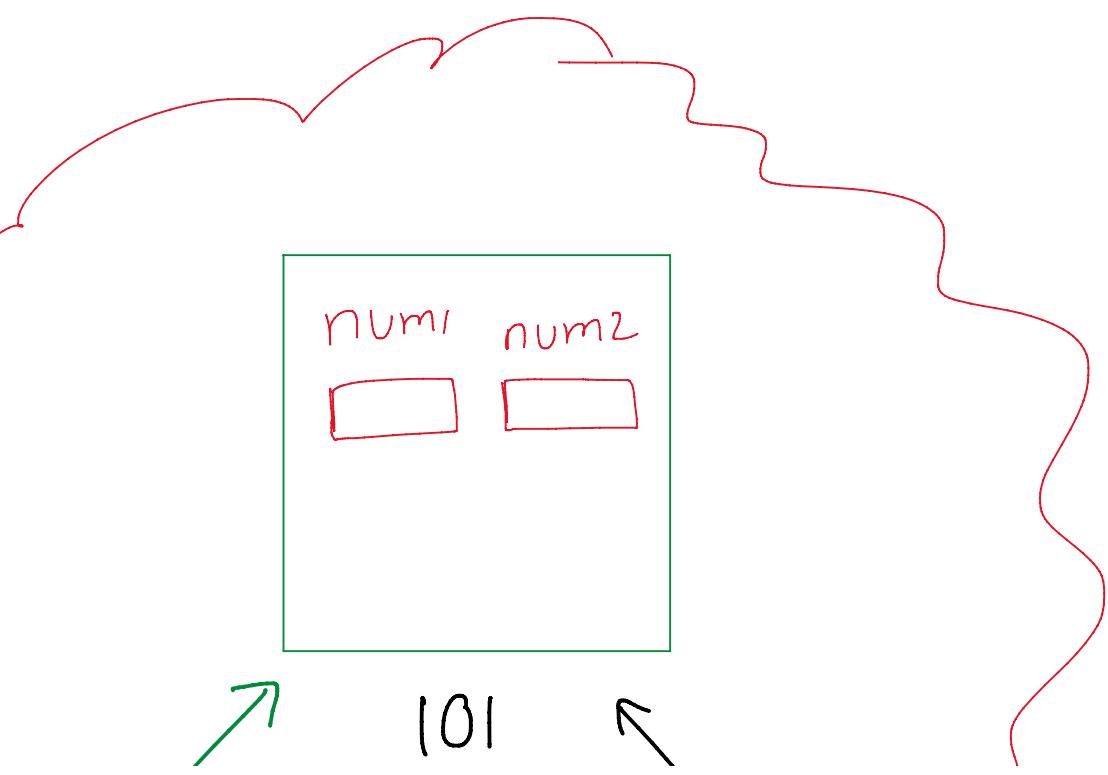




```

File Edit Selection View Go ...
JS MapMethod.js JS AnonymousFunctions.js X
20.12.2023 > JS AnonymousFunctions.js > add
1 function add(num1, num2)
2 {
3     console.log(num1 + num2);
4 }
5

```





File Edit Selection View Go ... ← →

JS MapMethod.js JS AnonymousFunctions.js X

20.12.2023 > JS AnonymousFunctions.js > ...

```
1 let add = function (num1, num2)
2 {
3     console.log(num1 + num2);
4 }
5
6 add(2,43);
7
8
```

Handwritten notes:

- A red oval highlights the line `let add = function (num1, num2)`.
- A callout bubble points to it with the text `let func = add;` and `(address)`.
- A box labeled `add` contains the value `101`.

nums [ 7, 5, 12, 22, 24 ]  
newArr[]  
for (  
{  
if (arr[index] % 2 == 0)  
{ newArr.push (arr[index]);  
}  
}

File Edit Selection View Go ... ← →

CodeFiles

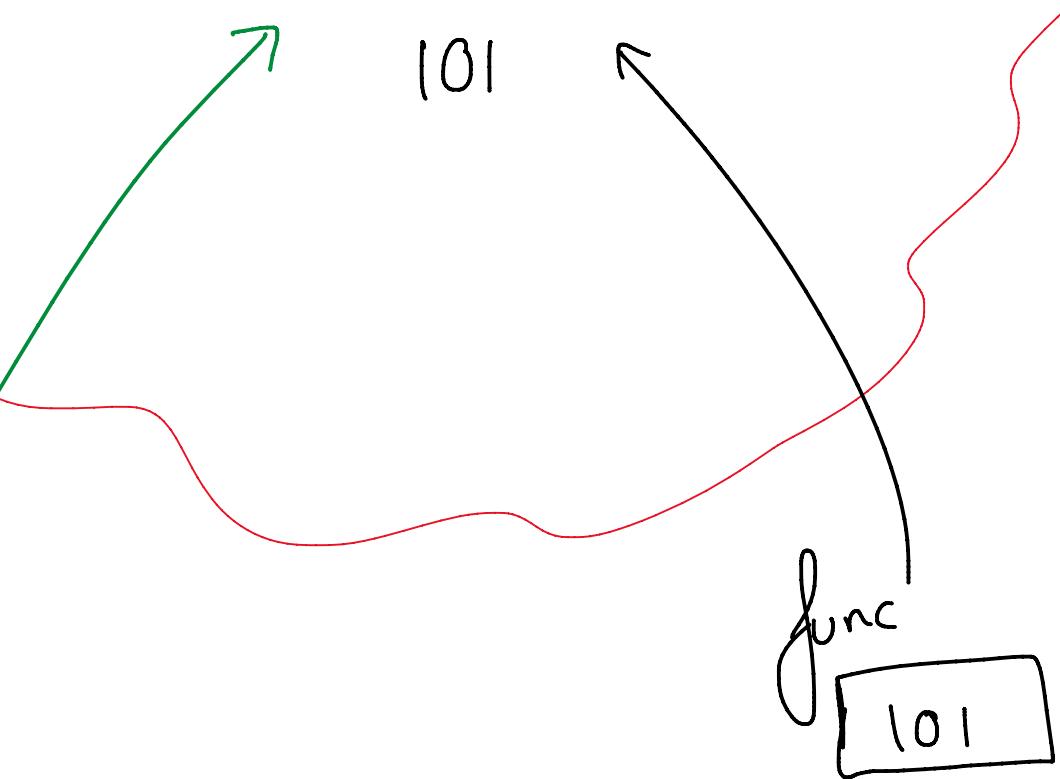
Jan '24 > 8.1.24 > filterMethod.js •

```
1 let nums = [4, 12, 35, 67, 22, 8];
2
3 // Let filteredNums = [];
4
5 function filterFunc(number) {
6     return number % 2 === 0;
7 }
8
9 let filteredArray = nums.filter(filterFunc);
10
11 console.log(filteredArray);
12
13
14 is number giving 0 as
15 the remainder upon
16 getting divided by
17 2?
18
19
20
21
22
```

Handwritten notes:

- A green bracket highlights the array `[4, 12, 35, 67, 22, 8]`.
- A green box labeled `number` contains the value `22`.
- A green cloud contains the numbers `4, 12, 22, 8`.
- A green box labeled `filter chi pishvi` contains the array `[4, 12, 22, 8]`.
- A note says `Callback functions` and `return`.
- A note says `converts into array` and `f returns the array`.

[ 4, 12, 22, 8 ]



```

File Edit Selection View Go ... ↵
CodeFiles
ArrayFiltering.js filterMethod.js
Jan 24 > 8.1.24 > filterMethod.js ...
3 // Let filteredNums = [];
4
5 let filteredNums = nums.filter((number) => number % 2 === 0);
6
7 console.log(filteredNums);
8
9 // Let number;
10
11 // for (index = 0; index < nums.length; index++) {
12 //   number = nums[index];
13 //   if (number % 2 === 0) {
14 //     filteredNums.push(number);
15 //   }
16 // }
17

```

```

File Edit Selection View Go ... ↵
CodeFiles
ArrayFiltering.js filterMethod.js
Jan 24 > 8.1.24 > filterMethod.js ...
3 // let filteredNums = [];
4
5 let filteredNums = nums.filter((number) => number % 2 === 0);
6
7 console.log('filtered Num : ' + filteredNums);
8
9 let mapArray = filteredNums.map((number) => number * 2);
10
11 console.log('After Running Map method : ' + mapArray);
12
13 // Let number;
14
15 // for (index = 0; index < nums.length; index++) {
16 //   number = nums[index];
17 //   if (number % 2 === 0) {
18 //     filteredNums.push(number);
19 //   }
20 // }
21

```

```

File Edit Selection View Go ... ↵
Jan '24
CodeFiles
Z_ReduceMethod.js
10:1.24 > Z_ReduceMethod.js > (o) number > @ nums.reduce() callback
1 let nums = [5, 6, 7, 8, 34, 23];
2
3 let number = nums.reduce((prev, next) => [
4   console.log(`prev : ${prev}, next : ${next}`);
5   return prev + next;
6 ]);
7

```

OUTPUT

```

es/Jan '24/10.1.24
$ node "r:\APTECH_Batches\2. MERN Stack Development BATCH 2\JavaScript ES-6\CodeFiles\Jan '24\10.1.24\Z_ReduceMethod.js"
prev : 5, next : 6
prev : 11, next : 7
prev : 18, next : 8
prev : 26, next : 34
prev : 60, next : 23
rahul@Romeo-Alpha09 MINGW64 /r/APTECH_Batches/2. MERN Stack Development BATCH 2/JavaScript ES-6/CodeFile
es/Jan '24/10.1.24
$ 

```

Number  
24

$\sqrt{0.5}$

$4 \cdot 9$  receivedValue

$2^2 = 4$

$(4)^{1/2} = \sqrt{4} = 2$

$(9)^{1/2} = \sqrt{9} = 3$  return

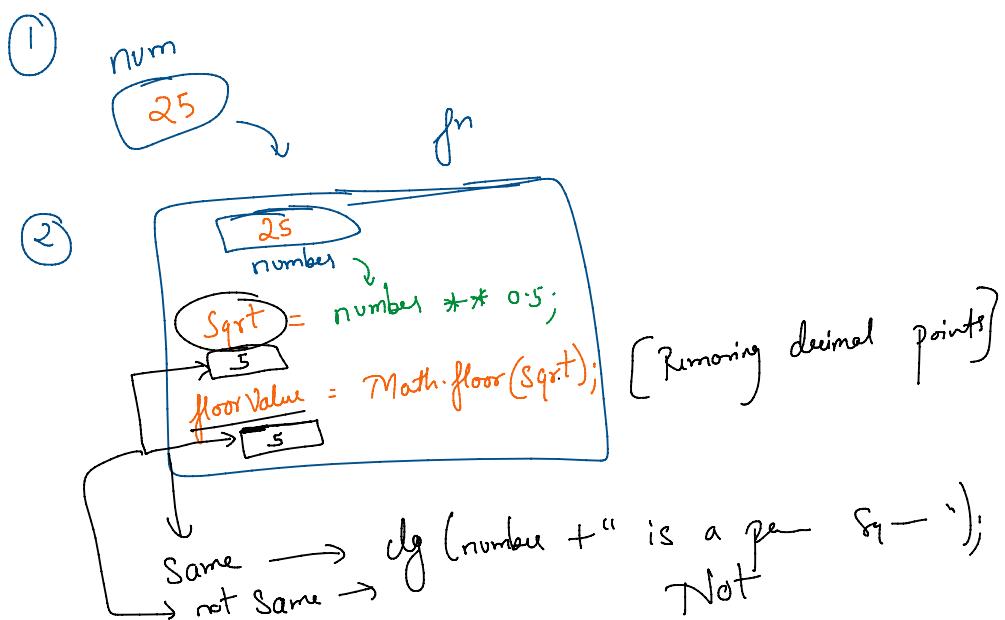
perfect square Yes

Not a perfect square No



$\sqrt{9} = \sqrt{9} = 3$  return  
 perfect square  
 Not a perfect square number.  
 No

num = 2  
 result = num \*\* 4;  
 [ num raised to 4 ]  
 [ 2 \*\* 4 ]





OUTPUT DEBUG CONSOLE TERMINAL PORTS

ods.js"

a : 2, b : 1 >0, b should come first [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ]

a : 3, b : 2 >0, b [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ]

a : 12, b : 3 >0, b should [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ]

a : 4, b : 12 <0, a - [ 1, 2, 3, 4, 12, 5, 6, 7, 8, 23 ]

a : 4, b : 3 >0, b - [ 1, 2, 3, 4, 12, 5, 6, 7, 8, 23 ]

a : 4, b : 12 >0, [ 1, 2, 3, 4, 12, 5, 6, 7, 8, 23 ]

a : 5, b : 3

a : 5, b : 12

a : 5, b : 4

a : 6, b : 4

a : 6, b : 12

a : 6, b : 5

a : 7, b : 4

a : 7, b : 6

a : 7, b : 12

a : 8, b : 5

a : 8, b : 7

a : 8, b : 12

a : 23, b : 5

a : 23, b : 8

a : 23, b : 12

[

Diagram:

A handwritten diagram on the right side of the terminal window shows a sequence of numbers from 1 to 23. The numbers 1, 2, and 3 are circled with a red circle labeled "1". The number 12 is circled with a red circle labeled "2". The number 4 is circled with a red circle labeled "3". The sequence continues as 5, 6, 7, 8, 23. Above the sequence, there are three numbered circles (1, 2, 3) with arrows pointing to the circled numbers 1, 2, and 3 respectively. Below the sequence, there are three sets of brackets: [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ], [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ], and [ 1, 2, 3, 4, 12, 5, 6, 7, 8, 23 ]. To the left of the sequence, there is handwritten text: "a : 2, b : 1 >0, b should come first [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ]", "a : 3, b : 2 >0, b [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ]", "a : 12, b : 3 >0, b should [ 1, 2, 3, 12, 4, 5, 6, 7, 8, 23 ]", "a : 4, b : 12 <0, a - [ 1, 2, 3, 4, 12, 5, 6, 7, 8, 23 ]", "a : 4, b : 3 >0, b - [ 1, 2, 3, 4, 12, 5, 6, 7, 8, 23 ]", and "a : 4, b : 12 >0, [ 1, 2, 3, 4, 12, 5, 6, 7, 8, 23 ]". Below these, the line "a : 5, b : 3" is underlined.

Diagram Summary:

>0 → No Swap

<0 → Swap

$> 0 \rightarrow$  No Swap  
 $< 0 \rightarrow$  Swap

