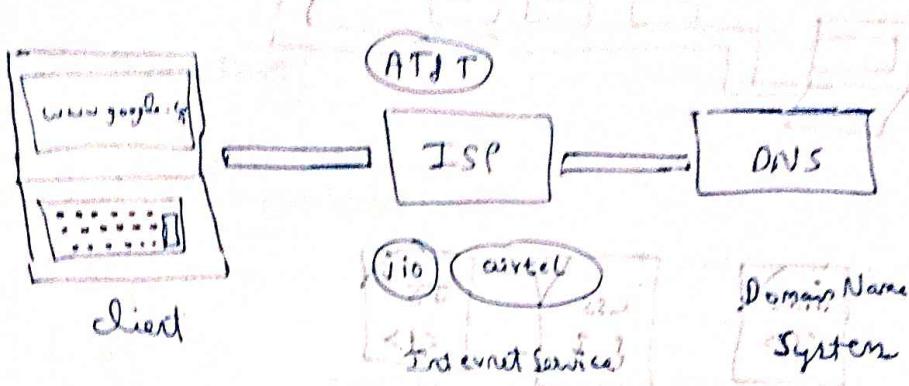


FULL STACK DEVELOPER

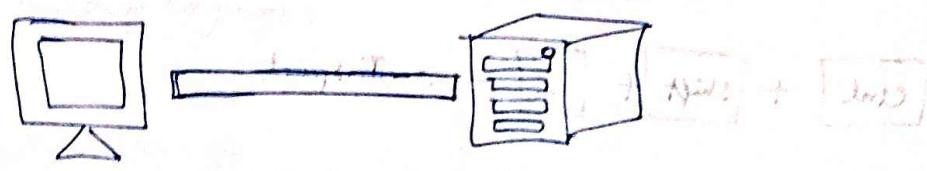
EACH AND EVERY WEBSITE HAS INDIVIDUAL IP ADDRESS.

For ex. IP 142.251.32.14 → www.google.com



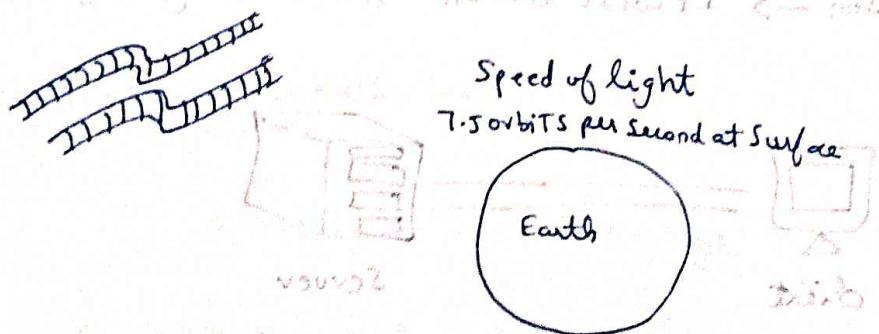
→ whatever we type in Google → jio and airtel (ISP) will know that :-)

The Internet



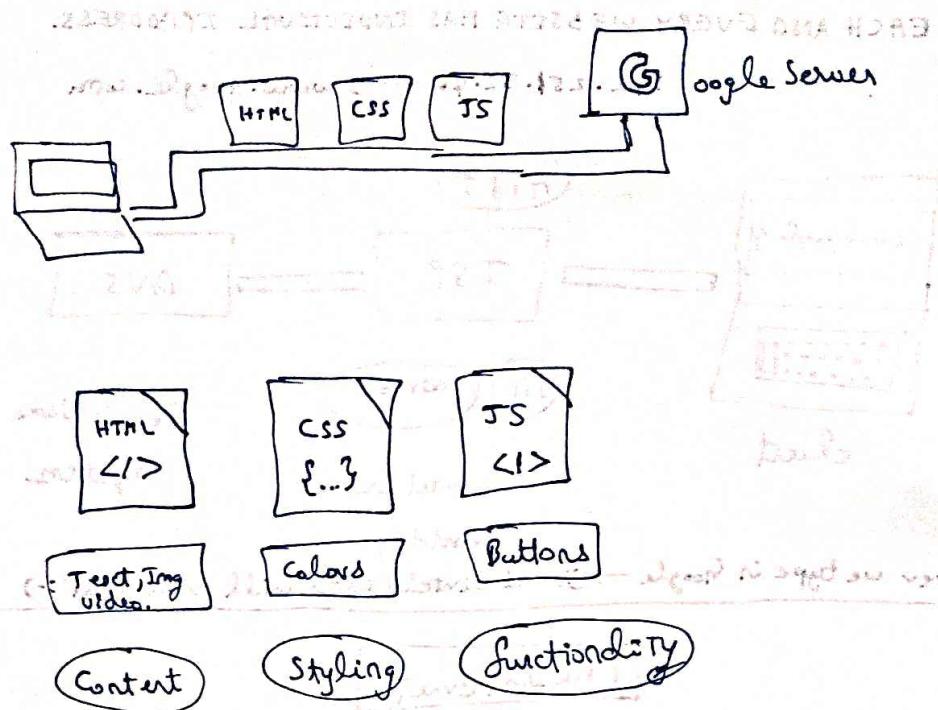
Submarine cables map - com → To see underwater cables

Every time we reload or click a button → Tiny electric signals passing under the water.



Every Time we reload or click a button → Tiny electric signals passing under the water.

How do websites work

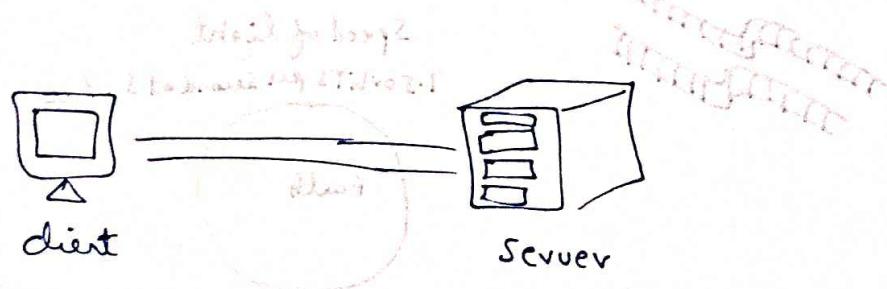


`ctrl + shift + I` → Inspect

we can edit anything in the website

but if we click Refresh, it will again get back to original

Reason → It will fetch information again from server.



Client will contact server no browser wait page
will start after above process done

FRONTEND WEB DEVELOPMENT

HTML

(Tag Based Language)

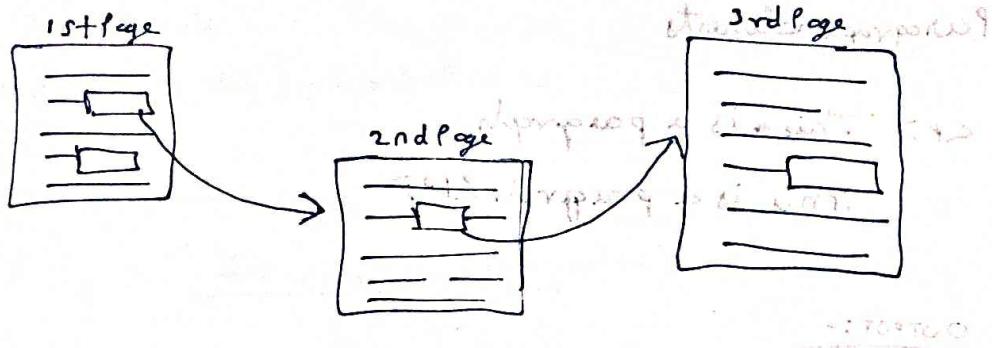
HTML

HTML → Hypertext Markup language

① what is Hypertext?

information about

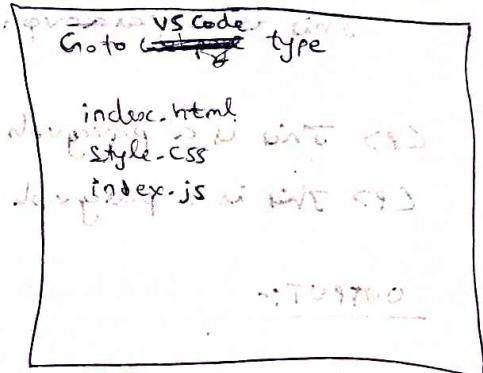
HyperText ↗ hyperLinks



② what is Markup language?

① English Language

② HTML Tag



HTML Heading Elements

heading Tags $\langle h1 \rangle$ Hello World $\langle /h1 \rangle$

opening Tag

Closing Tag

① $\langle h1 \rangle$ to $\langle h6 \rangle$ ~~Ch~~

→ which is the Biggest Tag $\langle h1 \rangle$

② $\langle h1 \rangle$ to $\langle h6 \rangle$ → To write index pages in book.

Don't

<h1> Title </h1>
<h3> Subtitle </h3>

Don't go from h1 to h3

Do

<h1> title </h1>
<h2> subtitle </h2>

h1 → h2

? margin in b/w

Paragraph Elements

<p> This is a paragraph

This is a paragraph </p>

OUTPUT:-

This is a paragraph This is a paragraph

? margin exist in b/w

<p> This is a paragraph </p>

<p> This is a paragraph </p>

OUTPUT:-

Output HTML

This is a paragraph

size-> This is a paragraph

standard white space

<p>

[

<td> block allow <td> <td> <td> standard

allow

allow

allow

</p>

[

Default Space

</p>

[

<td> get target all sideless <

</p>

[

allow in every webkit browser < - can't < /td >

Non-void element `<hr>` `</hr>`

void element `
`

visit start but finite

`<p>` This is a paragraph `</p>`

`
`

`<p>` This is a paragraph `</p>`

visit start but queue full `<div>`

visit end but queue full `</div>`

Output:- this is a paragraph

`<div> visit now open as <div>`

this is a paragraph

attaching item `<div>`

`</div>`

`
` → break

`<div>` with this separates `<div>`

`
` → to add horizontal line.

visit registered line `<div>`

`<div>` with this type of line `<div>`

`<div>` with this separated `<div>`

Intermediate List (HTML)

``

``

`` item 1 ``

``

The List Element.

List

Unordered List

list item has no order

Unordered list

``

` Milk `

list item has no order

any of both `Milk`

list item has no order `Eggs`

list item has no order `Flour`

` Flour `

list item has no order `Yellow`

``

Ordered list

``

` Milk `

`` 1) Milk or 2) Eggs

2) Eggs

` Eggs `

3) Flour

` Flour `

``

Nesting and Indentation

 Wake up and Brush teeth

 Drink 500ml warm water

 Do yoga for an hour

 Make omelette

 whisk eggs with Milk

 Add butter to pan

 Add in eggs and Stir

 When solid add salt

 Start work

OUTPUT:-

• wake up and brush teeth

• Drink 500ml warm water

• Do yoga for an hour

• Make omelette

◦ whisk egg's with milk

◦ Add butter to pan

◦ Add in egg's and Stir

◦ When solid add Salt

• Start work.

Nesting → inside or

Anchor Elements

<a> This is a link <a> (X)

<\ "div" = over grid

 this is a link <a>

<a> this is a link to Google <a>

OUTPUT:-

This is a link to Google

spani with fo email

This is a link to Google

<a>

OUTPUT:-

This is a link to Google

This is a link to Google

<a>

OUTPUT:-

This is a link to Google

we can drag this

draggable

spani with fo email

Image Elements

``

→ `src` attribute → location of the image

``

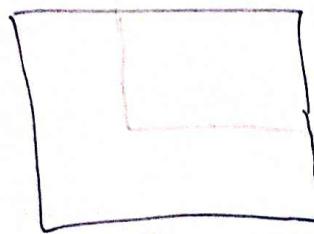
↓ ↳ self closing tag (void elements)

Source of the image

``

→ "https://i.imgur.com/11.png" is provided

Output:



Rend. image

``

Audio will read what is in alt.

KM. (D) (D) with

[with alt text]

``

Self closing Tag

alt → Is not visible in Webpage

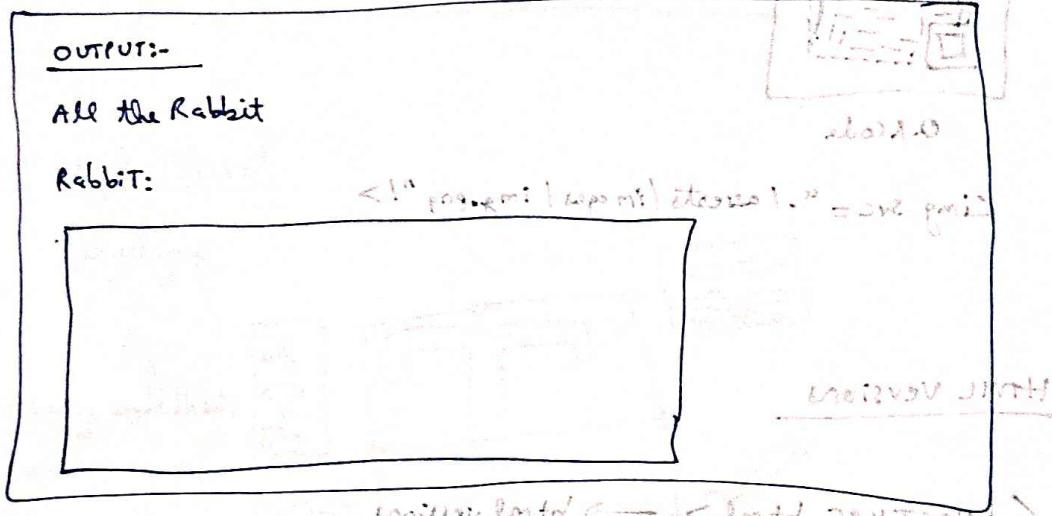
PROJECT

Birthday invite

Category <model, view>
Multipage websites

paths: C://Folder / Full stack

<h2> Rabbit </h2>



① what are webpages?

Project

index.html

images

img.png

0.5MTH

1.5MTH

2.5MTH

3.5MTH

index.html

about.html

sub

sub

sub

folder

project

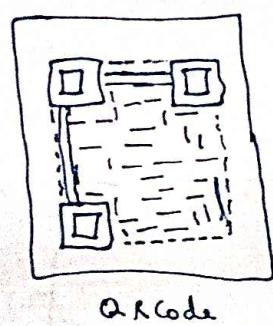
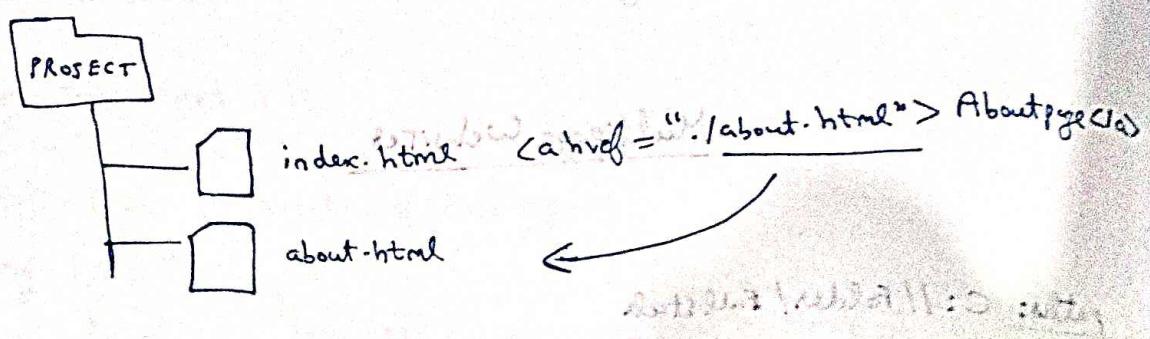
index.html

about.html

users

img

img.png



QR Code

``

``

alt="QR code"

QR code with URL

QR code

``

HTML Versions

`<!DOCTYPE html>` → HTML versions

`<html lang="en">`

↓
English

HTML

1991

→ example site: www.w3.org

HTML+

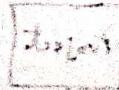
1993

HTML2.0

1995

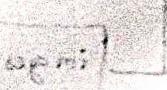
HTML3.2

1997



HTML4

1999



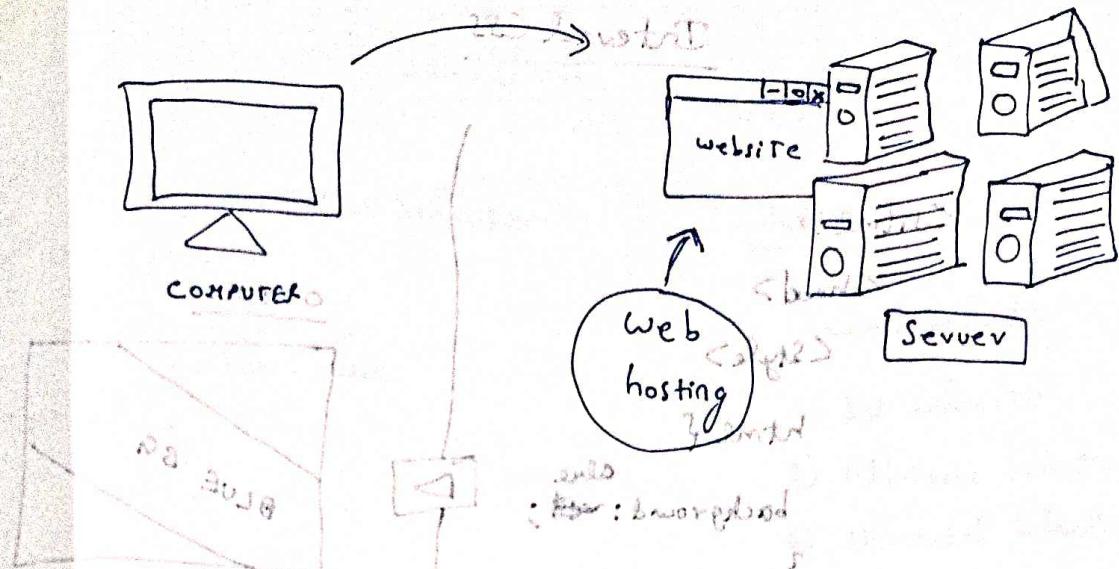
HTML5

2012

`<head>`
`</head>`
`<body>`
`</body>`

`</html>`



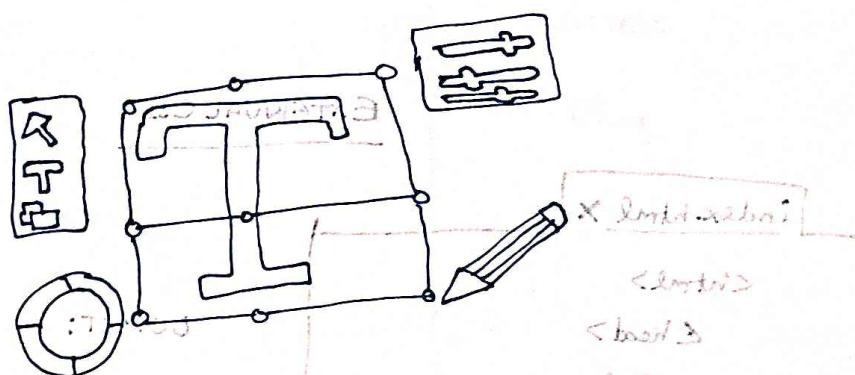


Cascading Style Sheets

Style Sheet

Language

- Cascading Style Sheets
- SASS
- LESS
- ...



- 1) Inline CSS
- 2) Internal CSS
- 3) External CSS

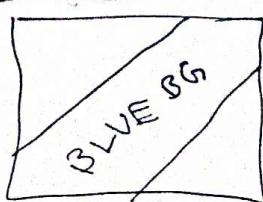
Inline CSS

```
<html style="background: blue">
```

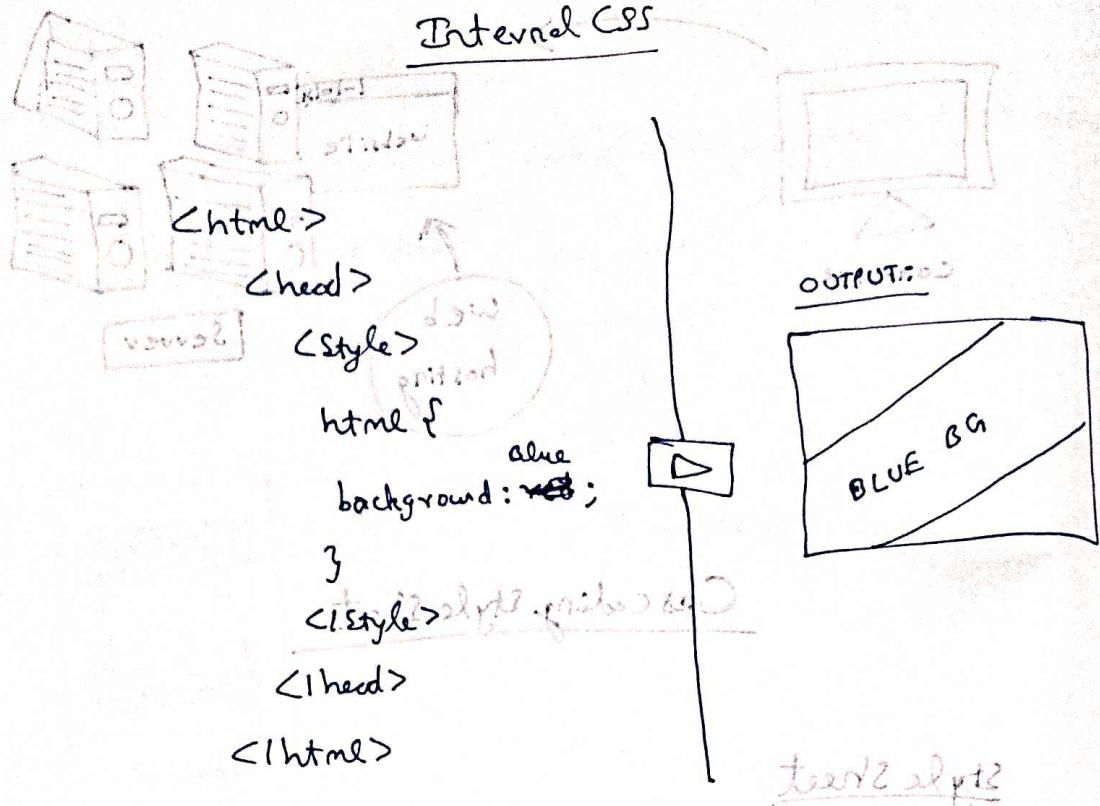
```
</html>
```

{ code: browserized }

OUTPUT:

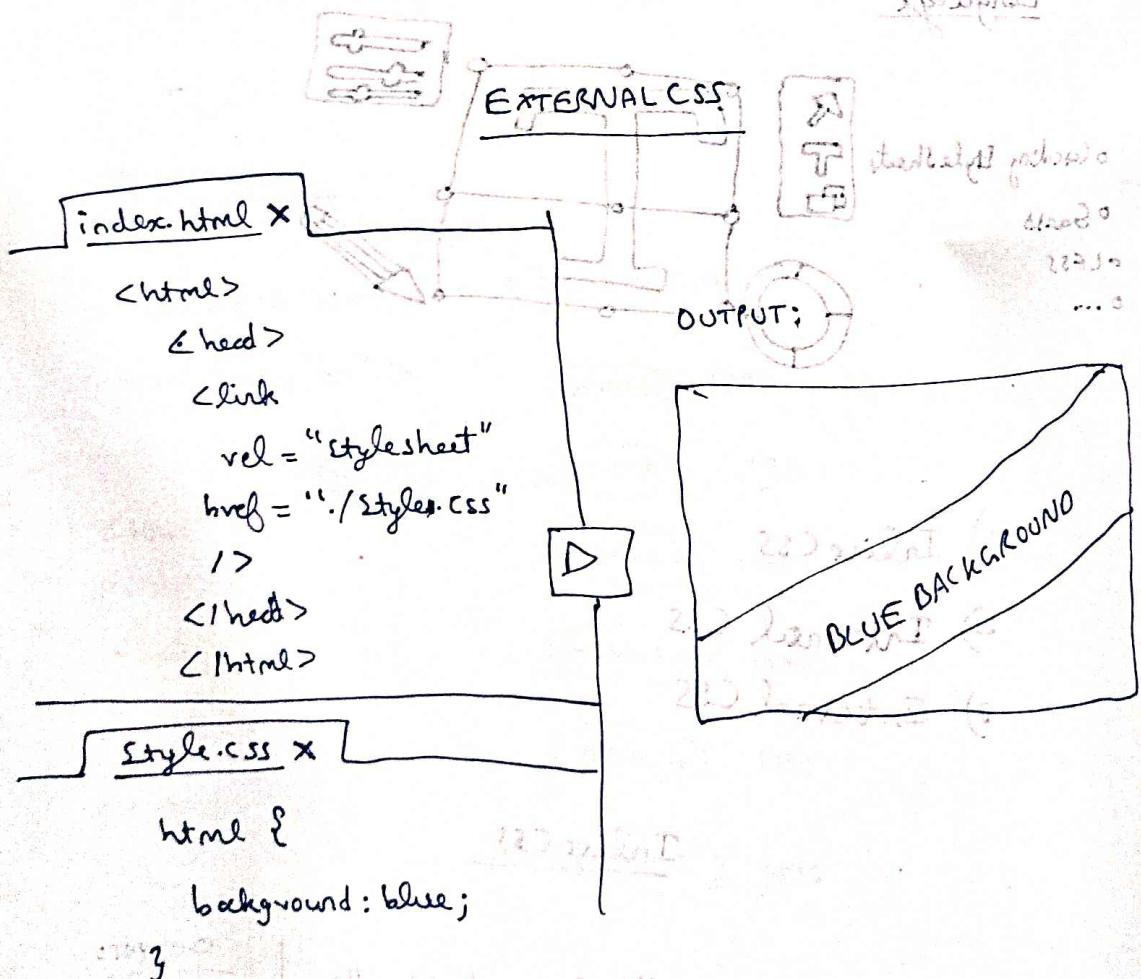


Internal CSS



tree & style

EXTERNAL CSS



Internal CSS = External CSS

css

Selectors

(below) h1 {
color: blue;
}

Types of Selectors

- 1) class selector
- 2) ID selector
- 3) Attribute Selector
- 4) Universal Selector.

Program

[index.html] <h2> Red </h2>
<h3> Green </h3>
<h4> Blue </h4>

Style.css

h2 {
color: red;
}

Red (turn red color)

Green

Blue (no effect)

index.html

<h2 class="red-text"> Red </h2>
<h2> Green </h2>
<h2> Blue </h2>

Style.css

.red-text {
color: red;
}

CLASS SELECTOR

Red (turn red)

Green

Blue

index.html

<h2 class = "red-text"> Heading 2 </h2>

<h3> Heading 3 </h3>

<p class = "red-text"> Paragraph </p>

style.css

```
selector {  
    .red-text {  
        color: red;  
    }  
}
```

Heading 2 (turned red)

Heading 3

Paragraph (turned red)



(X) 1. ID selector <h2 id = "main">

Color: red

<h2 id = "main">

ID SELECTOR

index.html

<h2 id = "main"> Red </h2>

<h2> Green </h2>

<h2> Blue </h2>

style.css

```
#main {
```

```
    color: red;
```

```
}
```

Red (turned red)

Green

Blue



id = unique

ID vs class

- one element

- many elements

Attribute Selector

```
p[draggable]{}
```

color:red;

}

color:red

color:red

index.html

```
<p draggable="true">Drag me</p>
```

```
<p draggable="false">Don't Drag me</p>
```

```
<p draggable="false">Don't Drag me</p>
```

Styles

```
p[draggable="true"] {
```

color:red;

}



Drag me
Don't Drag me
Don't Drag me

Universal Selector

```
*
```

color:red [All three get]

}

index.html

```
<h1 class="title">Hello</h1>
```

```
<h2 id="heading">World</h2>
```

```
<p draggable="true">
```

this is a website

1103

Styles

```
*
```

color:red;

z



Hello (true red)
World (true red)
this is a website
(true red)

CSS Properties

css colors

```
html {
```

```
background-color: red
```

```
}
```

```
h1 {
```

```
color: blue
```

(hex code) 

www.w3schools.com → for more colours

rgb, hex

Font Family

```
h1 {
```

```
font-family: Helvetica, sans-serif
```

```
}
```

```
h2 {
```

```
font-family: "Times New Roman", serif
```

```
}
```

[font exact (2)] horiz.

```
h1 {
```

```
text-align: center
```

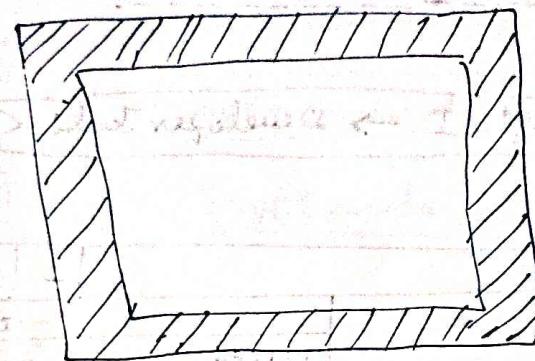
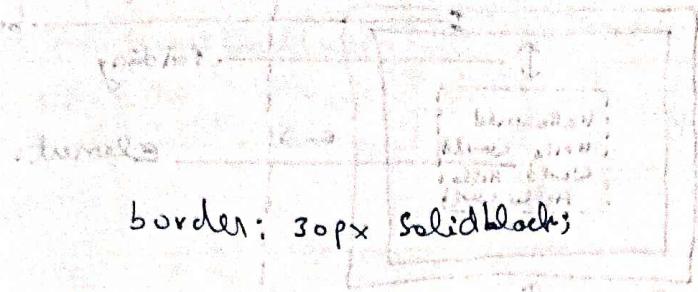
```
}
```

text-align: left; text-align: right;

text-align: center; text-align: justify;

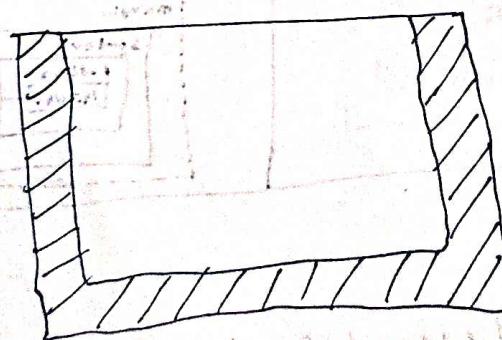
CSS Repetition

ctrl + shift + I → Developer Tools



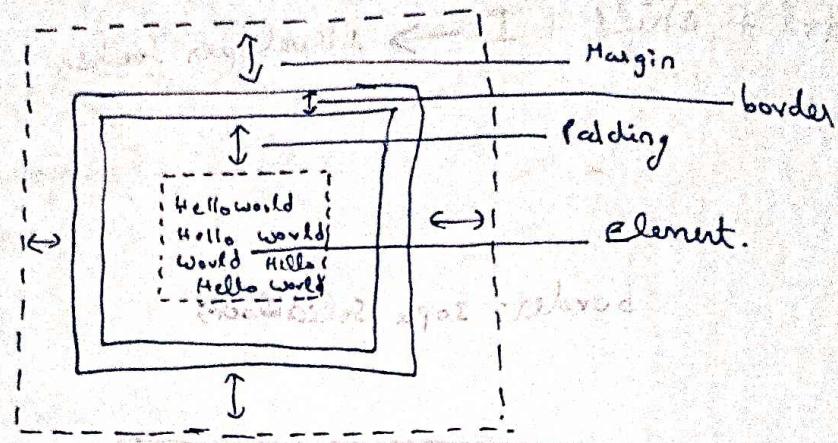
border: 30px solid black;

border-top: 0px

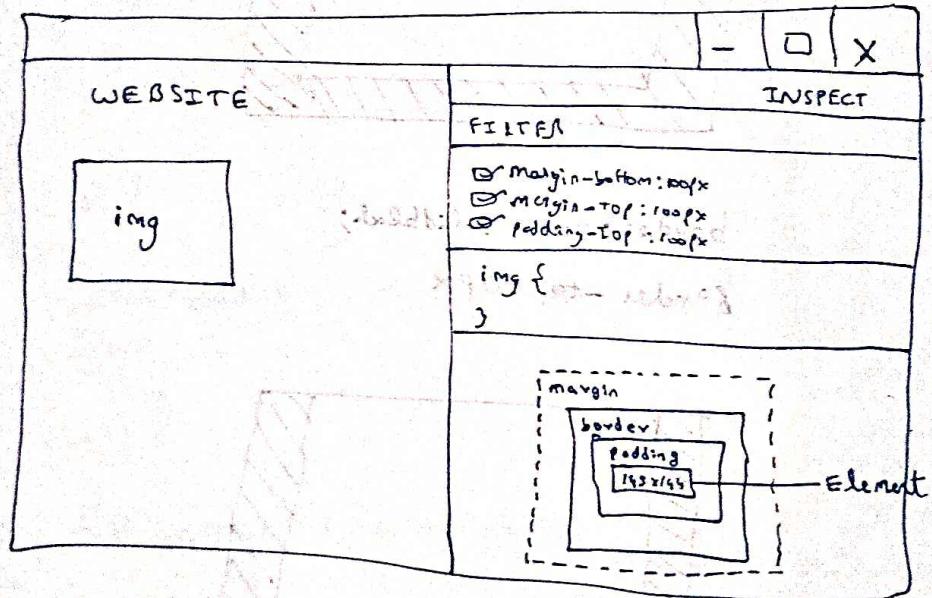


border-width: $\frac{0px}{\text{top}}$ $\frac{10px}{\text{right}}$ $\frac{20px}{\text{bottom}}$ $\frac{30px}{\text{left}}$

Margin / border / padding / Element

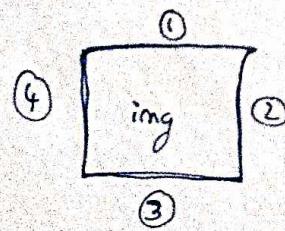


$\Sigma \text{ Ctrl} + \text{Shift} + \text{I} \rightarrow \text{Developer Tools}$



Margin / border / padding / Element.

① 0px ② 10px ③ 20px ④ 30px
① ② ③ ④ → Values



(0px) 10px 20px 30px
① ③ ④ ②
top/bottom left/right

<body>

```
<div>
  <p> Hello world </p>
  <img src = "./HelloWorld.png" />
</div>
```

```
<div>
  <p> Good Night world </p>
  <img src = "./GoodNightWorld.png" />
</div>
```

</body>

Hello world

img

Good Night world

img

<div> <div> → To Create Separate Sections

CSS Intermediate

CSS Cascade

index.html x

```
<ol>
  <li> One <li>
  <li> Two <li>
  <li> Three <li>
</ol>
```

3 styles. 3 rules.

color: red;

① position

② Specificity

③ Type

④ Importance.

style.css x

```
li {
```

```
  color: green; }
```

```
}
```

```
color: blue; }
```

inline CSS → only one particular line

internal CSS → only one particular webpage

external CSS → appears in all HTML files (link)

Q11.2

(Q) `<h1 class="a-class" id="an-id">Hello </h1>`

1. `a-class {
color: blue;
}`

VS

2. `#an-id {
color: green;
}`

{ default style sheet at 6.2.2.2 }
(Q) `<h1 class="a-class another-class">Hello </h1>`

1. `a-class {
color: green;
}`

3

VS

2. `.another-class {
color: blue;
}`

3

(Q)

`<html lang="en">`

`<head>`

`<style>#an-id {color: green}</style>;`

`</head>`

`<h1 id="an-id" style="color: green"></h1>`

`<body>`

`<h1 id="an-id" style="color: blue;">Hello </h1>`

`</body>`

`</html>`

1

Blue

VS

Green.

CSS Intermediate Selectors

Combining CSS Selector.

index.html

```

<p> YellowText </p>
<div class = "box inner-box">
    <p> WhiteText </p>
</div>

```

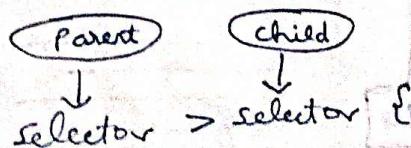
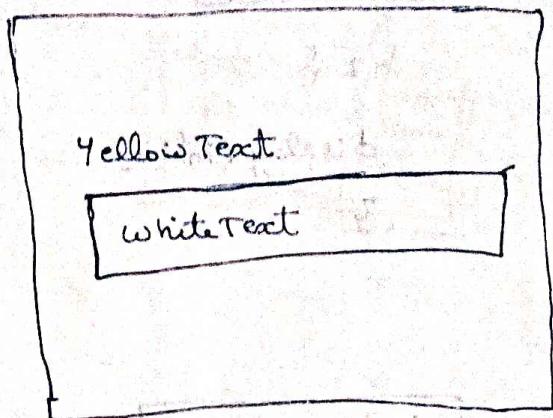
style.css

```

p {
    color: yellow;
}
.innner-box p {
    color: white;
}

```

OUTPUT:



```
color: firebrick;
```

}

CSS position

Relative. absolute. fixed. static Positioning

Advanced CSS

CSS Display

index.html x

<h2> Harry </h2>

<h2> Patter </h2>

style.css x

h2 {

display: inline;

}

style.css x

h2 {

display: block;

index.html x

<p> text... </p>

style.css x

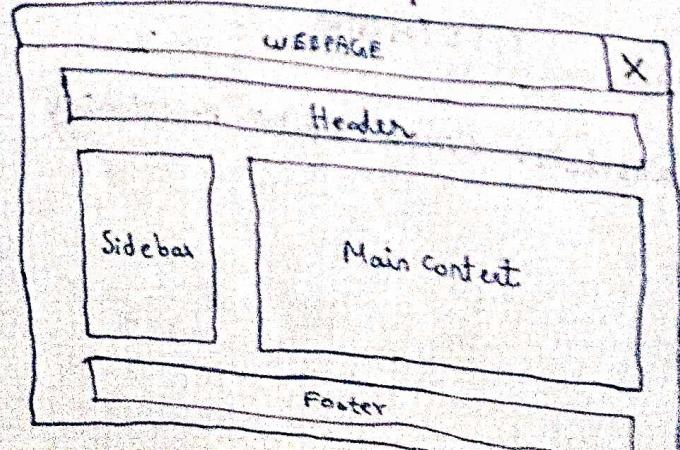
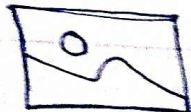
img {

float: left;

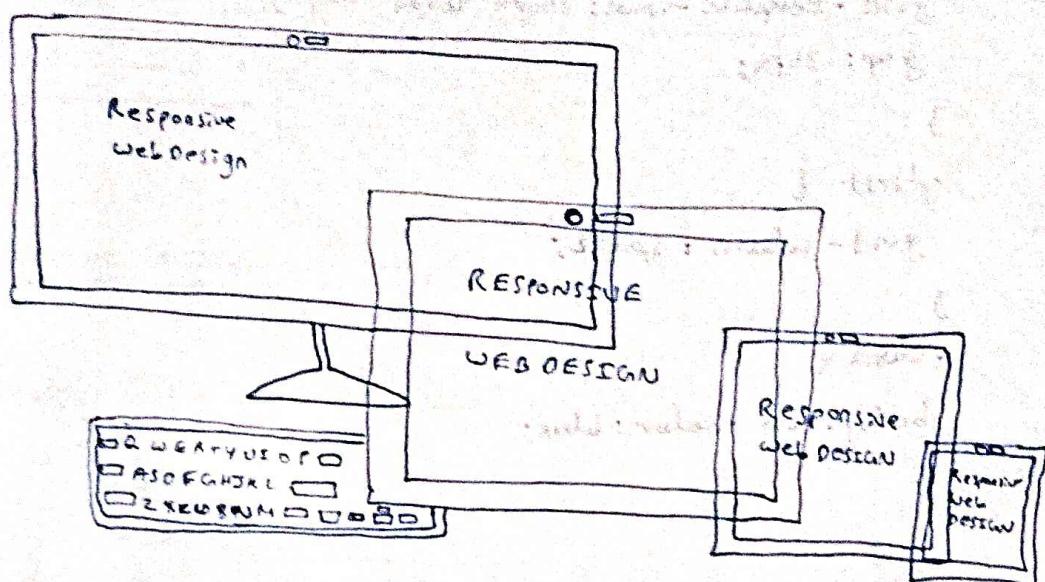
}

style.css x

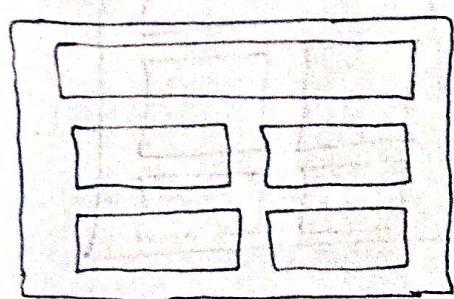
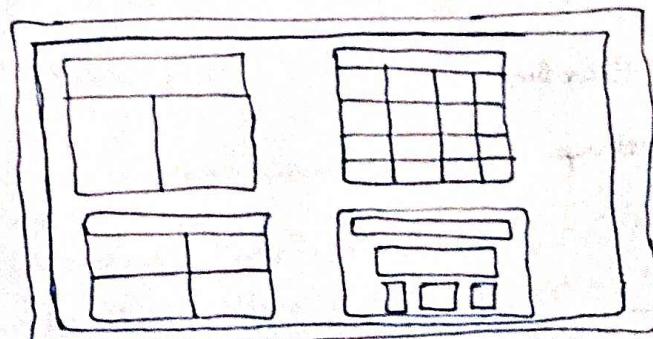
OUTPUT



Responsive web design → fits in all devices



CSS Grid



```
<div class = "grid-container">
  <div class = "first-card"></div>
  <div class = "card"></div>
  <div class = "card"></div>
  <div class = "card"></div>
</div>
```

• grid container {

display: grid;

grid-template-columns: 1fr 1fr;

grid-template-rows: 100px 200px 200px;

gap: 30px;

}

• first {

grid-column: span 2;

}

- card {

background-color: blue;

}

① Media Queries

② CSS Grid

③ CSS Flex Box

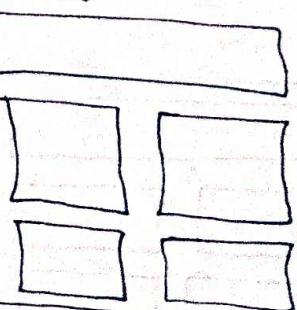
④ Bootstrap

OUTPUT:

Media Query



CSS GRID



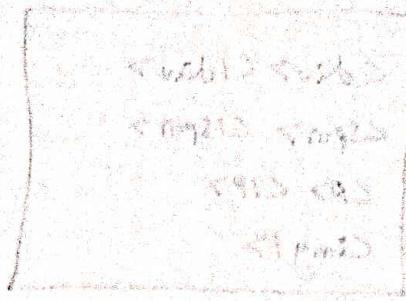
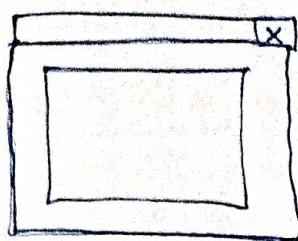
CSS FLEX BOX



Bootstrap Framework



Media Query



Web Design Agency

Creative Design → Graphics

FLEXBOX

CSS Flexbox (Flexible Box)

and also it's used

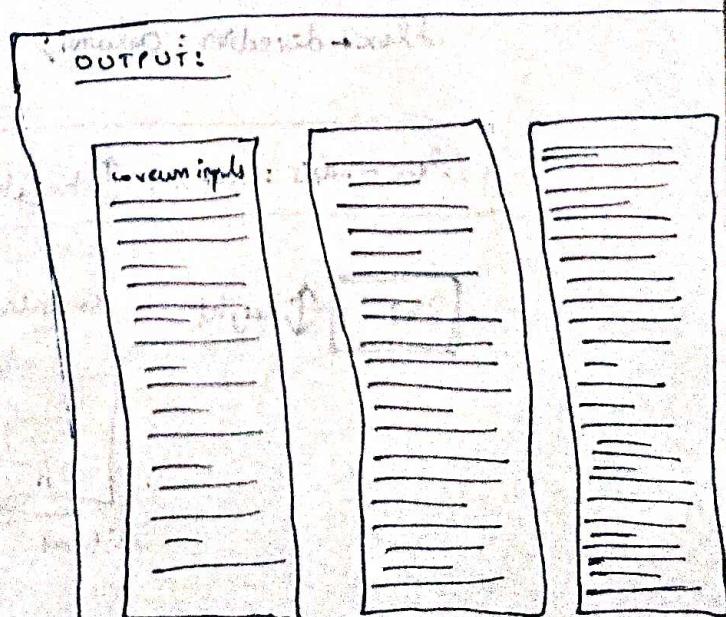
index.html x

```
<div class="container">
  <div class="one"><p>....</p></div>
  <div class="two"><p>....</p></div>
  <div class="three"><p>....</p></div>
</div>
```

Style.css x

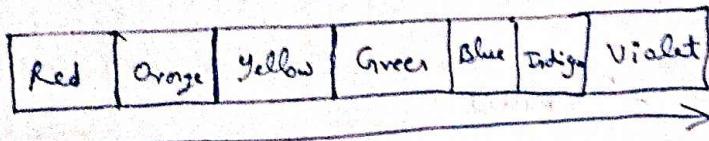
```
.container {
  display: flex;
  gap: 10px;
```

3



```
<div> </div>  
<span> </span>  
<p> </p>  
<img />
```

Flex direction



flex-direction main-axis
Left → Right

Red
Orange
Yellow
Green
Blue
Indigo
Violet

Colour of Rainbow

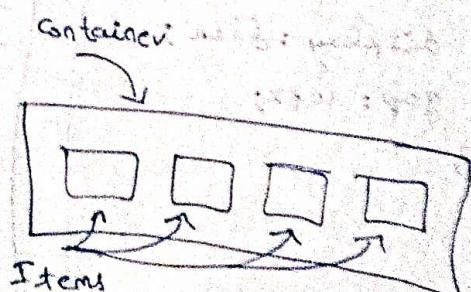
ROYGBIV

UIOBGYOR

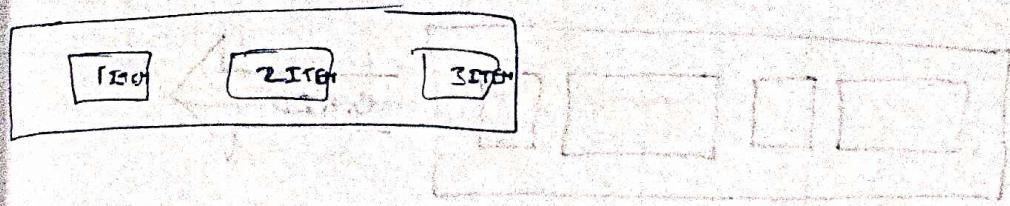
flex-direction: column;

flex-basis: 100px; ↑ height

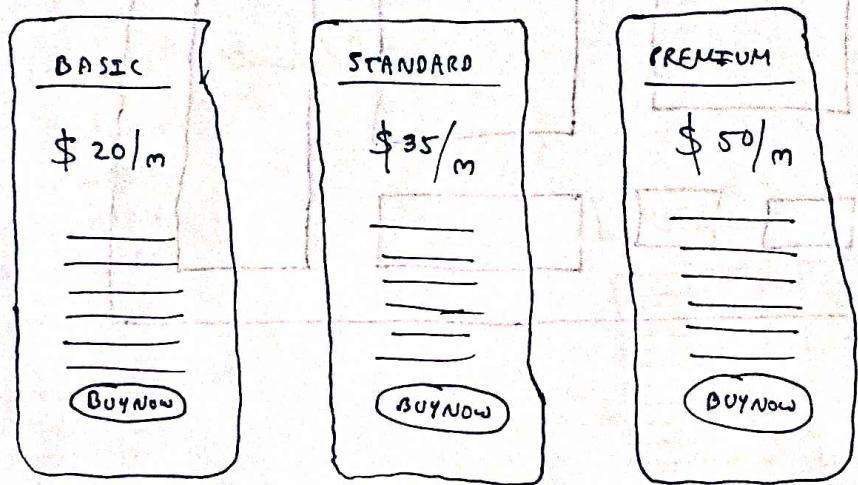
Red ↑ height



Flex Sizing

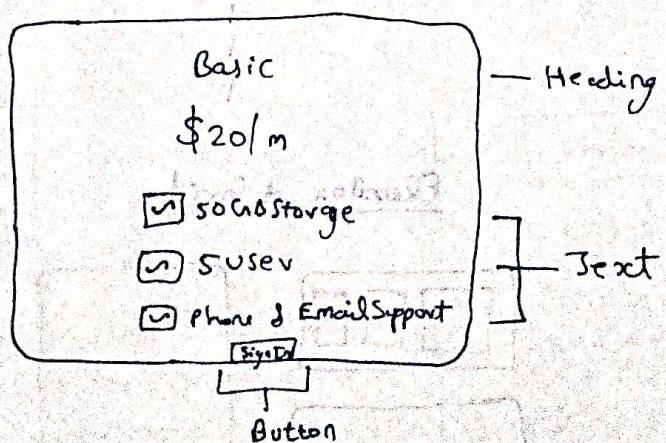


PROJECT PRICING TABLE

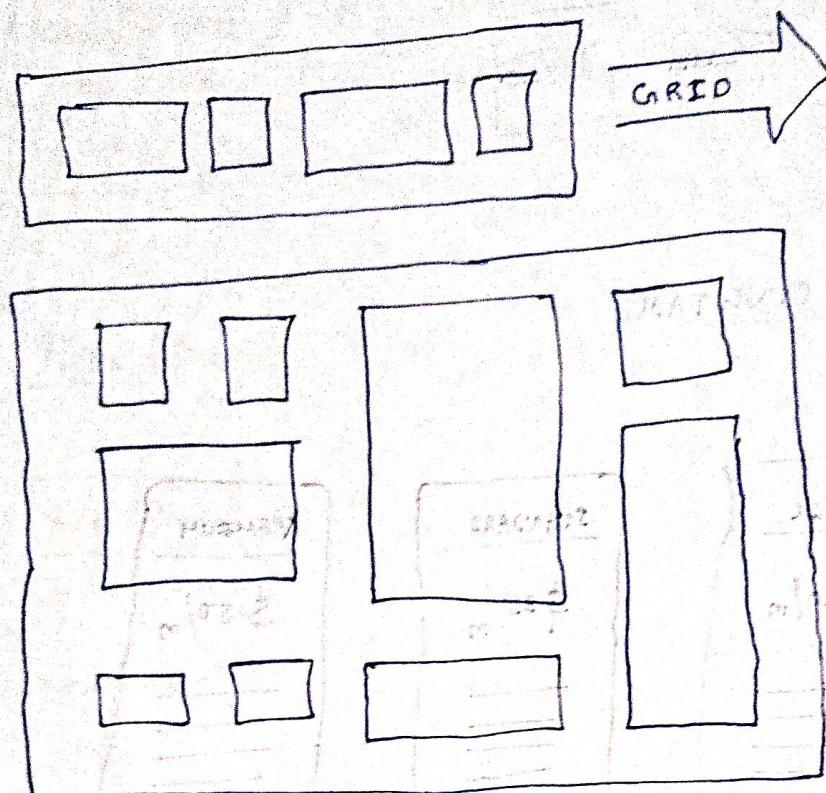


FLEXBOX → FEELXIBLE BOX

(CHANGING THE WIDTH DYNAMICALLY)



Grid



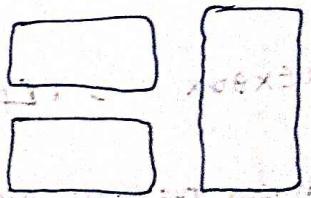
Flex Box



1D Layout

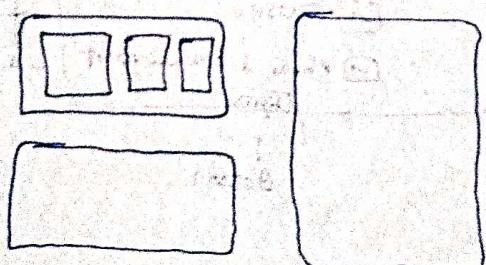
VS

Grid

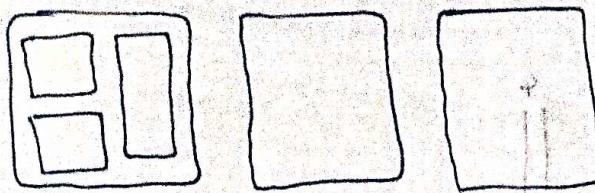


2D Layout

Flexbox + Grid



Grid + FlexBox



How To Create a Grid

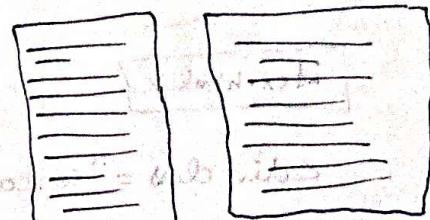
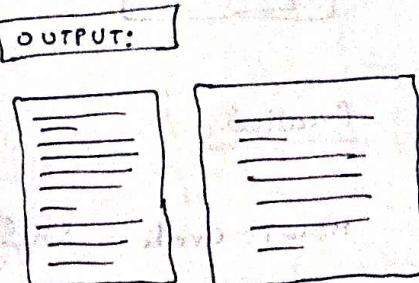
index.html X

```
<div class = "container">
  <p>....</p>
  <p>....</p>
  <p>....</p>
  <p>....</p>
</div>
```

Style.css X

.container {

display: grid;
 grid-template-columns: 1fr 2fr;
 grid-template-rows: 1fr 1fr;
 gap: 10px;



}

Grid Sizing

index.html X

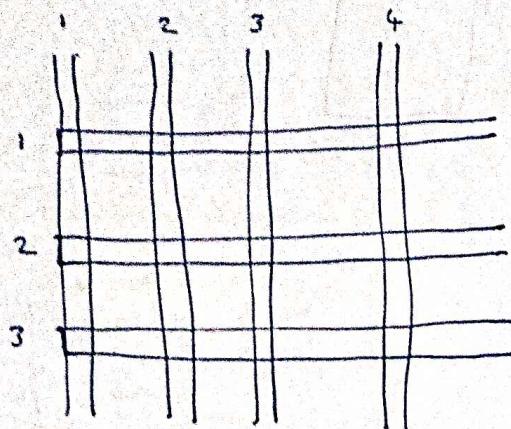
```
<div class = "grid - container">
  <div class = "grid - item">1</div>
  <div class = "grid - item">2</div>
  <div class = "grid - item">3</div>
  <div class = "grid - item">4</div>
</div>
```

Style.css X

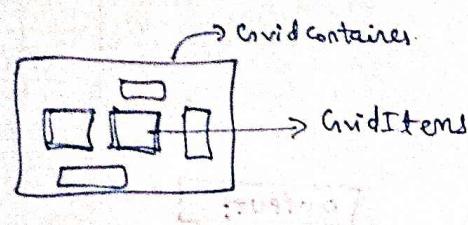
.grid-container {
 display: grid;
 grid-template: 100px 200px/
 400px 800px;



Grid Placement



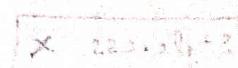
bins = item slot



"binets" = cells slots

<grid>
<grid>
<grid>
<grid>

<grid>



grid-template-columns: 1fr 1fr 1fr 1fr;

Practical

How To Create a Shelf / Cupboard / Library

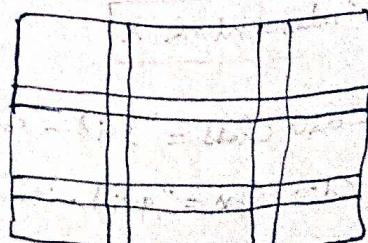
index.html X

```
<div class = "container">
    <div class = "item cowboy" > </div>
    <div class = "item astronaut" > </div>
    <div class = "item book" > </div>
</div>
```

Style.css X

```
.container {
    height: 100vh;
    display: grid;
    gap: 3rem;
    grid-template-columns: 1fr 1fr 1.5fr;
    grid-template-rows: 1fr 1fr;
```

OUTPUT:



3rem → gap between items



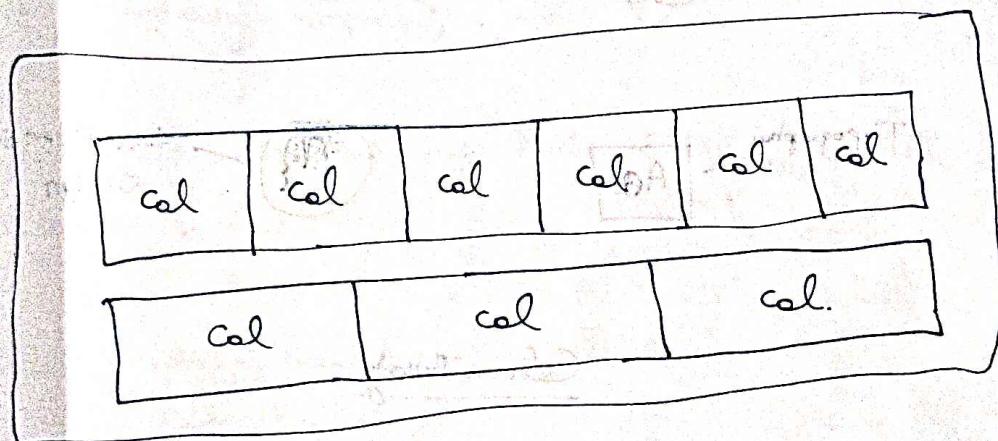
Bootstrap

Developed by twitter guys

Bootstrap → inbuilt CSS files

CDN → Content Delivery Network

Bootstrap Layout



TV Size

- Extra small < 576px
- Small > 576px
- Medium > 768px
- Large > 992px
- X-large > 1200px
- XX-large > 1400px

Mobile

iPod

Laptop

Desktop

TV

Theatre

SQL → Court & Suit
NoSQL → Formless T-Shirts

Web Design School.

Web design → Colours, Text, Aesthetics

Colour
theory



User Interface
design

Typography



User Experience
design

Colour Theory

flowers
SHOW YOUR LOVE
PAINTBALL
for the daring

BORING (X)

flowers

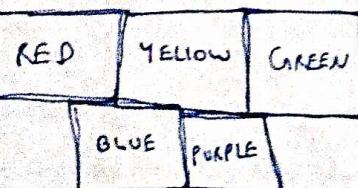
SHOW YOUR LOVE

PAINTBALL

for the daring

INTERESTING (1)

MOODS



Colors

Red → Energetic

Yellow → Joy, Attention

Green → Freshness, Safety, Growth

Blue → Stability, Trust

Purple → Royalty, Wealth, Femininity

Red (Car Company)

Yellow (Fitness Company)

Green (Grocery Company)

Blue (Payment Company)

Purple (Insurance Company)

Complementary Colours → Make Things pop out

(Green & Red)

Website: Adobe color.com
colorhunt.co

Typography

(TEXT)

You'll always
be mine... ❤



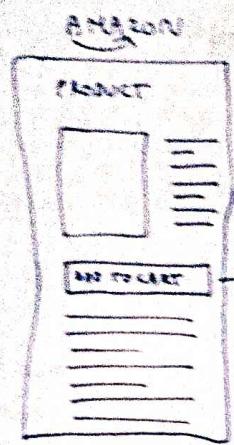
YOU'LL ALWAYS
BE MINE...



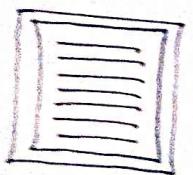
CHOOSE THE CORRECT FONT.

Different font → Different Meaning

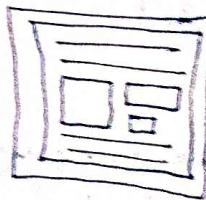
UI Design



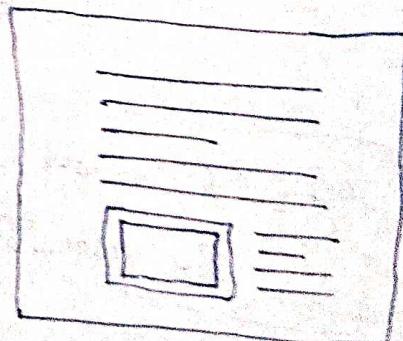
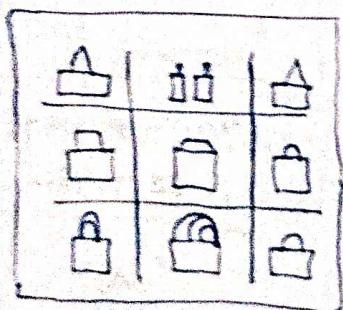
Layout



vs.



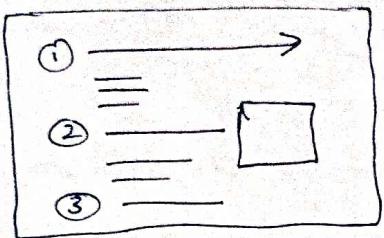
WhiteSpace



UX Design

Simplicity

F-layout → Eyes moves from left to right.



Website: canva.com.

JavaScript

Power of Webpages

Brenden → How To Make a language in 10 Days

(JavaScript)

Java vs JavaScript

Java = 1.5 years
JavaScript = 10 days

Interpreted

- JavaScript
- Python
- Ruby

Compiled

- Java
- C/C++
- Swift

iOS → Swift

Android → Java

Edit → JavaScript Console

alert("Hello"); S
alert("world"); S

say("Hello"); X

window.alert(); → displays alert dialog.

String
alert("Hello");

Boolean
→ True
→ False

Java Script Variables

Var myname = "Computer";

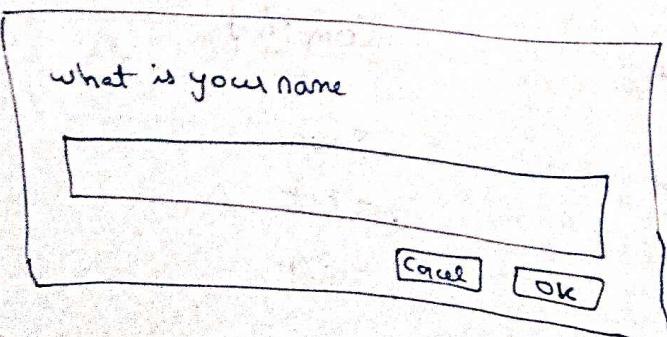
myname = "Science";

alert(myName);

Var myName = "Computer";

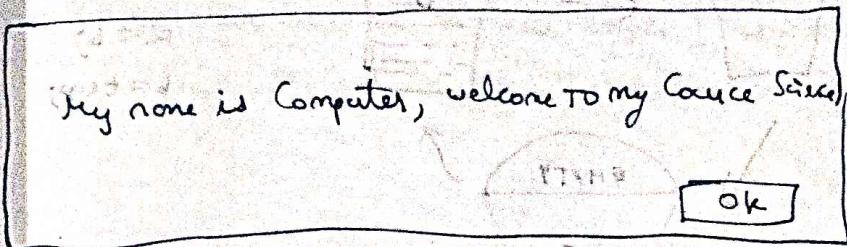
Var yourName = prompt("What is your name?");

OUTPUT:



`alert (" My name is " + myName + ", welcome to my course " + yourName + "!");`

Output:



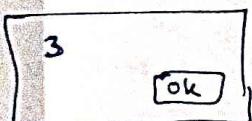
`var gonelevel = 1;`

`gonelevel = 2;`

`gonelevel = 3;`

`alert(gonelevel);`

Output:



Problem :

Given to the existing code below.

`Var a = "3";`

`Var b = "8";`

a holds value of "8")

b holds value of "3";

a is 8

b is 3

Code Selection

`function test () {`

`var a = "3";`

`var b = "8";`

`var temp = a;`

`a = b;`

`b = temp;`

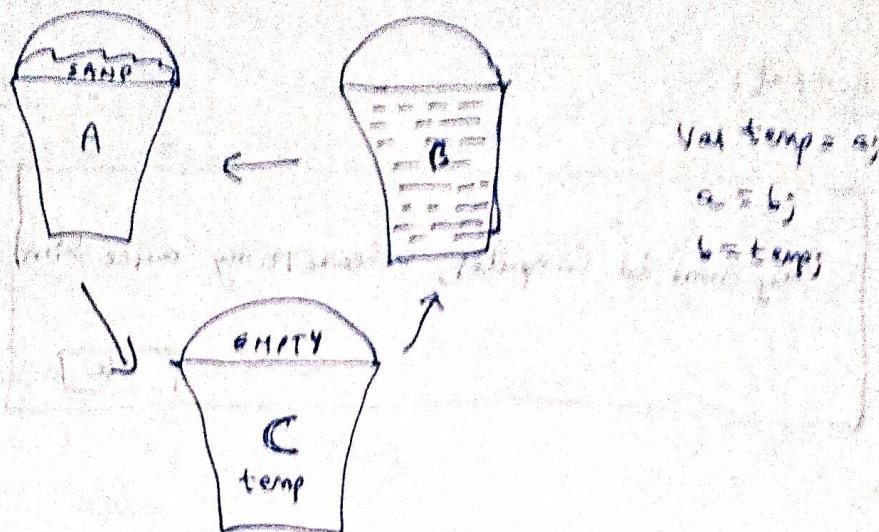
`console.log ("a is "+a);`

`console.log ("b is "+b);`

`}`

① All test cases passed.

Explanation



var my Age = 12; **(X)**

var name = "Computer";
name.length;

Output
8

twitter

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

① character's Entered | 280 character's remaining

② Allows only 280 character's.

Java Script Code for Twitter

```
var tweet = prompt ("Compose your tweet");  
var tweetCount = tweet.length;  
alert ("You have written " + tweet.length + " characters, you have " +  
(140 - tweet.length) + " characters remaining.");
```

OUTPUT:

Compose your tweet.

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1360  
1361  
1362  
1363  
1364  
1365  
1366  

```

Var nome = "Margolin"

name := uppercase()

Var nome = "Margolin"

name := lowercase()

Number

Addition Var a = 3+2

Subtraction Var a = 3-2

Multiplication Var a = 3*2

Division Var a = 8/2

Modulo Var a = 8%2 / remainder

Var test = (3+5*2) % 10 (BODMAS)

[= 10]

Var teste = (3+5)*2 (BODMAS)

[= 16]

Var dogge = prompt("How old is your dog?");

Output:

How old is your dog?

[7]

[UK]

Increment & Decrement Operator

`x++;`

Increment

`x--;`

`var x = 5;`

`x++ // 6`

Decrement

`var x = 5;`

`x-- // 4`

Problem:

`Var x = 3`

`Var y = x++;` // X is first assigned to 4 so y = 3 (I made a mistake 😊)
`y += 1;`

o 4

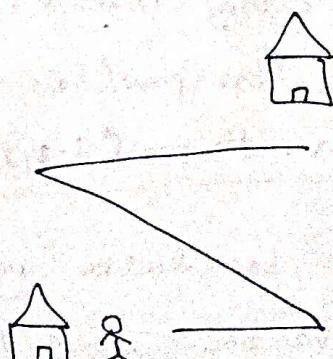
o 5

o 6

o 7

Functions

```
alert ("leave house");
alert ("move right");
alert ("move up");
alert ("Buy Milk");
alert ("move right");
alert ("enter House");
```



functions

get Milk() {

```
    alert ("");
    alert ("");
    alert ("");
    alert ("");
```

reusability of Code.
again & again.

move();

moveleft();

move right();

This function has some block of code somewhere.....

Math.floor(1.6); // to round off the number.

Math.round(20.016); // To ~~round~~ discard decimal points

BMI (Body Mass Index) Calculator Challenge:

$$\text{BMI} = \frac{\text{weight (kg)}}{\text{height}^2 (\text{m}^2)}$$

Program:

weight = 65
height = 1.8

```
function bmiCalculator(weight, height) {  
    var bmi = weight / (height * height);  
    return bmi;  
}
```

Instead of writing ($\text{height} * \text{height}$)

we can use $\text{Math.pow}(\text{base}, \text{exponent})$;
 $\rightarrow \text{Math.pow}(1.8, 2)$

```
function bmiCalculator(weight, height) {  
    var bmi = weight / (height * height);  
    return bmi;  
}
```

```
var bmi = bmiCalculator(65, 1.8);  
console.log(bmi);
```

OUTPUT: 20.06172

Intermediate Java Script.

```
var n = Math.random();  
console.log(n);
```

OUTPUT: 1.1623529

```
n = Math.floor(Math.random());  
console.log(n);
```

OUTPUT: 1

Love Calculator ❤️

```
prompt ("What is your name");  
prompt ("What is their name");
```

```
var loveScore = Math.random() * 100;  
loveScore = Math.floor(loveScore) + 1; + " %"  
alert ("Your love score is " + loveScore);
```

OUTPUT: Your love score is 19%.

Comparitor and Equality

== => is equal to

!= => is not equal to

> => is greater than

< => is less than

>= => is greater or equal to

<= => is less or equal to

```
var a = 1;  
var b = "1";  
typeof(a);  
"number"  
typeof(b);  
"string"  
if(a === b) {  
    console.log("yes");  
} else {  
    console.log("No");  
}
```

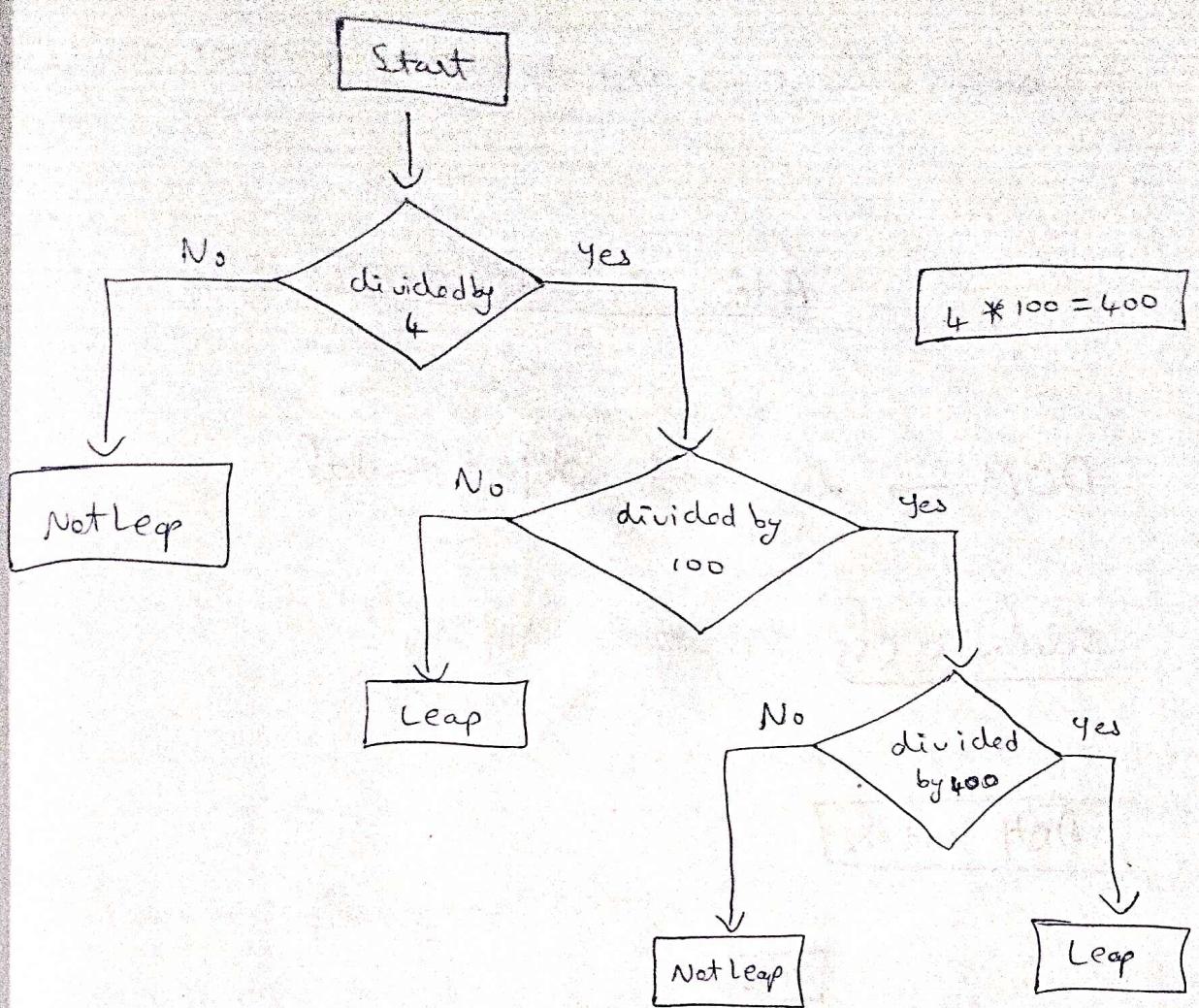
And
OR
Not

Leap Year

- A normal year has 365 days
- A leap year has 366 days (the extra day is 29th Feb)

How to know if it is a leap year.

- 1) divided by 4 $\textcircled{\text{S}}$
- 2) divided by 100 \textcircled{X}
- 3) divided by 400 $\textcircled{\text{S}}$



```
var guestlist = ["Mike", "Jack", "Pam", "James", "Lara", "Jason"]
```

```
var guestName = prompt ("What is your name?");
```

```
guestlist.includes(guestName);
```

Output: Mike
true

Loops → for, while, do while

```
var i=1;
while (i<2) {
    console.log(i);
    i++;
}
```

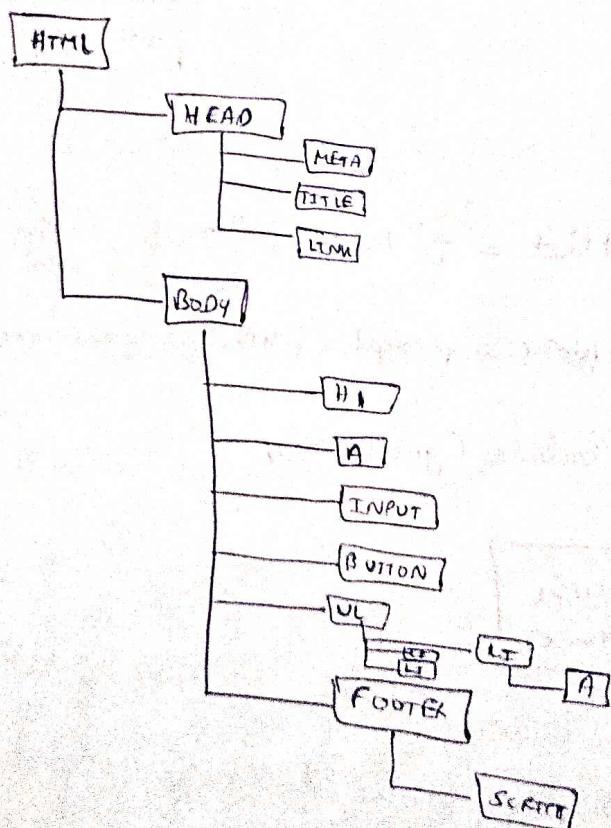
fibonacci challenge → 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144.

Adding JavaScript to Websites

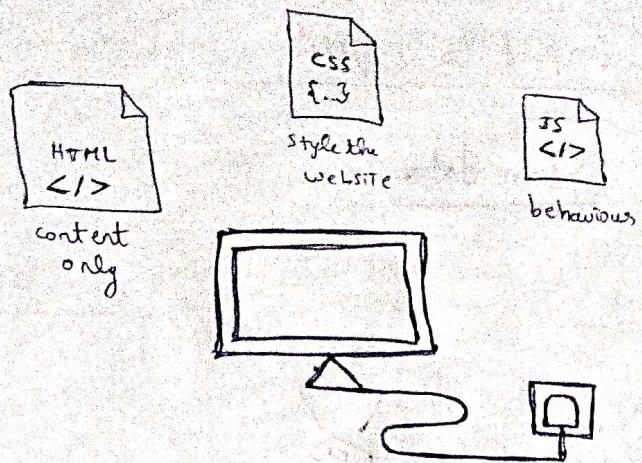
DOM → Document Object Model.

Selectors in CSS

DOM Model



document.querySelector("button").style.backgroundColor = "yellow";
click here



Advanced Javascript
and
DOM

jQuery → JavaScript Library

jQuery → written in JavaScript

Unix Command line

Backend Web development.

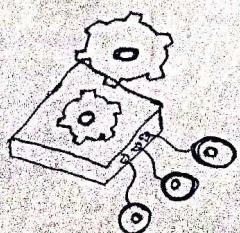
Full Stack = Frontend + Backend

what is a Backend

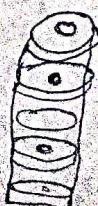
Server



Application

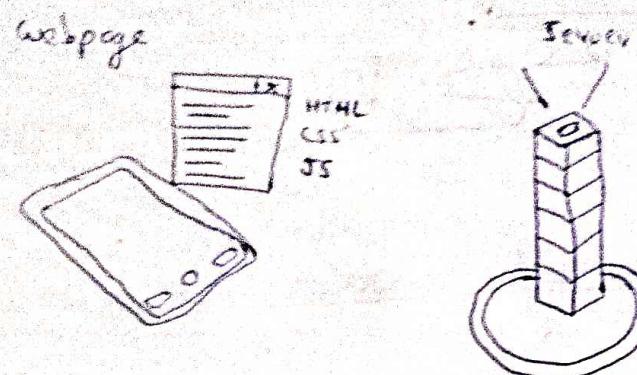


Database

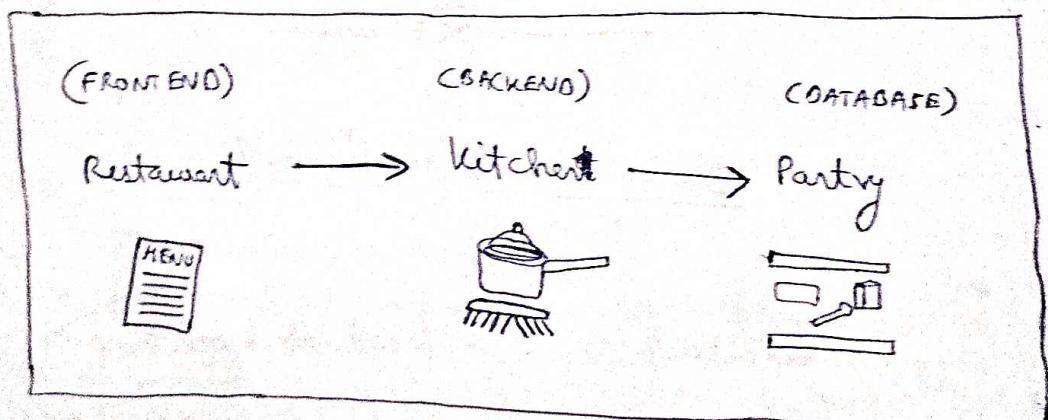
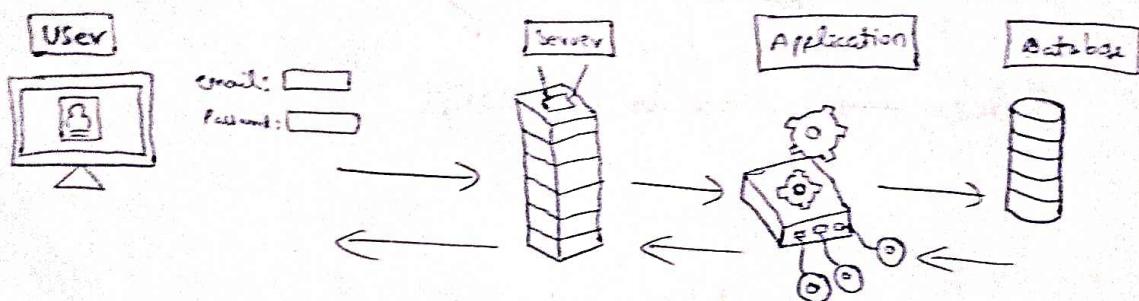


Application → Respond to request

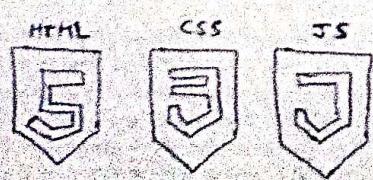
Database → Store data



web App



FRONTEND



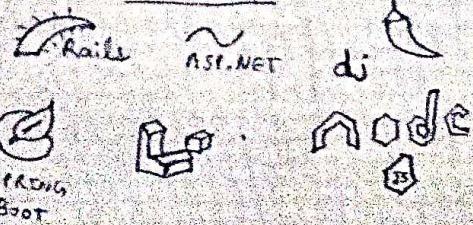
Framework



BACKEND

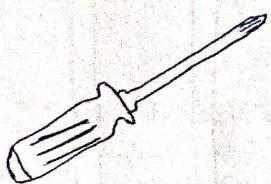


Framework

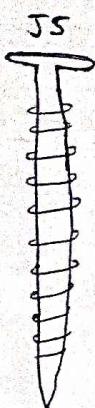


Windows : 98.js.org.

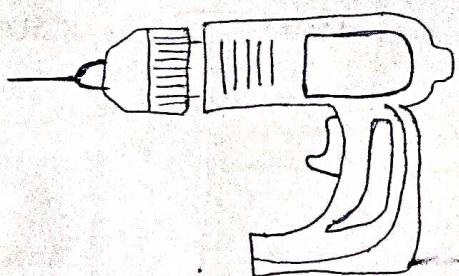
Node.js



Normal Screwdriver



Express.js

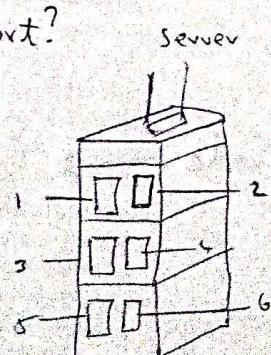


Electric Screw Driver

Creating an Express Server.

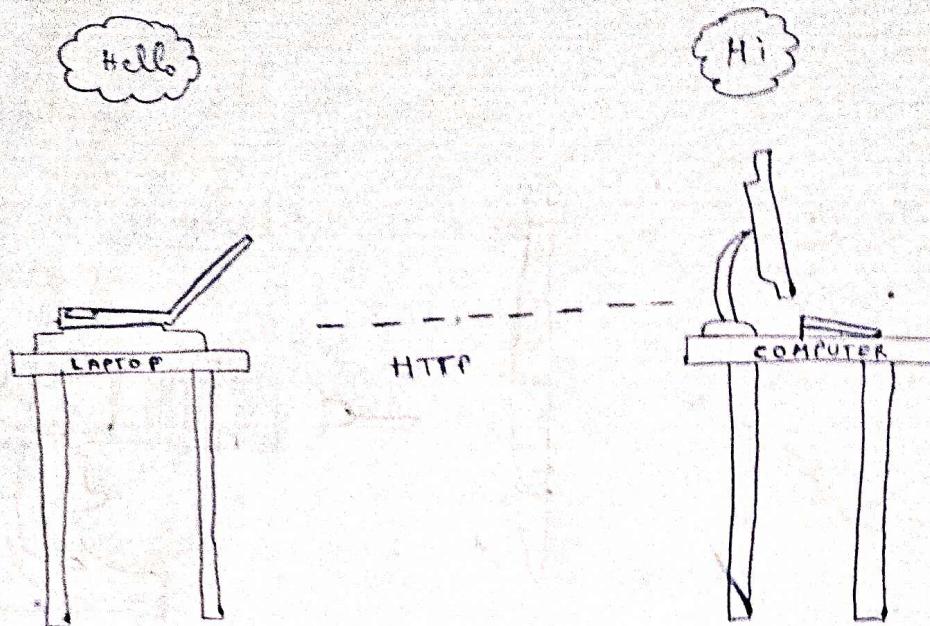
1. Create directory
2. Create index.js file
3. Initialise NPM.
4. Install the Express package.
5. write server application in index.js
6. Start Server

① What is a port?



① Ports are the doors in a Server

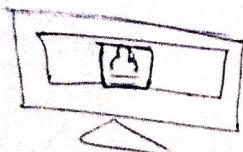
Computers want to communicate with each other



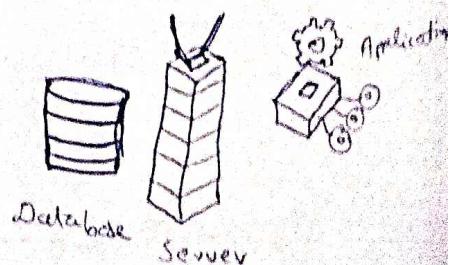
Request verb

- [Get] → Request resource
 - [Post] → Send resource
 - [Put] → Replace resource
 - [Patch] → Patch up a resource
 - [Delete] → Delete Resource.
- (Ex) ④ Newone

Client Side



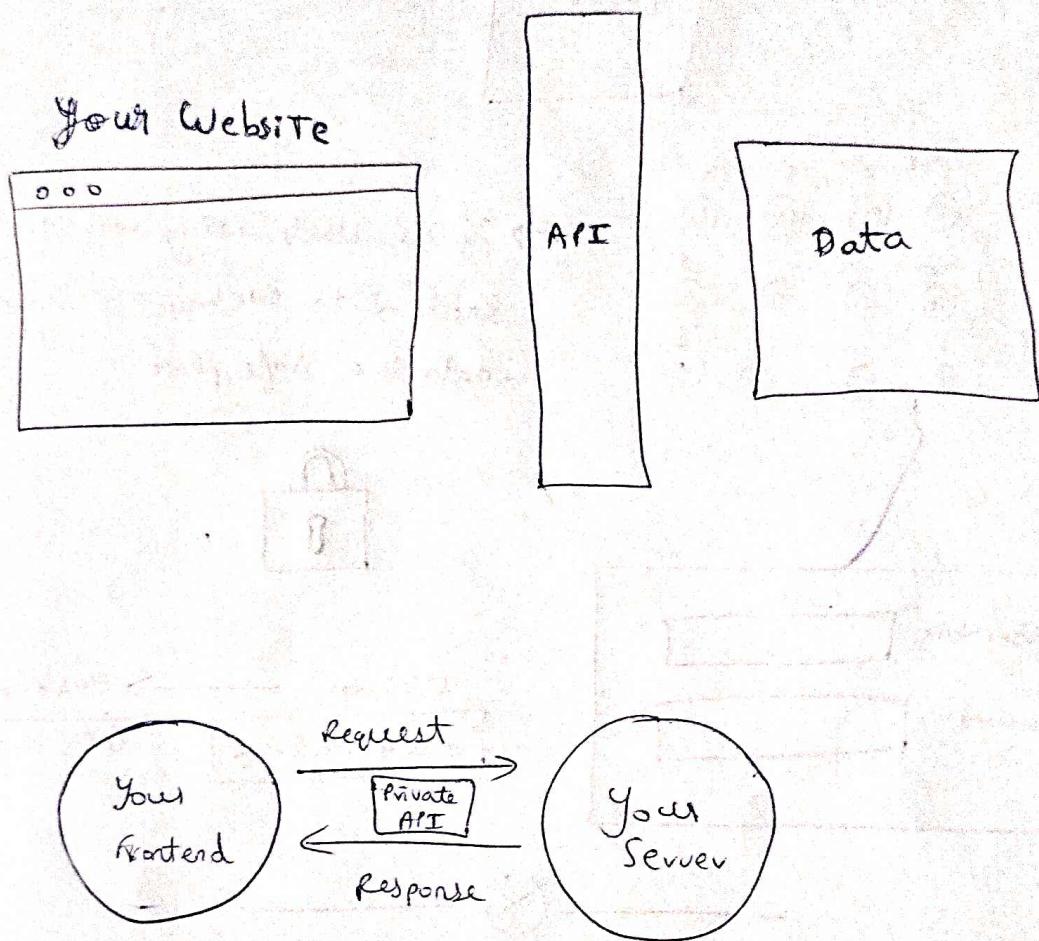
Server-Side



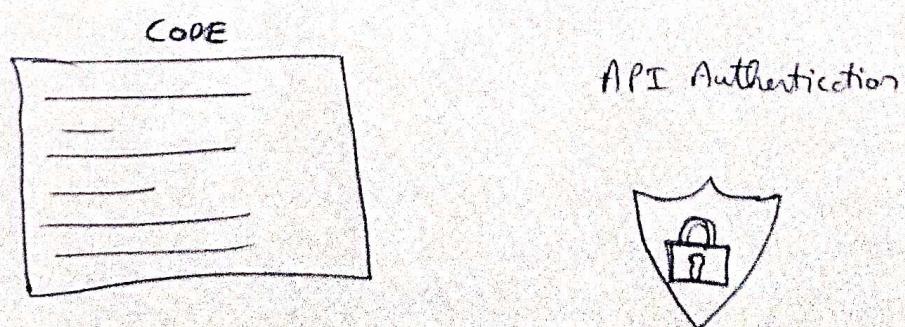
Visual Studio Code → GitHub

Embedded Java Script - API

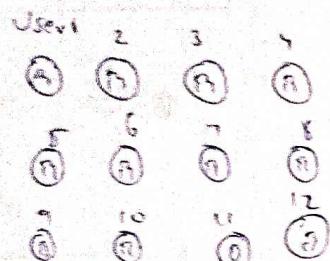
Application Program Interface.



JSON - Server Side API



Authentication



→ If user starts using website They
create Data We have to store the
data in a safe place



User Name	<input type="text"/>
password	<input type="text"/>

User name → Hash Code
 abc@gmail.com d687236ca4

