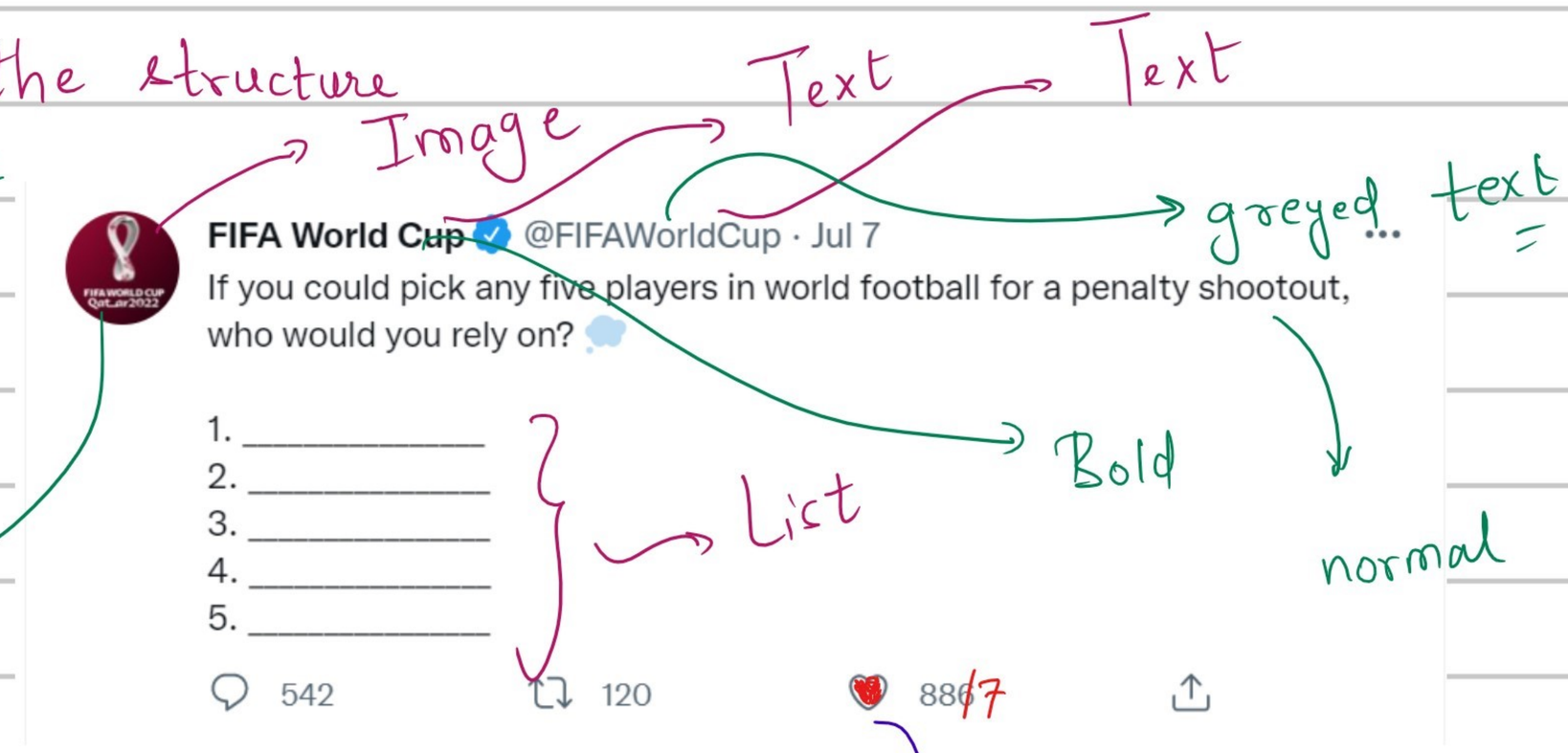
# HTML, CSS & Bootstrap

## Learning Path

HTML → create the structure

CSS → style the structure

JS → actions




HTML → Hyper Text Markup Language → Markup Language

CSS → Cascading Style Sheet → Styling Language

JS → JavaScript → Programming Language

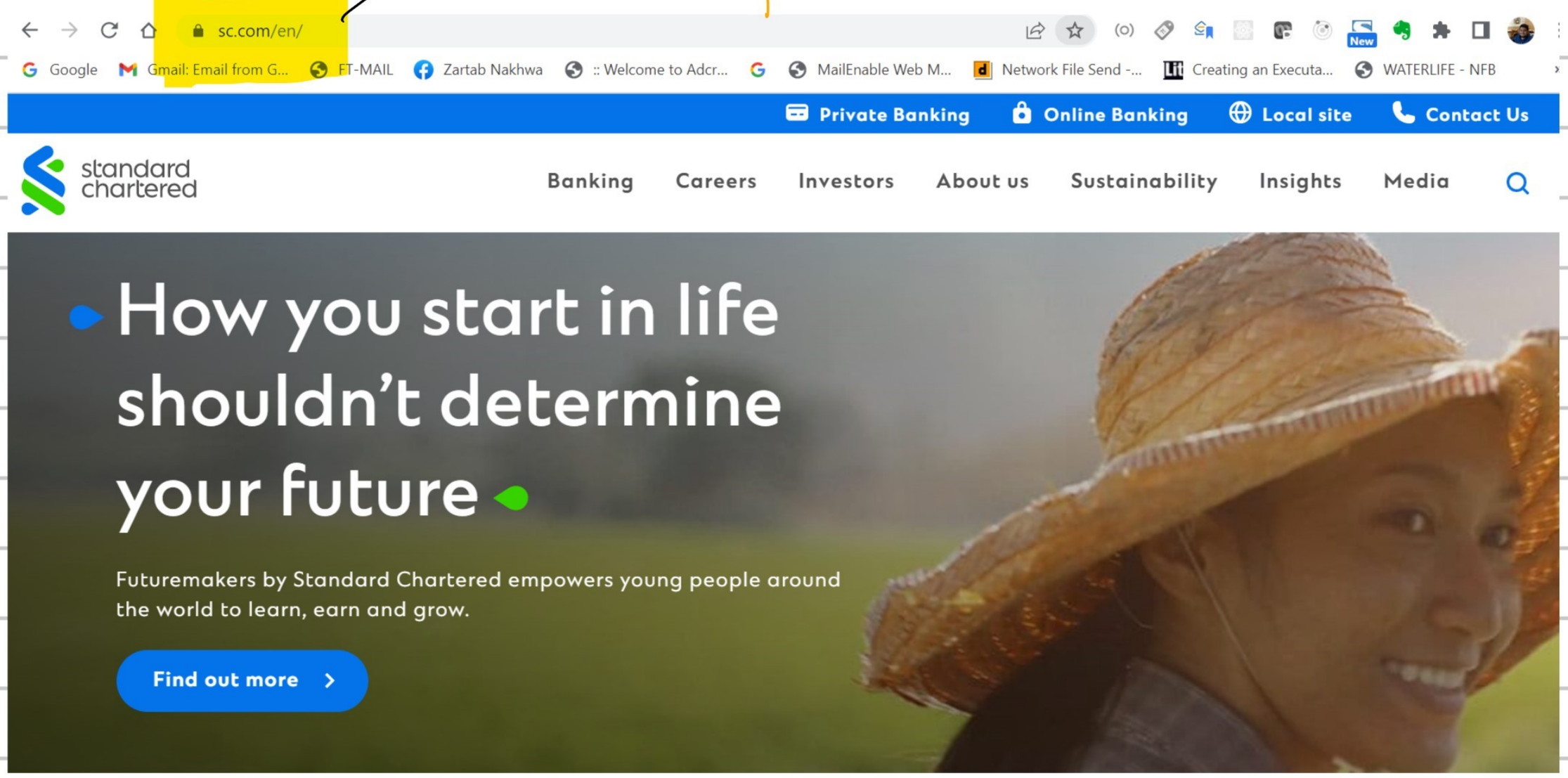
IHU → I Have Understood

RM → Revive Me

# How Web Works?

Makes a request for a resource

URL → Uniform Resource Locator



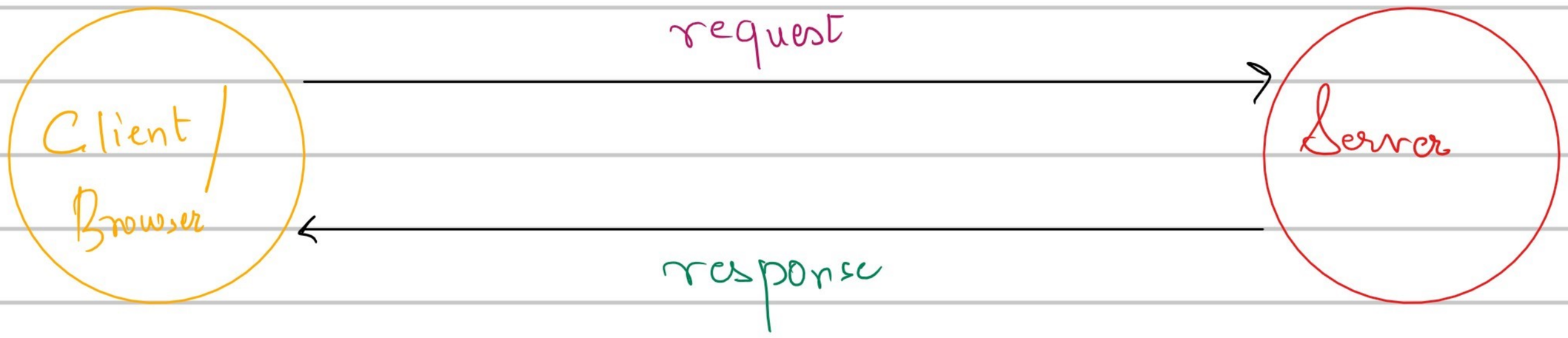

## Structure Of URL

protocol used for communication

https://www.sc.com/resource

Unique address for a website

html
image
video
font
folder
css
js

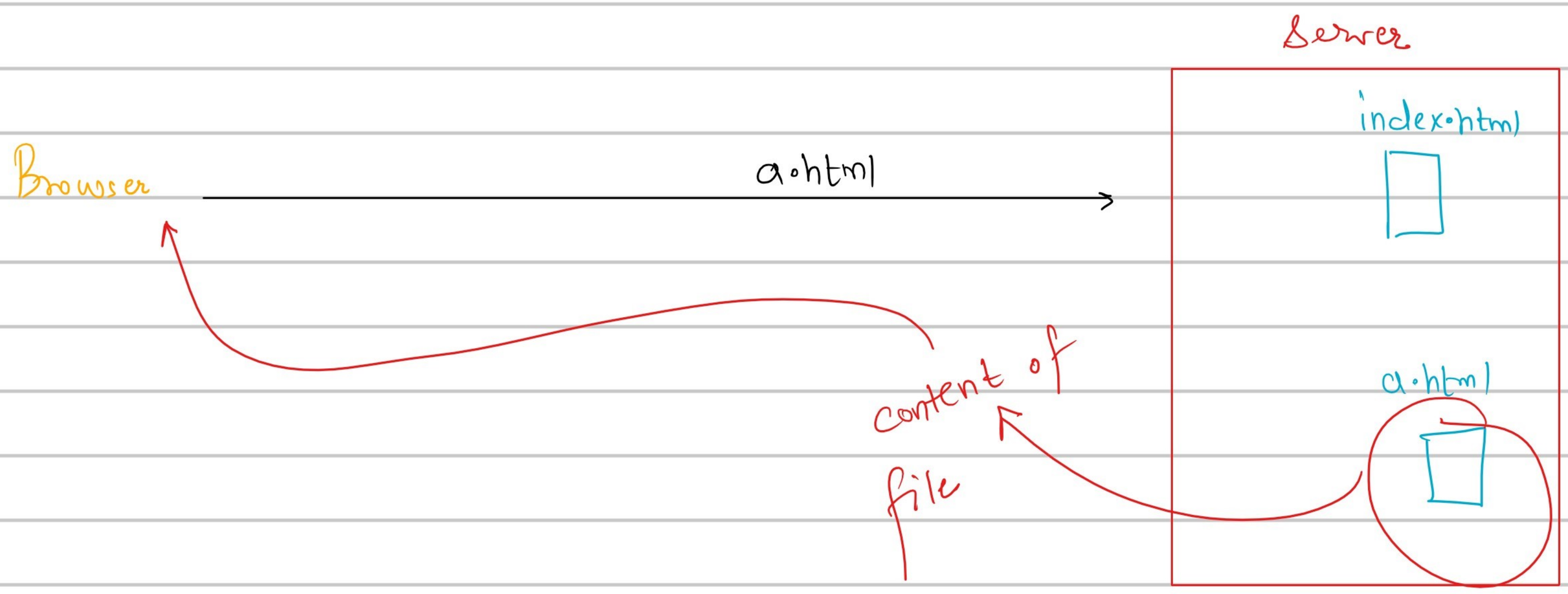


HTTP

Hyper Text Transfer Protocol

HTTPS

HTTP + Encryption




## HTTP REQUEST

GET
POST

GET /index.html HTTP/1.1

resource

under what protocol

HOST: www.xc.com

from where

Accept Language: en-us

language

## HTTP RESPONSE

HTTP/1.1 200 OK

RESPONSE STATUS CODE

Date: 1 Jan 2022 10:00

Timestamp

Content-type: text/html

MIME

<DOCTYPE ---->
<HTML>
</HTML>

HTML Content

▽ → space

# HTML

<tag ▽ attribute = value ▽ attribute = value>

children

</tag>

Self Closing

<tag />

<html>

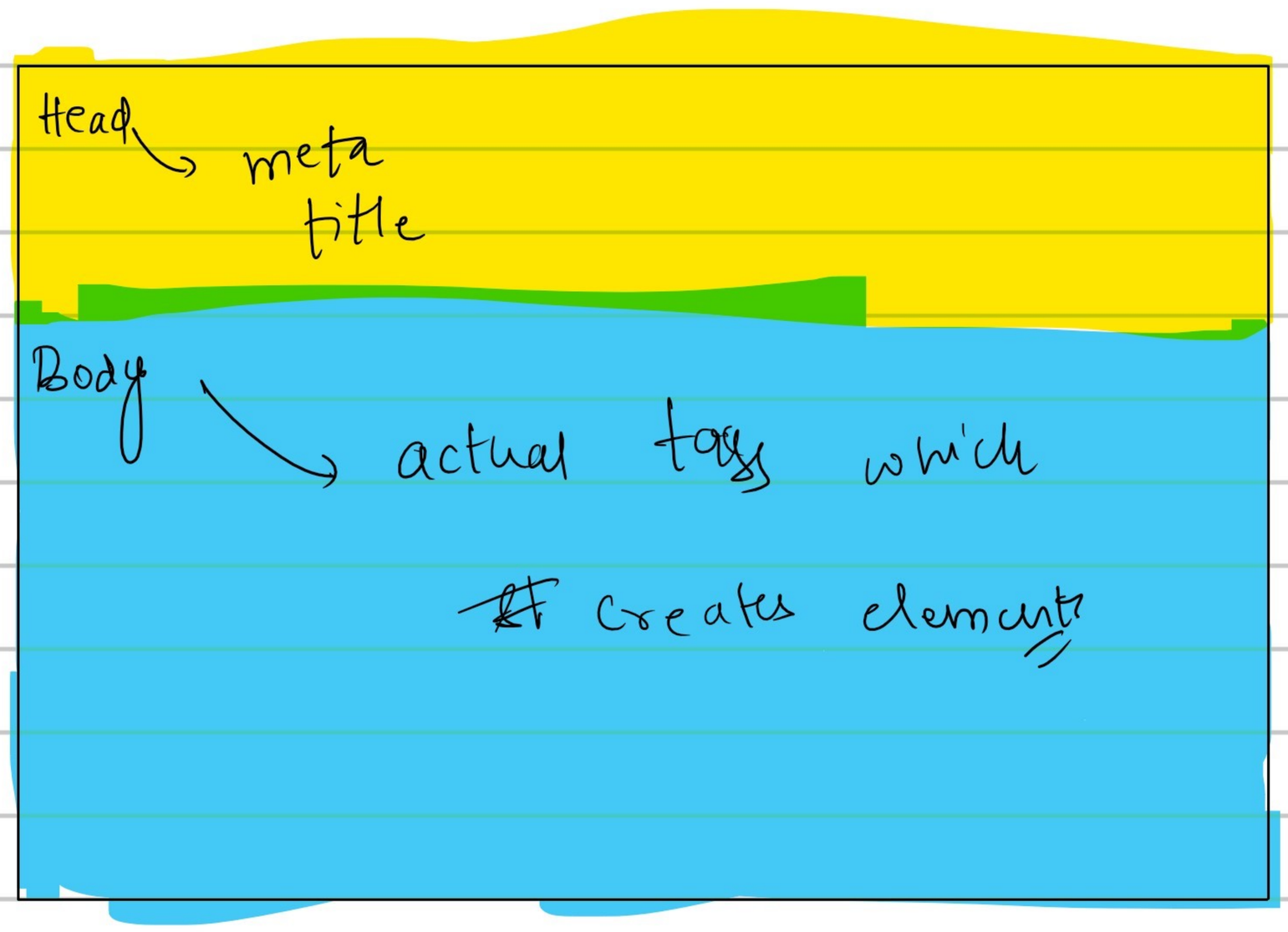
<head>

</head>

<body>

</body>

</html>



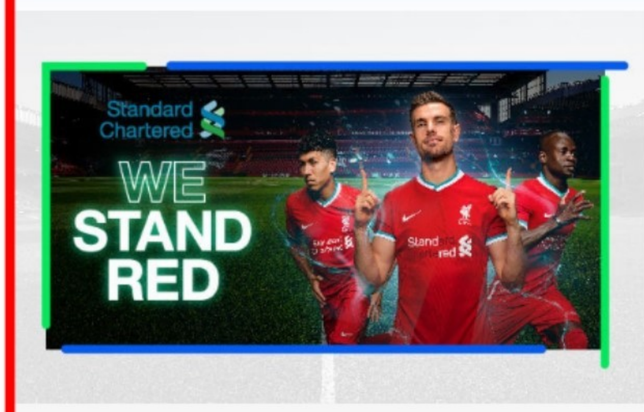

HTML Page

containing

### Be fraud alert

We are aware of a number of scams and attempted frauds, including by companies posing as us, or individuals impersonating our staff

Read more >




### Live Red, Breathe Red, Stand Red

Standard Chartered has been the Main Sponsor of Liverpool FC since 2010. Join us as we create a host of unique experiences for our clients around the world

#StandRed >

### SC Ventures reports first quarter as separate segment

With the recent release of the bank's financial results, SC Ventures reports first quarter out as a separate segment.

Learn more >

div

Block Element

Inline Element

span

block element

span & style seperately

in line element

table

thead

row → <tr>

<th>

| id | name | age | gender |
|---|---|---|---|
| 1 → <td> | Tom | 25 | M |
| 2 | Alex | 26 | M |
| 3 | mike | 27 | M |

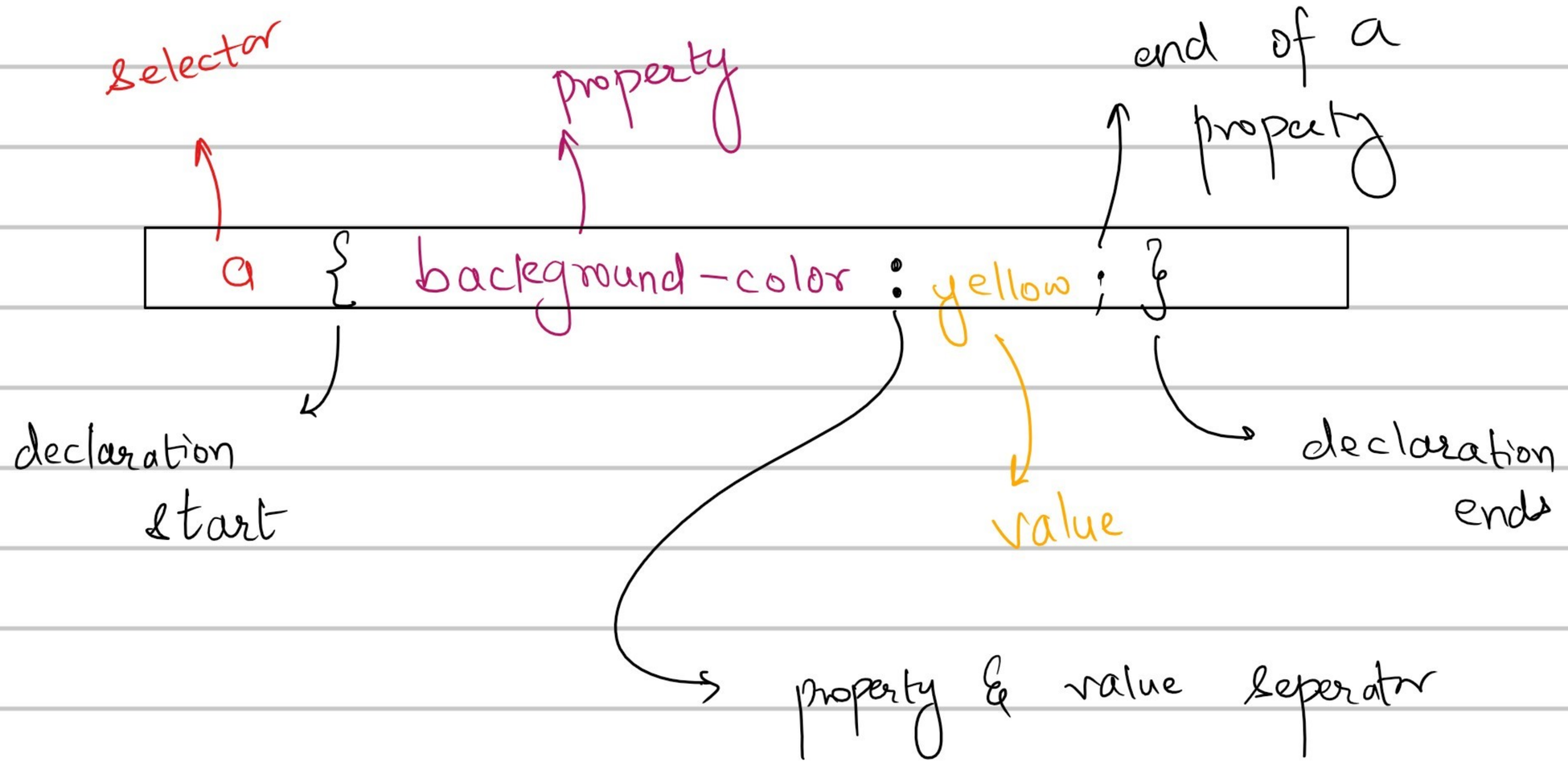
tbody

# CSS

↳ Cascading Style Sheet

→ it is not a programming
→ it is a styling language
→ layouts & designing

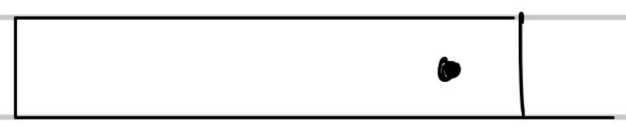
## CSS Selectors




**CSS can be used in 3 ways.**

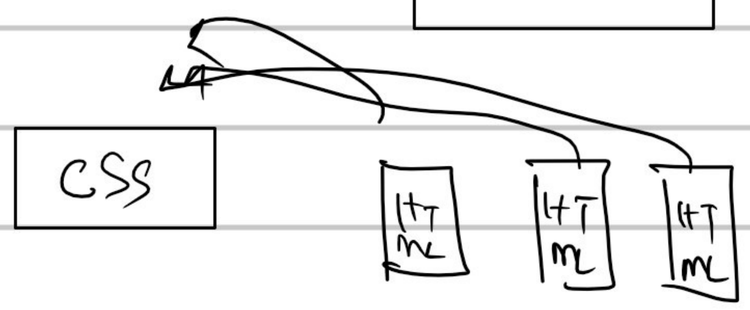
① Inline CSS → directly applied inside the style attribute of HTML element

↳ NO!!



② Internal CSS → use the <style> tag for CSS

③ External CSS → link an html to CSS file



# CSS Selectors

1. Tag
2. Id
3. Class
4. Attribute
5. Wild card Selector.

## Tag

```
<html>
  <head> </head>
    <body>
      :
    </body>
</html>
```

CSS

```
body{
    background-color: red;
}
```

## Id

```
<body>
  <p id="p1" >
  </p>

  <p id="p2" >
  </p>
</body>
```

id has to be unique across the page

CSS

```
#p1{
    background-color: black;
    color: white;
}

#p2{
    color: black;
    font-weight: bold;
}
```

## Class

```
<body>
  <div class="product">  ——  </div>
  <div class="product">  ——  </div>
  <div class="review">  ——  </div>
  <div class="review">  ——  </div>
```

CSS

```
.product{
    background-color: white;
    border: 1 px
}

.review{
    background-color: blue;
    border: 2 px
}
```

# Attribute

```
<a href="——" > — </a>

<a href="——"  target="_blank" > — </a>

<a href="——"  target="_parent" > — </a>
```

CSS

```
a{
  color: purple;
}
a[target]{
  color: yellow;
}
a[target="_blank"]{
  color: green;
}
```

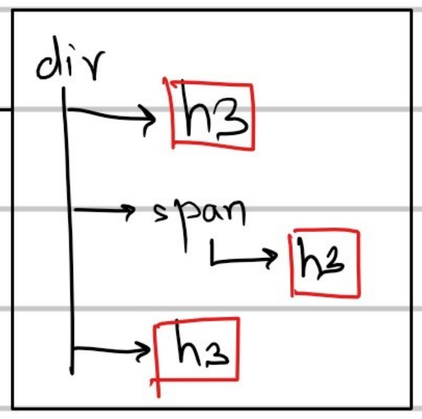
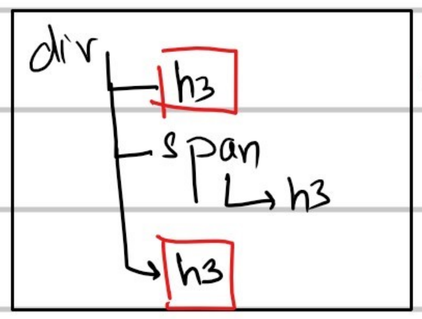
# Wild Card

```
<body>
  <p> ——— </p>

  <hr>
  <p> ——— </p>

</body>
```

CSS

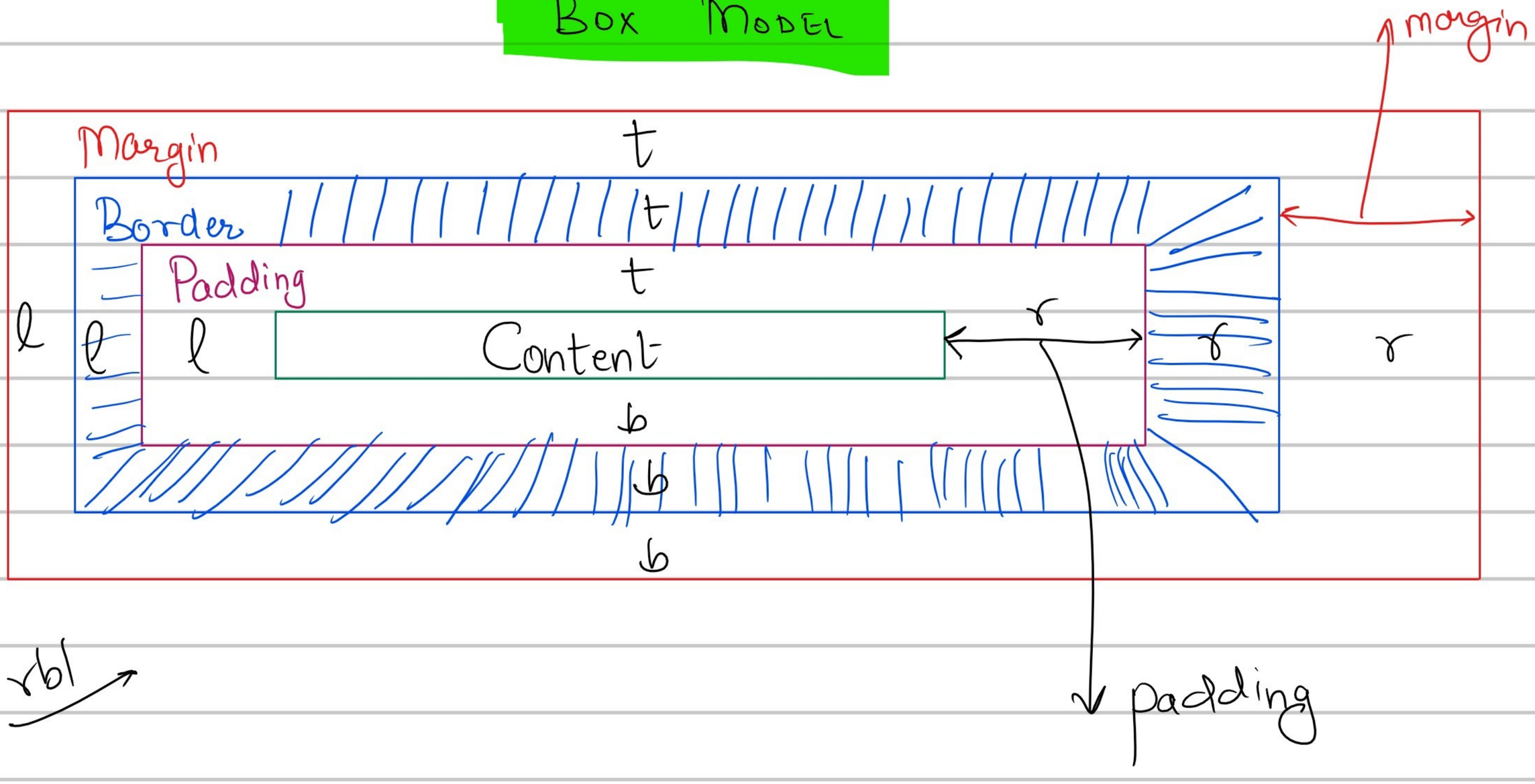
```
*{
  font-family: 'Arial'
}
```

# Combinator

| Selector | Icon | CSS | | |
|---|---|---|---|---|
| Descendant | space | div h3 {<br>color: red;<br>} |  | every occurence of the tag under the main tag |
| Child | > | div > h3 {<br>color: red;<br>} |  | every direct occurrence of the tag |
| Adjacent Sibling | + | div + h3 {<br>color: red;<br>} | <br> <br><h3><br><h3><br><h3> | → only the adjacent tag will be affected |
| General Siblings | ~ | div ~ h3 {<br>color: red;<br>} | <br> <br><h3><br><h3><br><h3> | → all occurence of the tags after main tag |

# Box Model

trbl




trbl

# Display Elements

none → removes the element completely

inline → displays an element like inline (no width & height)

inline-block → display an element with inline-block. (w & h)

block → display an element like a block

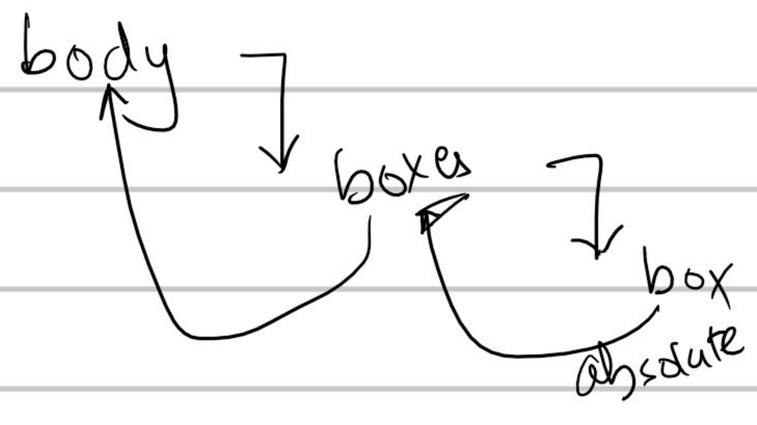
## Positions

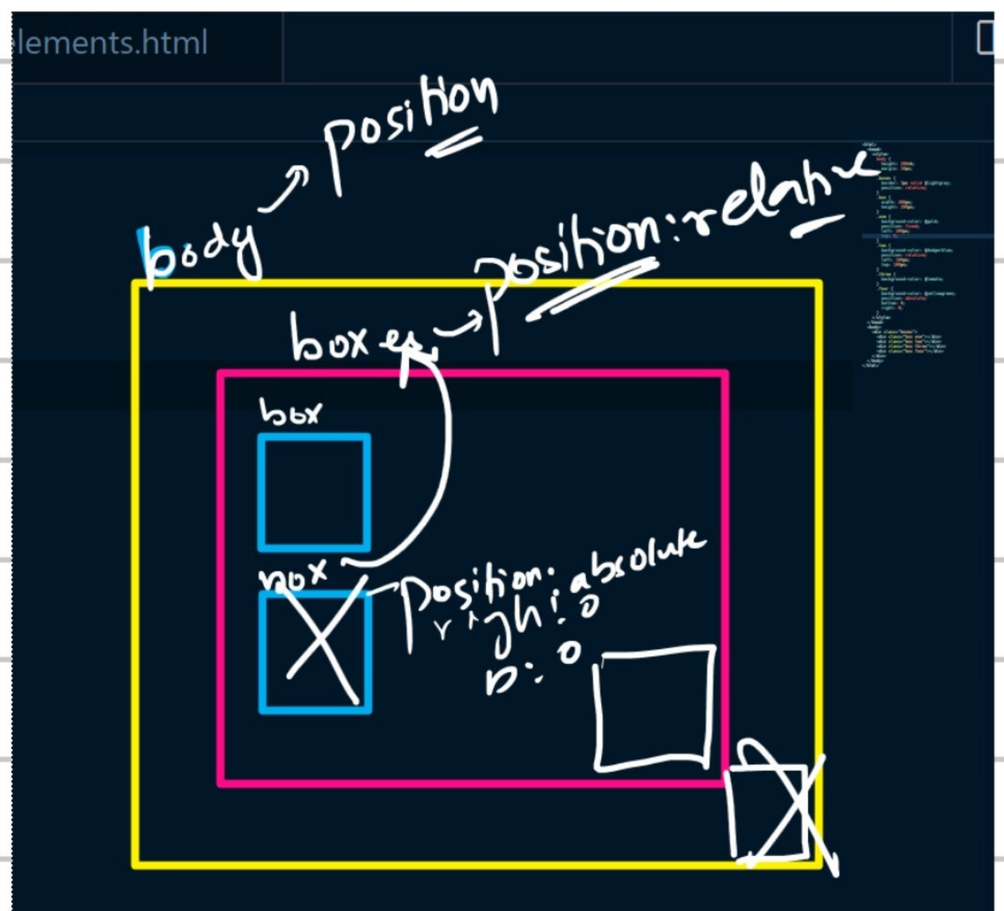
static → default value → all the elements are in the order as they appear

relative → element is positioned relative to it's normal position

absolute → element is positioned absolutely to it's first positioned parent.

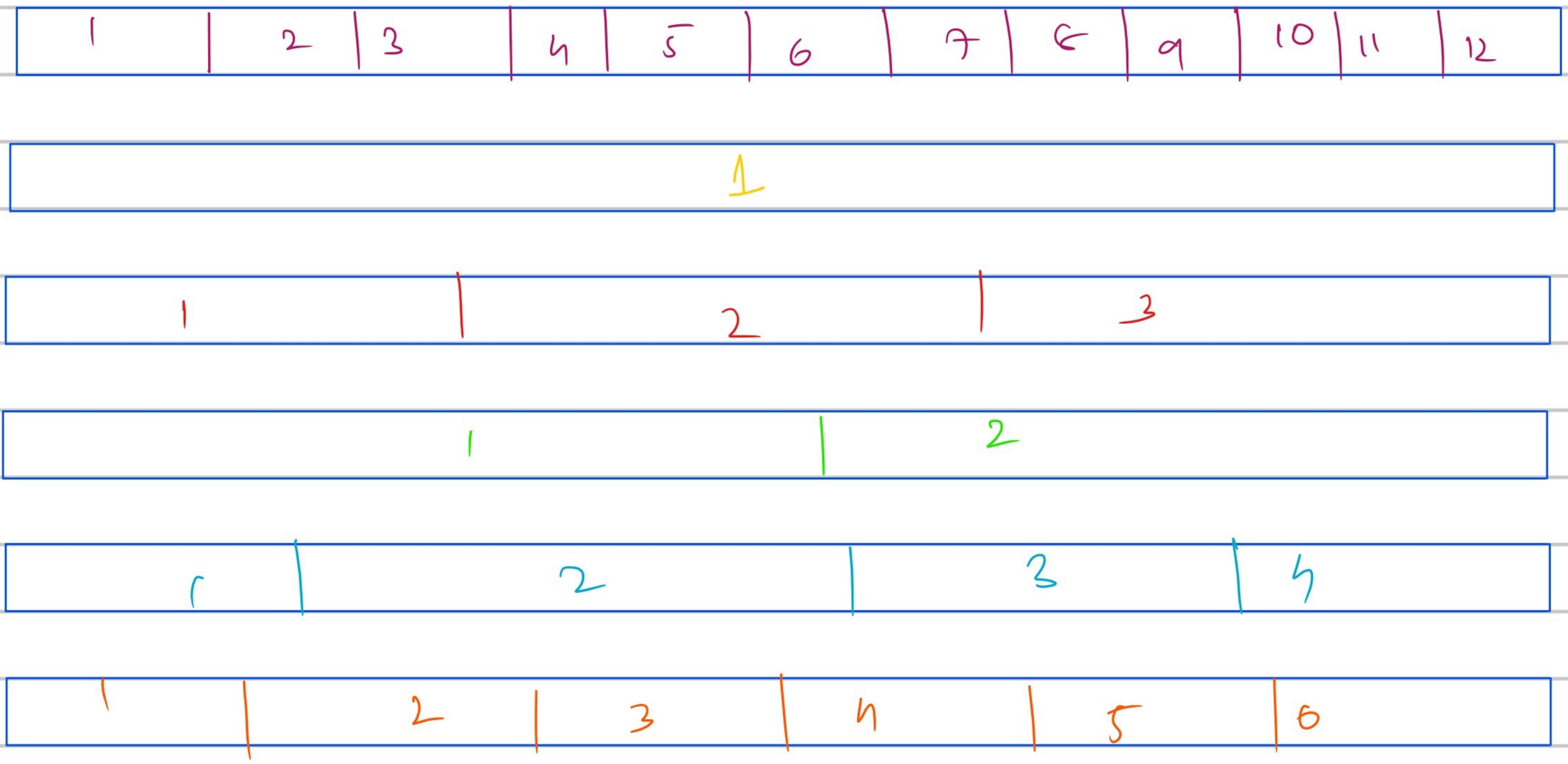
fixed → element is fixed in context to browser window



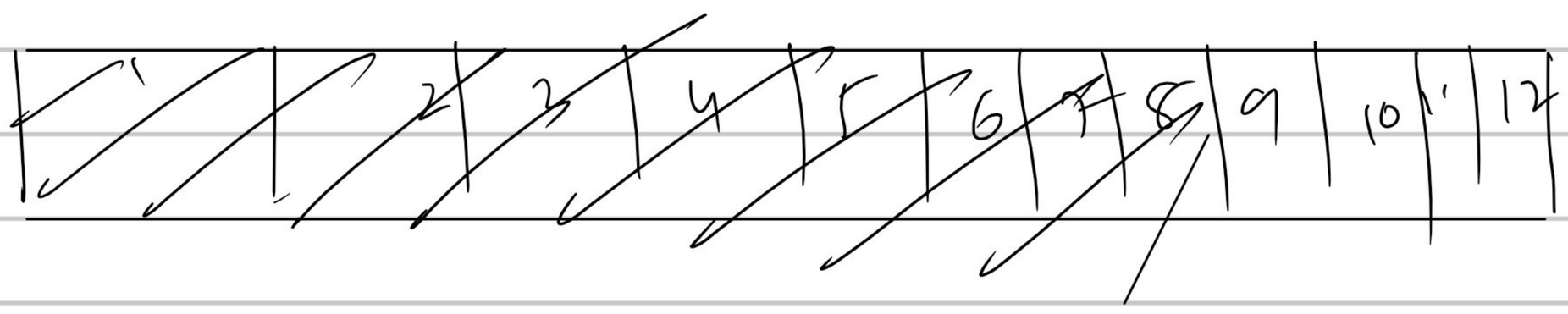
# Bootstrap

## Grid System




## Responsive Breakpoints

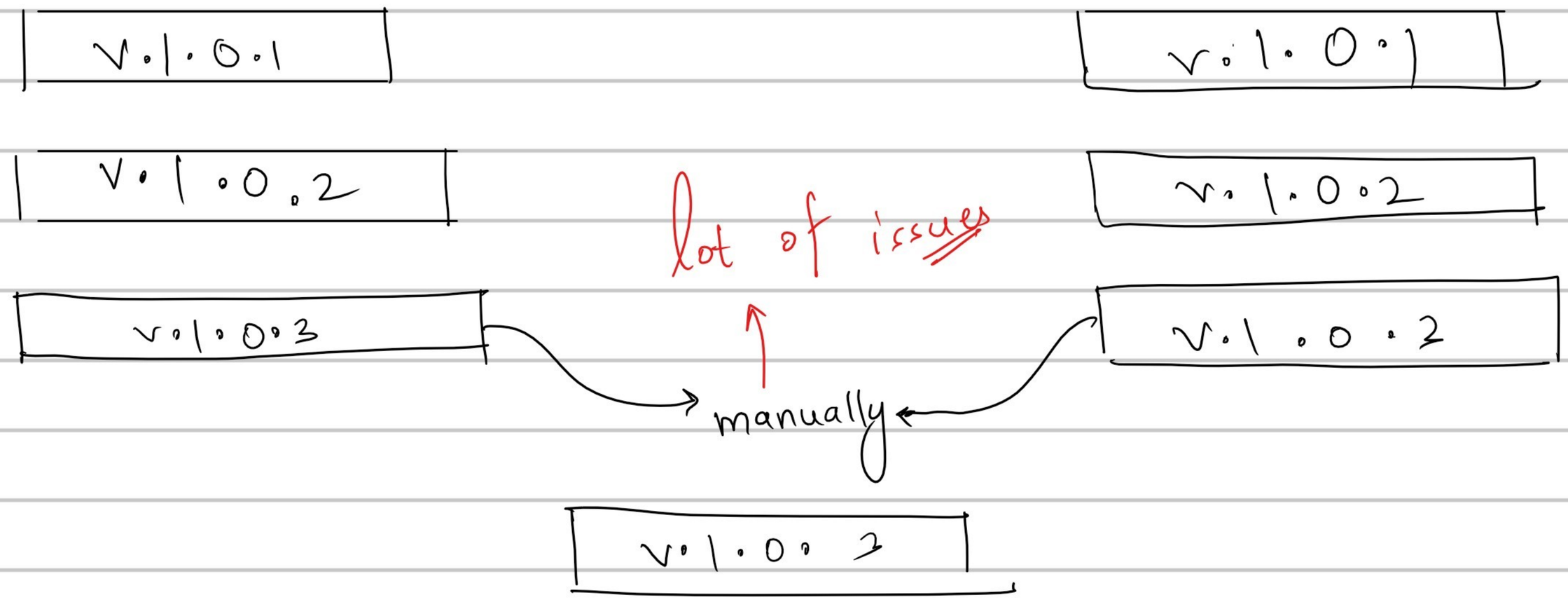
| | Extra small <576px | Small ≥576px | Medium ≥768px | Large ≥992px | Extra large ≥1200px |
|---|---|---|---|---|---|
| **Max container width** | None (auto) | 540px | 720px | 960px | 1140px |
| **Class prefix** | `.col-` | `.col-sm-` | `.col-md-` | `.col-lg-` | `.col-xl-` |
| **# of columns** | 12 | | | | |
| **Gutter width** | 30px (15px on each side of a column) | | | | |
| **Nestable** | Yes | | | | |
| **Column ordering** | Yes | | | | |



# Git → Version Control System.

Tom

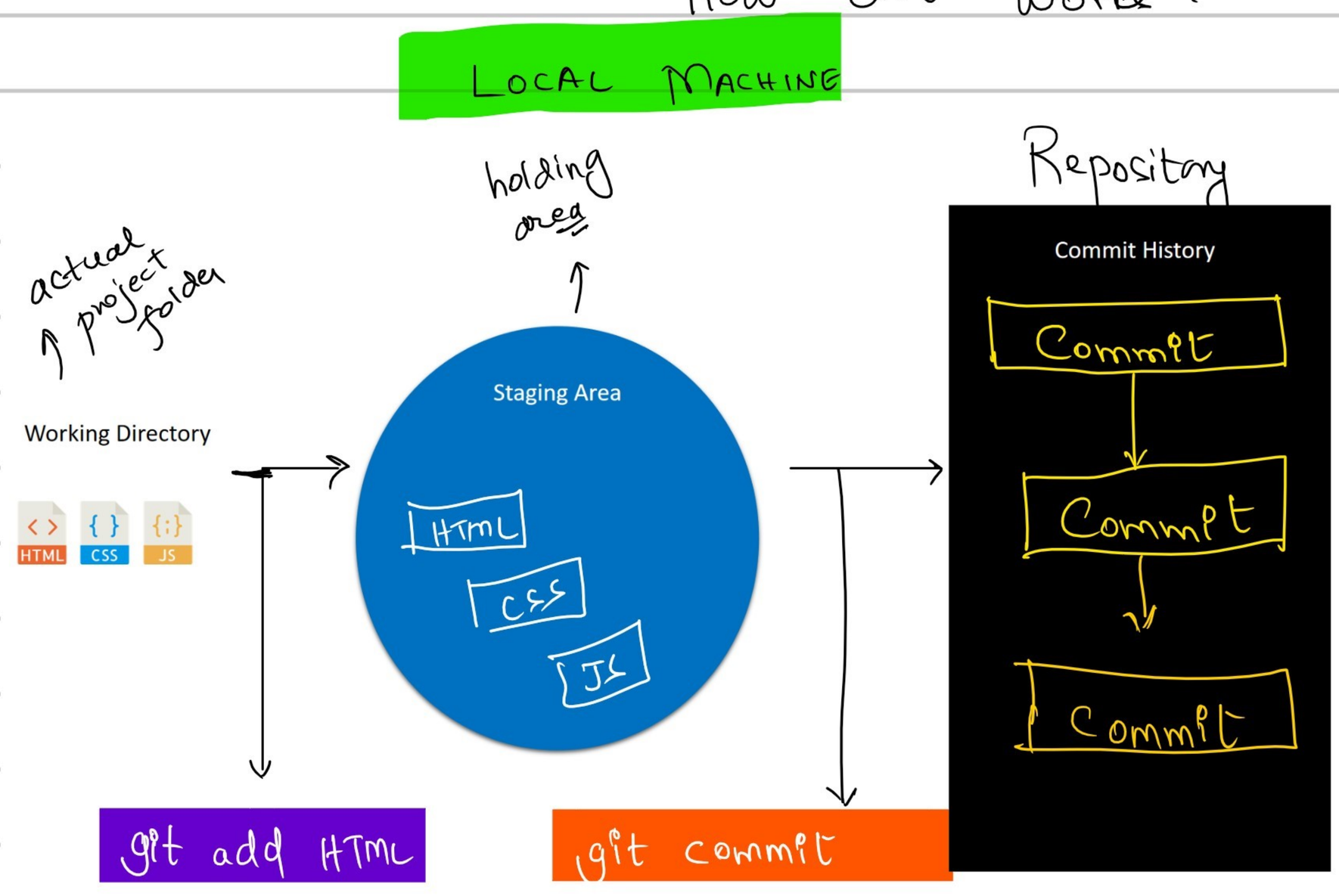
Elizabeth

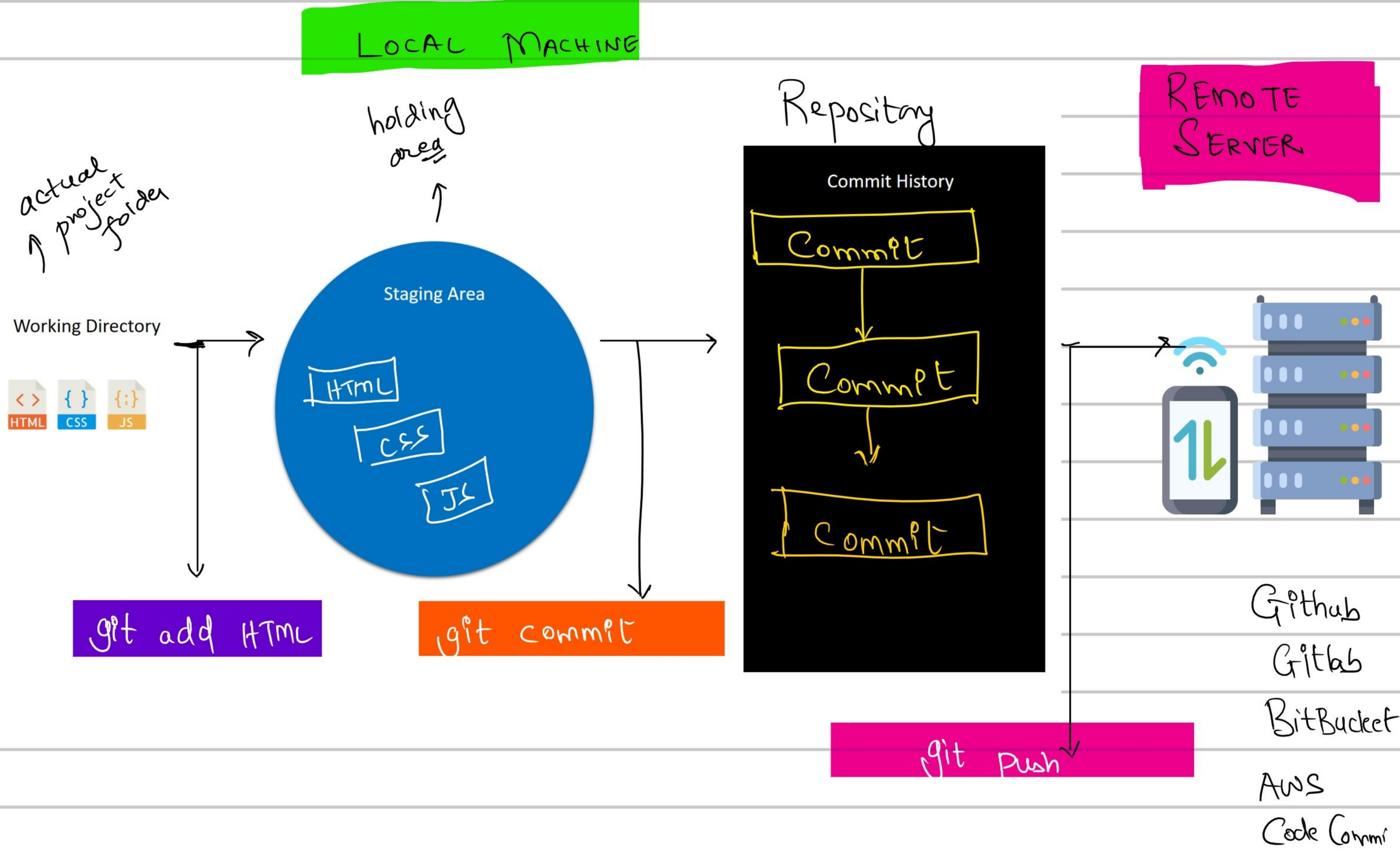



## Version Control System.

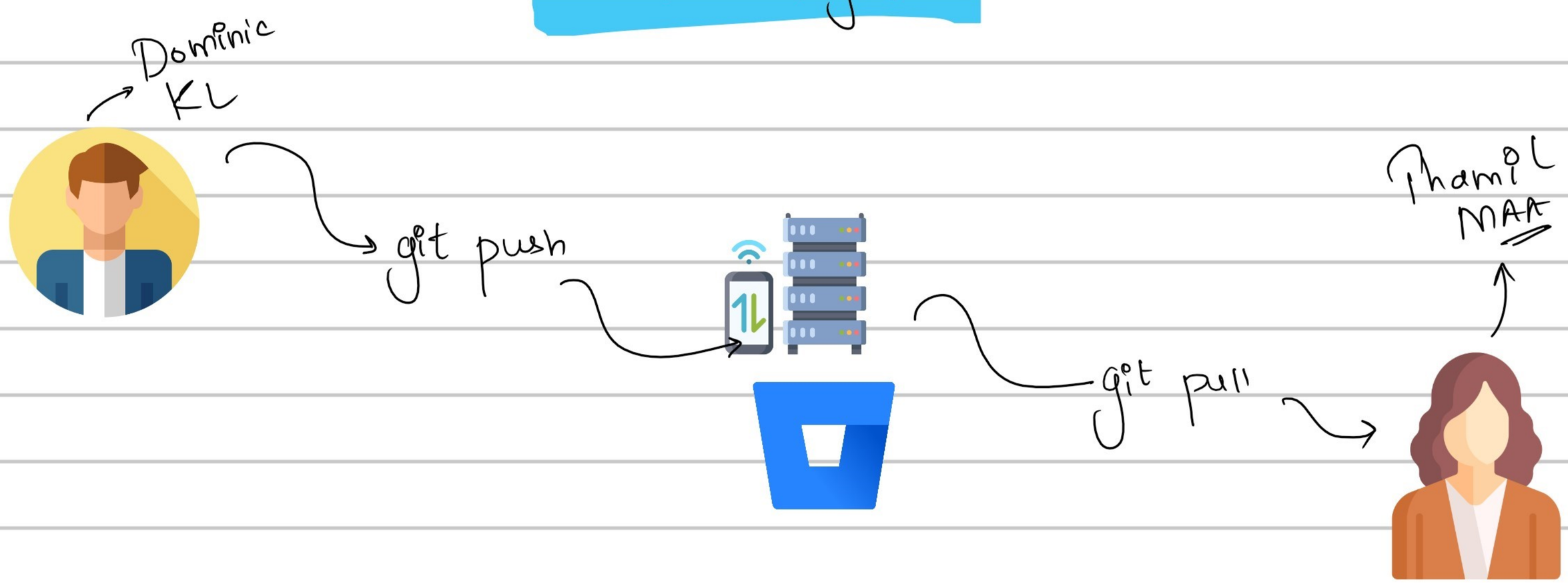
- Source Code Management
- a system which maintains your codebase
- Track History
- Work Together

## How Git works ?






## Work Together




## Why Git ?

* Free & Open Source
* Super Fast
* Scalable
* Cheap Branching & Merging

# Configure Git

1. Name
2. Email
3. Default Editor
4. UI Colors

**SYSTEM** → All users

**GLOBAL** → All the repositories of current user

**LOCAL** → The current repository

## Configuring Git

MINGW64:/c/Users/Zartab Nakhwa

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$ git config --global user.name "Zartab Nakhwa"

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$ git config --global user.email "zartab@codewithz.com"

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$ git config --global core.editor "code"

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$ git config --global color.ui auto

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$
```

.gitconfig × | Release Notes: 1.69.1

C: > Users > Zartab Nakhwa > .gitconfig

```
[user]
    name = Zartab Nakhwa
    email = zartab@codewithz.com
[core]
    editor = code
```

## Checking all config

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$ git config -l
credential.helper=manager-core
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Zartab Nakhwa
user.email=zartab@codewithz.com
core.editor=code
color.ui=auto
```

## Switch to Project directry

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$ d:
bash: d:: command not found

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 ~
$ cd d:

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d
$ cd SCB/2022/Bootcamp\ July\ 2022/Git/myproject/

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproje
ct
$
```

## ① Initiallze the Repositry

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproje
ct
$ git init .
Initialized empty Git repository in D:/SCB/2022/Bootcamp July 2022/Git/myproject
/.git/

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproje
ct (master)
$ ls -a
./  ../  .git/  index.html  styles/

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproje
ct (master)
$
```

## ② Add some/all files to Staging Area

```
nothing added to commit but untracked files present (use "git add" to track)

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add index.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        styles/
```

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   styles/style.css

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
```

## ③ Commit the files to repositry

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git commit -m "Initial Commit"
[master (root-commit) b4b31aa] Initial Commit
 2 files changed, 21 insertions(+)
 create mode 100644 index.html
 create mode 100644 styles/style.css
```

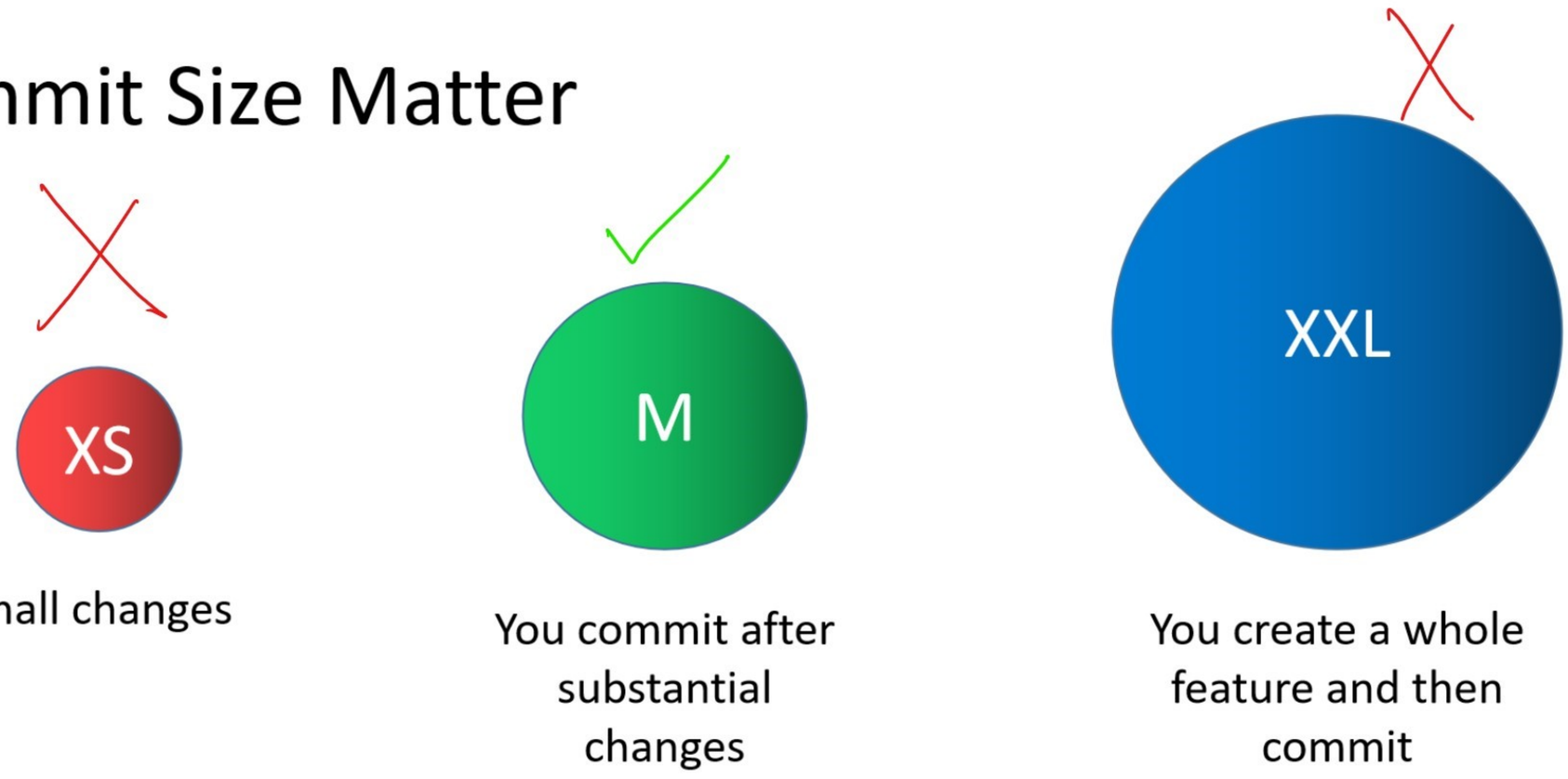
## To check commit history

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log
commit b4b31aa25d2bfd7b2aeab4b1bab52cadf221259a (HEAD -> master)
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 11:45:13 2022 +0530

    Initial Commit
```

Give meaningful commit messages

# Commit Best Practices

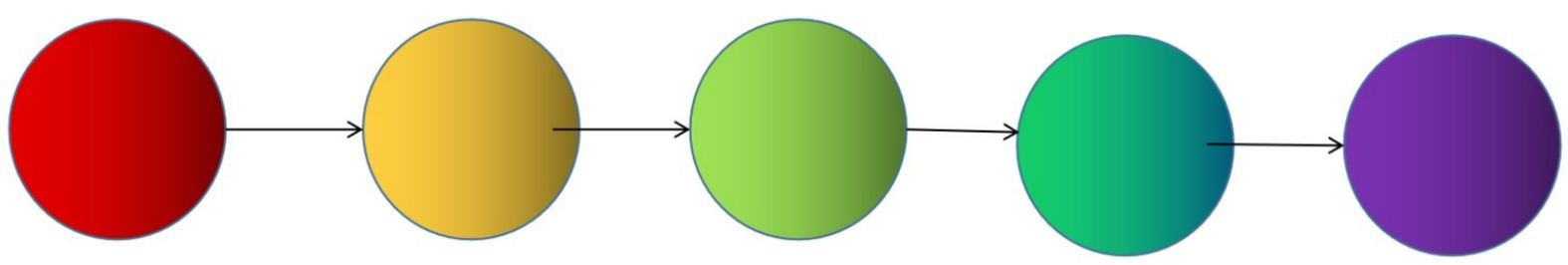
## Commit Size Matter




# Commit Best Practices

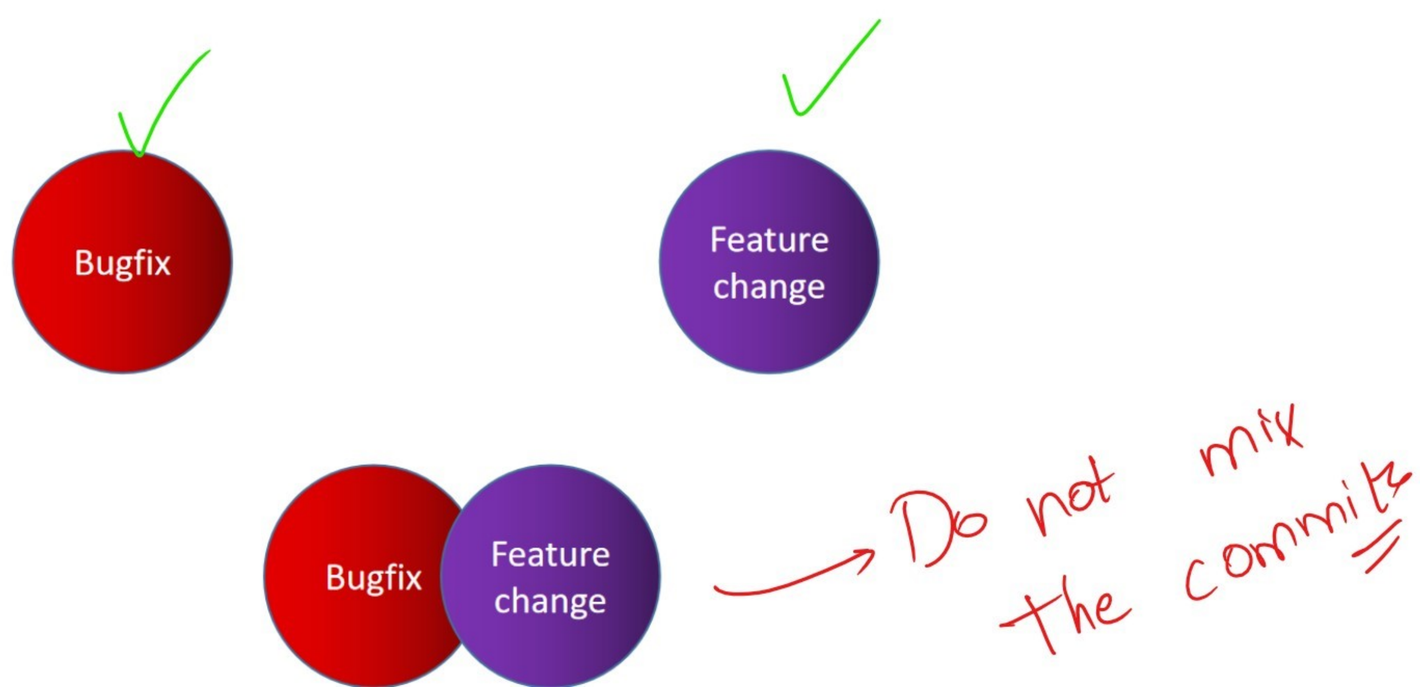
## Commit Often -

As you reach a state you want to RECORD then make a commit



# Commit Best Practices

## Don't mix different types of commit:



## Made changes in html & git status

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

## Add modified file to SA & commit

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git commit -m "Updated the title in index.html"
[master 7d08c1b] Updated the title in index.html
 1 file changed, 1 insertion(+), 1 deletion(-)

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log
commit 7d08c1b309f6a3002a1363c1cb4818b284f35437 (HEAD -> master)
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 11:58:05 2022 +0530

    Updated the title in index.html

commit b4b31aa25d2bfd7b2aeab4b1bab52cadf221259a
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 11:45:13 2022 +0530

    Initial Commit

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
```

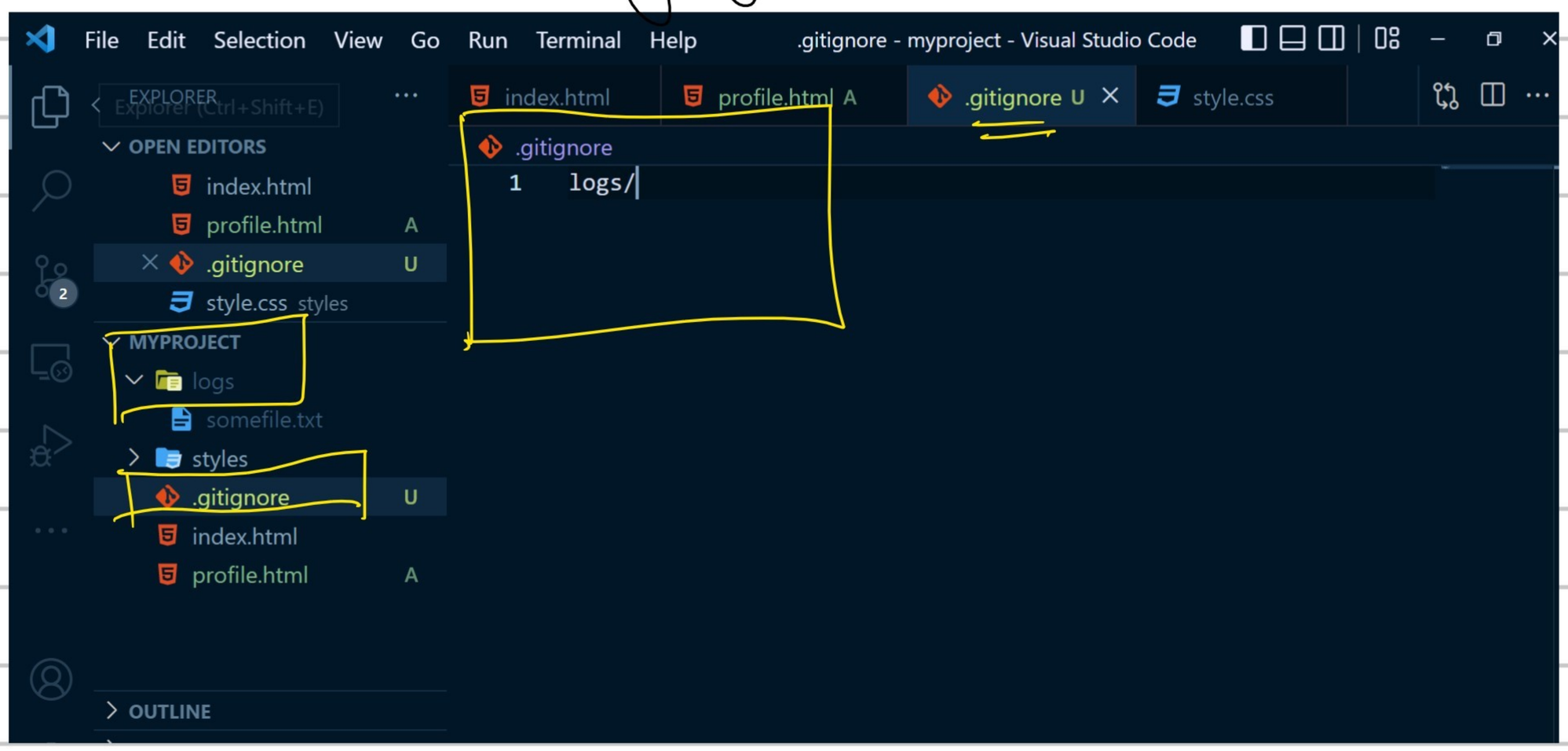
## Checking for files in Staging Area

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        profile.html

nothing added to commit but untracked files present (use "git add" to track)

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git ls-files
index.html
profile.html
styles/style.css

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

# Ignoring the files

In working directory → log
→ compiled source code

→ .gitignore




```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   profile.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore


Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add .gitignore

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git commit -m "Added .gitignore"
[master cd81f08] Added .gitignore
 2 files changed, 6 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 profile.html
```

## Status Short Hand

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status -s
 M index.html
?? product.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status -s
M  index.html
A  product.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

(M) - Modified
(??) - Untracked
Red → unstaged

(M)
(A) - Added
Green → staged

# Viewing Difference in Working Directory & Staging Area

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git diff
diff --git a/index.html b/index.html
index 0980b44..065abdd 100644
--- a/index.html
+++ b/index.html
@@ -14,6 +14,7 @@
        </p>
        <h2>Description</h2>
        <p>Lorem ipsum dolor sit amet.</p>
+       <p>Lorem ipsum dolor sit amet.</p>
      </div>
    </body>
  </html>

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

# Viewing difference in Staging Area & Repository

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git diff --staged
diff --git a/index.html b/index.html
index 94441dc..0980b44 100644
--- a/index.html
+++ b/index.html
@@ -13,6 +13,7 @@
          veritatis nulla quisquam maxime.
        </p>
        <h2>Description</h2>
+       <p>Lorem ipsum dolor sit amet.</p>
      </div>
    </body>
  </html>
```

# Checking the history

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log
commit bc58150278de5f5b5ac05f0457d0b2104fe5e442 (HEAD -> master)
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 12:15:42 2022 +0530

    Updated product and index

commit cd81f08f4acb68108ec208c625903bbbd9d6ef5e
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 12:09:26 2022 +0530

    Added .gitignore

commit 7d08c1b309f6a3002a1363c1cb4818b284f35437
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 11:58:05 2022 +0530

    Updated the title in index.html

commit b4b31aa25d2bfd7b2aeab4b1bab52cadf221259a
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 11:45:13 2022 +0530

    Initial Commit
```

Commit id

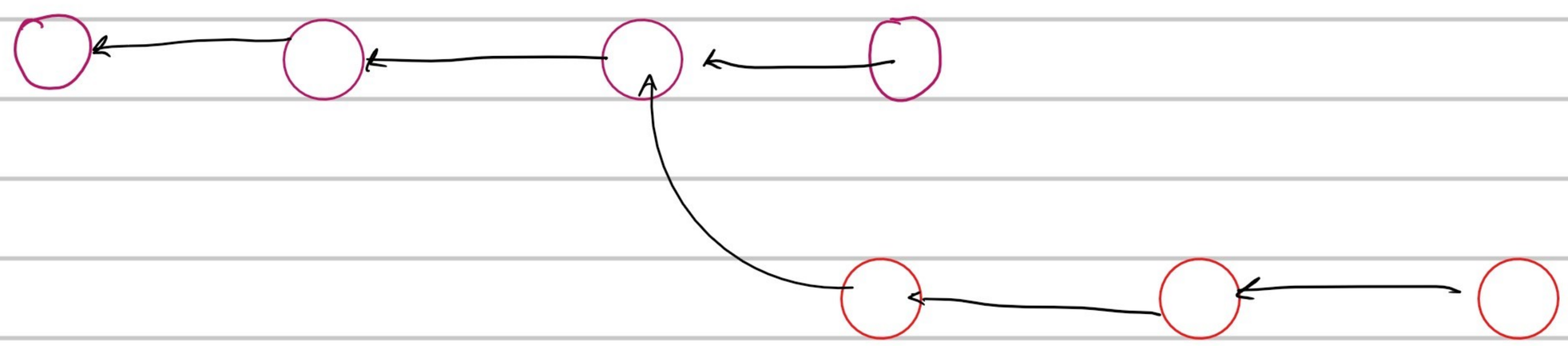
Commit message

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --oneline
bc58150 (HEAD -> master)  Updated product and index
cd81f08 Added .gitignore
7d08c1b Updated the title in index.html
b4b31aa Initial Commit
```

## Unstage a file

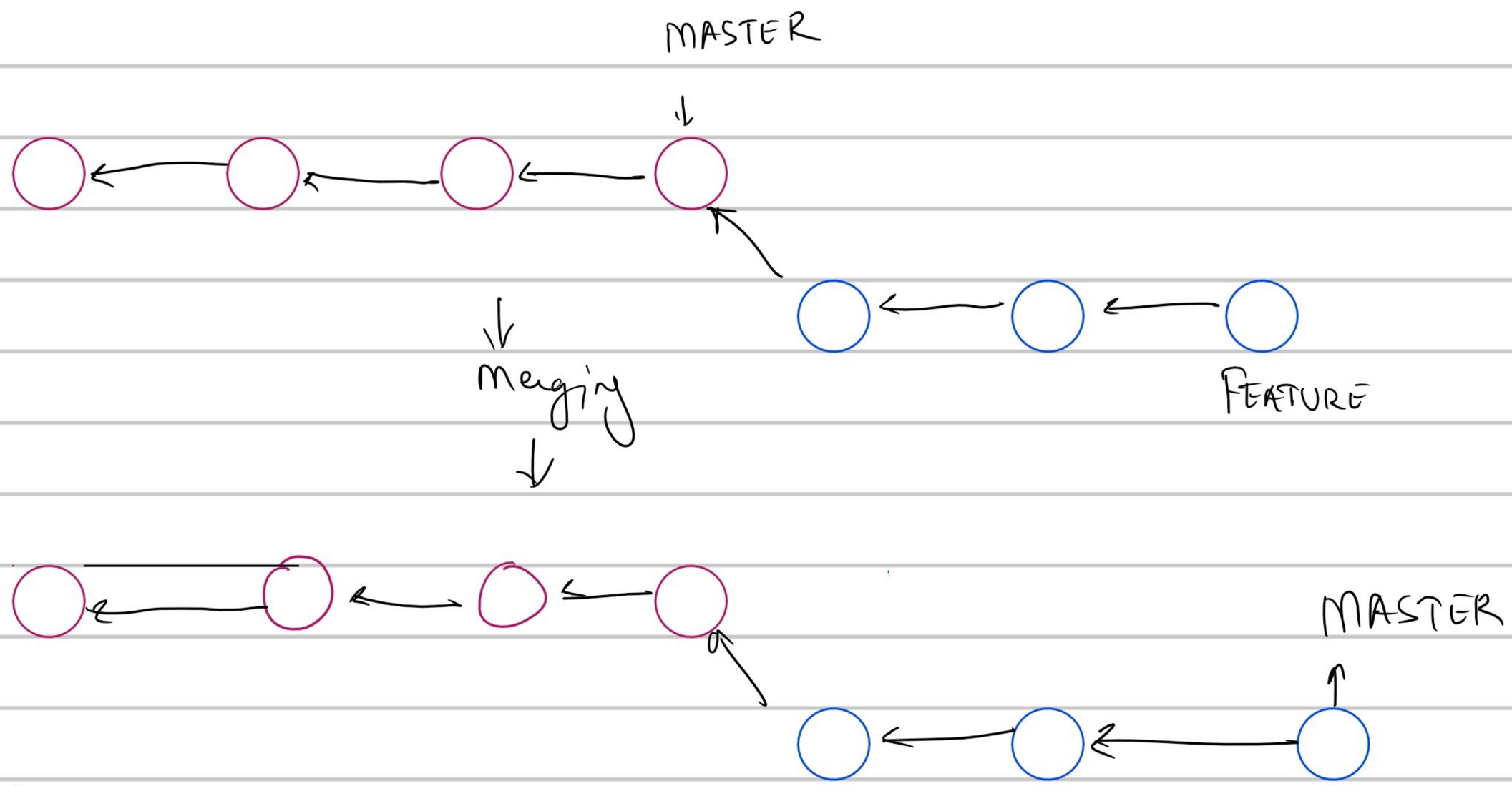
```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html


Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git restore --staged index.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

## Branching



allows us to diverge from the main line of work & work on something else

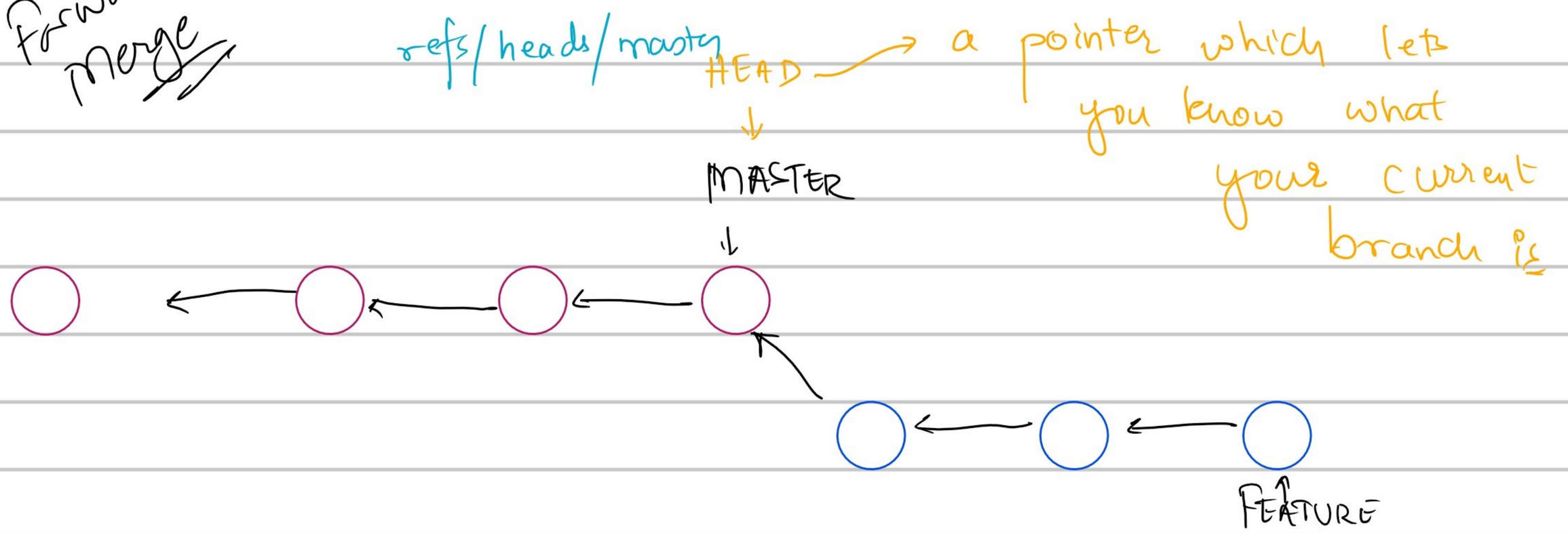
## Branching




fast forward merge

refs/heads/master

HEAD → a pointer which lets you know what your current branch is




refs/heads/feature

## Creating a Branch

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject

You're sharing this screen

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git branch feature-signup

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git branch
  feature-signup
* master

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

(list the branches)

# Switching in the branches

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git checkout feature-signup
Switched to branch 'feature-signup'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-si
gnup)
$
```

git switch feature-signup

`git checkout` is a bit of a swiss army knife in that has several unrelated uses.

If you modify a file but haven't staged the change, then `git checkout <filename>` will reverse the modifications... a quick and easy way to cancel changes to a file. You remain in the same branch.

`git checkout <branchname>` (as you noted) switches branches.

Two completely different purposes, which could lead to confusion if a file name and a branch name are similar.

Having it as two commands is clearer.

Switched to new branch; added a file; commited; checked log; switched back to master; checked log

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git switch feature-signup
Switched to branch 'feature-signup'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-si
gnup)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-si
gnup)
$ git commit -m "Added the signup form"
[feature-signup 0b094eb] Added the signup form
 1 file changed, 7 insertions(+)
 create mode 100644 signup.html
```

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-si
gnup)
$ git switch master
Switched to branch 'master'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git status
On branch master
nothing to commit, working tree clean

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --online
fatal: unrecognized argument: --online

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --oneline
6c34732 (HEAD -> master) Version 1
bc58150  Updated product and index
cd81f08 Added .gitignore
7d08c1b Updated the title in index.html
b4b31aa Initial Commit
```

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git switch feature-signup
Switched to branch 'feature-signup'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-si
gnup)
$ git log --oneline
0b094eb (HEAD -> feature-signup) Added the signup form
6c34732 (master) Version 1
bc58150  Updated product and index
cd81f08 Added .gitignore
7d08c1b Updated the title in index.html
b4b31aa Initial Commit
```

## To check history of all the branches

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --all --oneline
0b094eb (feature-signup) Added the signup form
6c34732 (HEAD -> master) Version 1
bc58150  Updated product and index
cd81f08 Added .gitignore
7d08c1b Updated the title in index.html
b4b31aa Initial Commit
```

## Compare the branches

MINGW64:/d/SCB/2022/Bootcamp July 2022/Git/myproject  You're sharing this screen

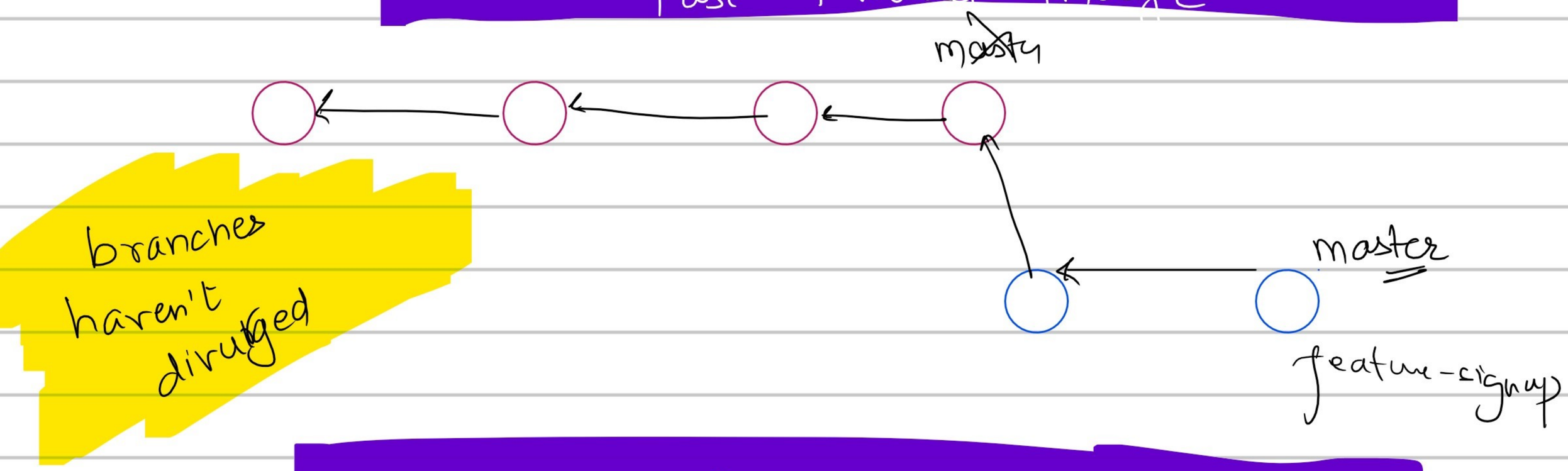
```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log master..feature-signup
commit 0b094eb9289ee7923a8f03eabe6175f69154888f (feature-signup)
Author: Zartab Nakhwa <zartab@codewithz.com>
Date:   Wed Jul 13 12:52:46 2022 +0530

    Added the signup form

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git diff master..feature-signup
diff --git a/signup.html b/signup.html
new file mode 100644
index 0000000..30dc094
--- /dev/null
+++ b/signup.html
@@ -0,0 +1,7 @@
+<html>
+  <body>
+    <form action="">
+      <label for="">Sign Up</label>
+    </form>
+  </body>
+</html>
```

## Merging

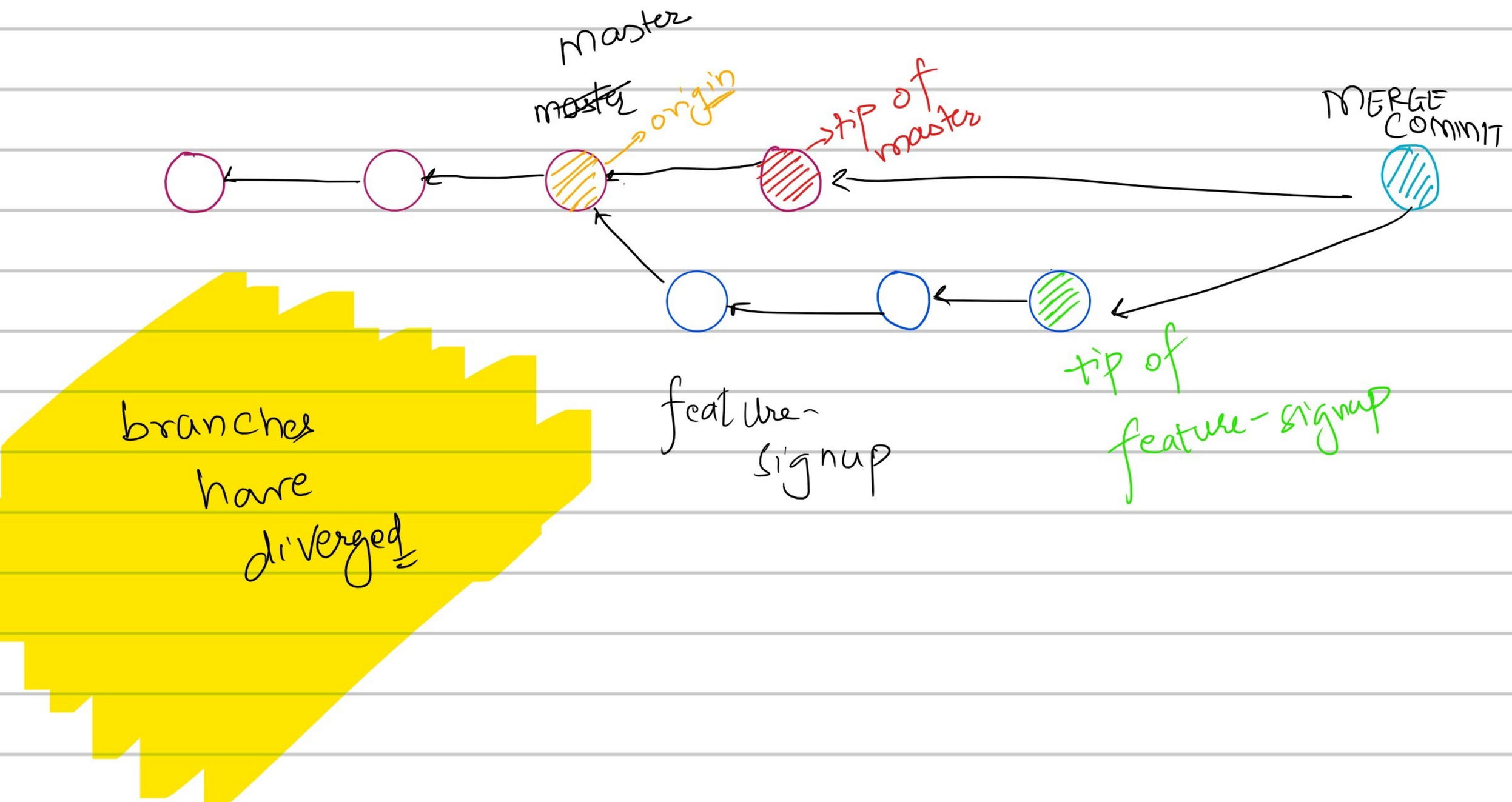
1. Fast Forward Merge
2. 3-way Merge

## Fast Forward Merge

# Fast Forward Merge




# 3-Way Merge




# Merge

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --all --oneline --graph
* 0b094eb (feature-signup) Added the signup form
* 6c34732 (HEAD -> master) Version 1
* bc58150  Updated product and index
* cd81f08 Added .gitignore
* 7d08c1b Updated the title in index.html
* b4b31aa Initial Commit

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git switch master
Already on 'master'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git merge feature-signup
Updating 6c34732..0b094eb
Fast-forward
 signup.html | 7 +++++++
 1 file changed, 7 insertions(+)
 create mode 100644 signup.html
```

# After Merge

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --all --oneline --graph
* 0b094eb (HEAD -> master, feature-signup) Added the signup form
* 6c34732 Version 1
* bc58150  Updated product and index
* cd81f08 Added .gitignore
* 7d08c1b Updated the title in index.html
* b4b31aa Initial Commit
```

# Once merge is done; branch should be deleted

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git branch -d feature-signup
Deleted branch feature-signup (was 0b094eb).

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

# Merging with --no-ff

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git branch feature-login

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git switch feature-login
Switched to branch 'feature-login'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-lo
gin)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-lo
gin)
$ git commit -m "Login Page Added"
[feature-login 122dc61] Login Page Added
 1 file changed, 5 insertions(+)
 create mode 100644 login.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-lo
gin)
$ git switch master
Switched to branch 'master'
```

```
* 122dc61 (feature-login) Login Page Added
* 0b094eb (HEAD -> master) Added the signup form
* 6c34732 Version 1
* bc58150  Updated product and index
* cd81f08 Added .gitignore
* 7d08c1b Updated the title in index.html
* b4b31aa Initial Commit

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git merge --no-ff feature-login
Merge made by the 'ort' strategy.
 login.html | 5 +++++
 1 file changed, 5 insertions(+)
 create mode 100644 login.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --oneline --all --graph
*   3d2d3fe (HEAD -> master) Merge branch 'feature-login'
|\
| * 122dc61 (feature-login) Login Page Added
|/
* 0b094eb Added the signup form
```

# Merge Conflict

both branches make changes to index.html

index.html
feature-CP committed

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git branch feature-change-password

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git switch feature-change-password
Switched to branch 'feature-change-password'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-ch
ange-password)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-ch
ange-password)
$ git commit -m "Changed title for CP"
[feature-change-password e06f327] Changed title for CP
 1 file changed, 1 insertion(+), 1 deletion(-)

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-ch
```

index.html
master committed

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git commit -m "Changed Title for Master Page"
[master bc6296b] Changed Title for Master Page
 1 file changed, 1 insertion(+), 1 deletion(-)

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

When trying to merge

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git merge feature-change-password
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master|MER
GING)
$
```

The file which has merge conflict looks like

when resolving merge conflict

do not add new changes

signup.html | login.html | index.html !

index.html > html > body > div > ?

```
1   <html>
2     <head>
3       <title>Welcome to SCB</title>
4       <link rel="stylesheet" href="styles/style.css" />
5     </head>
6     <body>
7       <div>
    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
8   <<<<<<< HEAD (Current Change)
9         <h1>Standard Chartered Bank - Master Page</h1>
10  =======
11        <h1>Standard Chartered Bank - Change Password</h1>
12  >>>>>>> feature-change-password (Incoming Change)
13        <p>
14          Lorem ipsum dolor, sit amet consectetur adipisicing elit
15          vero provident velit porro. Sint maiores facere, sequi n
16          nesciunt inventore cumque necessitatibus provident, cons
17          veritatis nulla quisquam maxime.
18        </p>
```

options from VS Code

Evil Commit

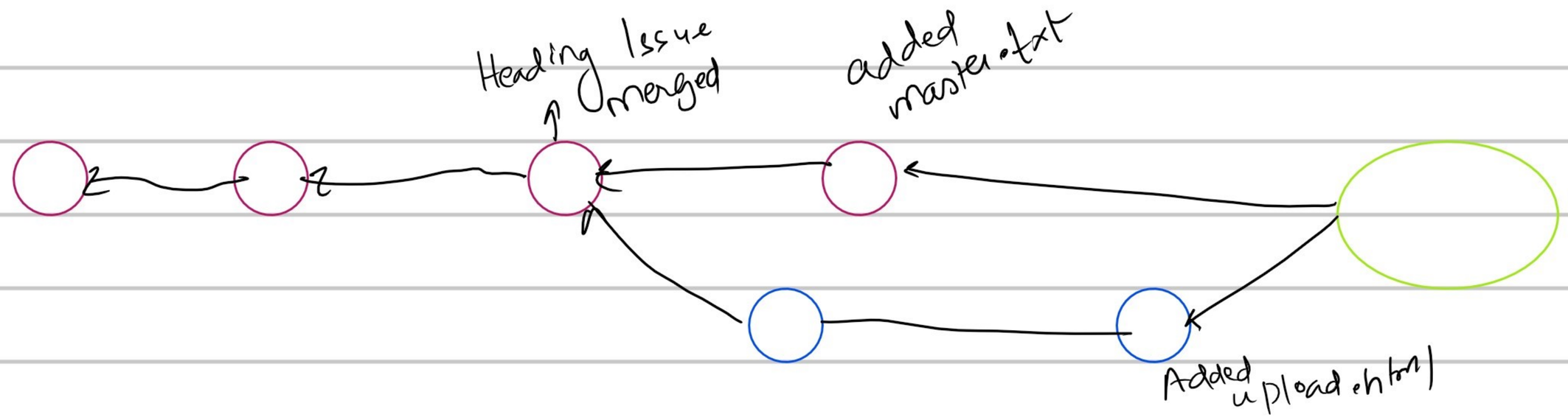
Once file is fixed

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master|MERGING)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master|MERGING)
$ git commit -m "Heading Issue Merged"
[master e6a27e0] Heading Issue Merged

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$
```

3-way-merge

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git branch -C feature-upload

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git switch feature-upload
Switched to branch 'feature-upload'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-upload)
$ # We added a new file by name of upload.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-upload)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-upload)
$ git commit -m "Added upload.html"
[feature-upload 5815b5e] Added upload.html
 1 file changed, 6 insertions(+)
 create mode 100644 upload.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-upload)
$
```

```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (feature-up
load)
$ git switch master
Switched to branch 'master'

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git add .

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git commit -m "Added master.txt"
[master 05e5b86] Added master.txt
 1 file changed, 1 insertion(+)
 create mode 100644 master-data.txt

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --oneline --all --graph
* 05e5b86 (HEAD -> master) Added master.txt
| * 5815b5e (feature-upload) Added upload.html
|/
*   e6a27e0 Heading Issue Merged
```




```
Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git merge feature-upload
Merge made by the 'ort' strategy.
 upload.html | 6 ++++++
 1 file changed, 6 insertions(+)
 create mode 100644 upload.html

Zartab Nakhwa@WINDOWS-L1BB9SS MINGW64 /d/SCB/2022/Bootcamp July 2022/Git/myproject (master)
$ git log --oneline --all --graph
*   3f25cdb (HEAD -> master) Merge branch 'feature-upload'
|\
| * 5815b5e (feature-upload) Added upload.html
* | 05e5b86 Added master.txt
|/
*   e6a27e0 Heading Issue Merged
|\
| * e06f327 Changed title for CP
* | bc6296b Changed Title for Master Page
|/
*   3d2d3fe Merge branch 'feature-login'
|\
| * 122dc61 Login Page Added
```

# Stashing

# JavaScript

* JavaScript is a dynamic, weakly typed programming language, which can be compiled at runtime.

var x = 10; ✓

x = "Zautab"; ✓

x = [1, 2, 3]; ✓

float f = 3.14f; ✓

strongly typed

f = "Zautab"; ✗

---

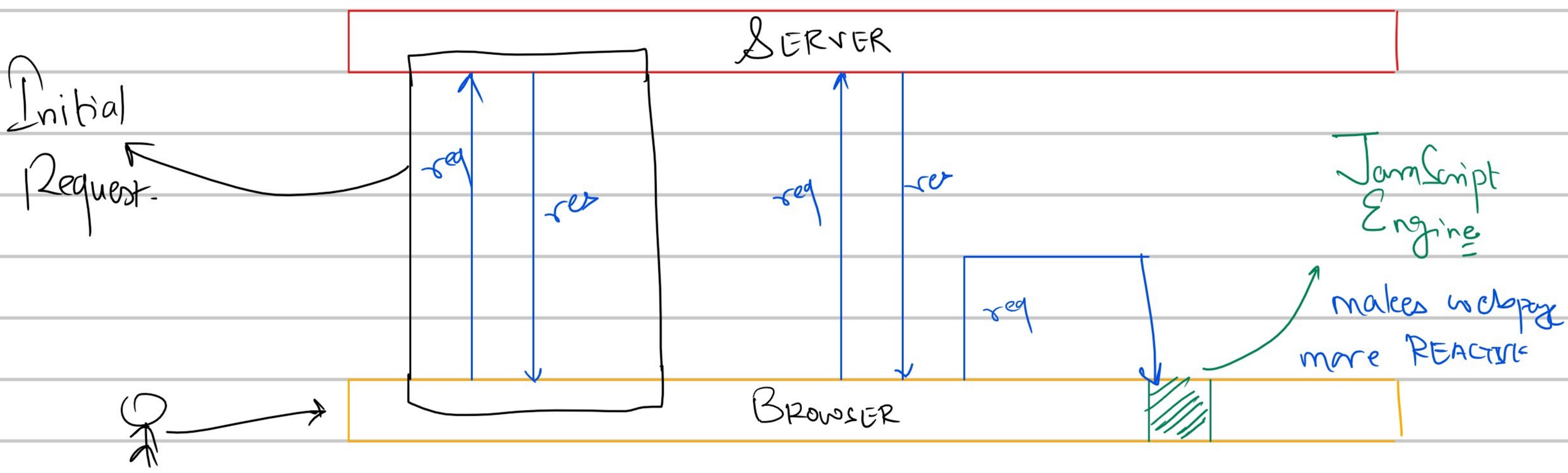
JavaScript can be executed as a part of web-page in a browser or on any machine.

JavaScript → makes the page dynamic

→ change the content of the page directly from the browser.

JavaScript → LiveScript

Java & JavaScript are completely independent



# How JavaScript works.

Computer doesn't understand JavaScript but BROWSER does

↓

JavaScript Engine

program that executes JavaScript code

* What can JavaScript do

① Access the DOM elements

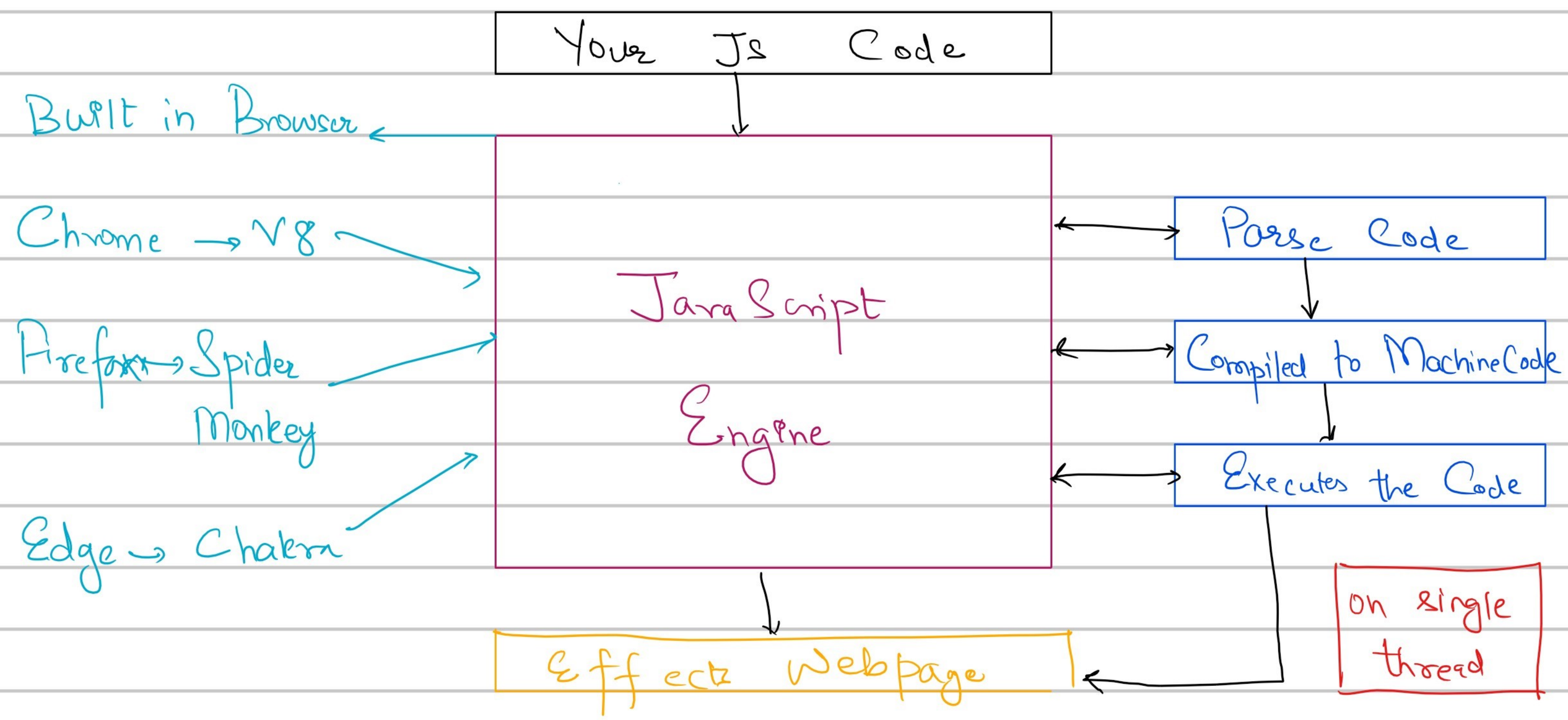
② Network Requests

③ Event Handling

↳ Mouse or Keyboard events

## JavaScript Engine

↳ a program written in C++;
it goes through all the JavaScript code line by line & transforms it into something computer can understand.



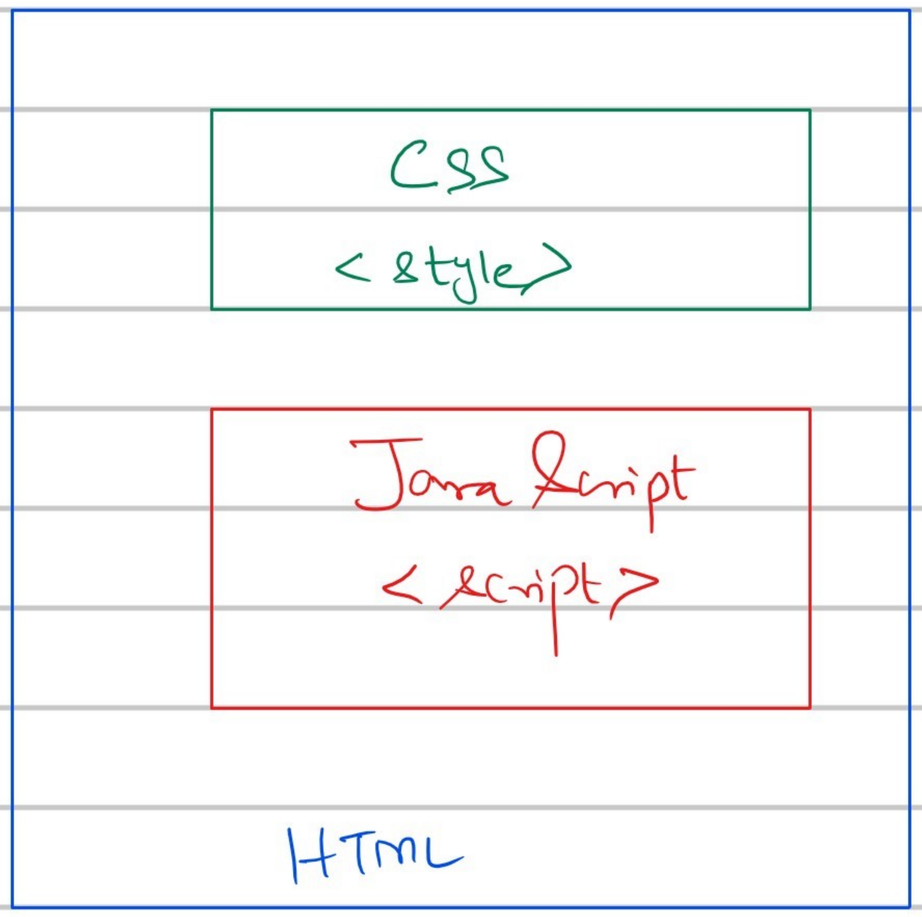
2008 → Google open sourced V8

2009 → Ryan Dahl → Node.js using V8

{ it made JS available for server side }

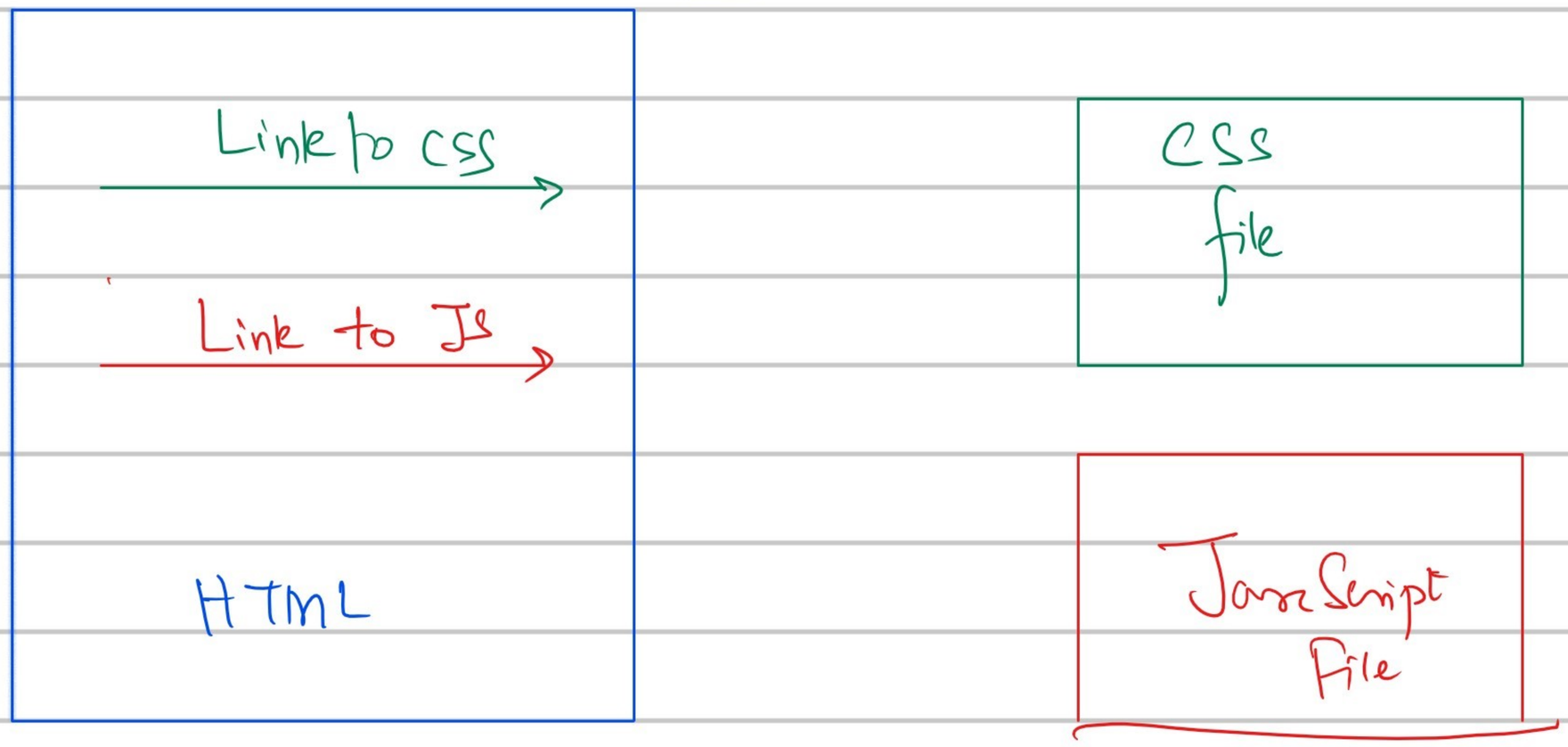
a runtime environment for JS outside the browser.

## How JavaScript fit in with HTML




Can be done; but not a good approach

## Seperation of Concerns



# Data Types in JavaScript

| | | |
|---|---|---|
| number | 2, -3, 23.45 | → calculations |
| string | 'Hi', "Hi", `Hi` | → display the output |
| boolean | true, false | → dealing with conditions |
| object | {name: 'Zartab', age: 31} | → grouping/organizing the data |
| array | [1, 2, 3, 4] | → listing the elements |

## Important things to know before coding JS programs

var x;

typeof → what is the data-type of x

**undefined**

→ default value for uninitialized variable

→ Never assign undefined manually

**null**

→ Never assumed by default

→ can be assigned explicitly ↓ reset/clear

**Not Entirely Equal**

Variables → a container that holds the value which can vary.

var x = somevalue

let

const

```
let username = "Codewithz";
username = "zartab", ✓
```

A data container / data storage where the value can change

```
const totalCandidates = 20;
totalCandidates = 15; ✗
```

A data container / data storage where a value cannot change.

# Functions

'CODE ON DEMAND'

collection of statements which needs to work together to achieve a specific task

## DEFINE A FUNCTION

```
function greetUser(name){
    alert("Hello" + user);
}
```

for a function to execute it always has to be invoked

## CALL A FUNCTION

```
greetUser('Zantab');
greetUser("Tom");
```

## Syntax of a function

```
function nameOfFunction( parameter ) {
    ...
    return someValue;
}
```

- function → keyword
- nameOfFunction → unique Identifier
- parameter → optional
- ... → statements
- return someValue; → optional

If a function wants to return something; last statement should be a return.

define

```
function square(num){
    return num*num;
}
```

call

```
let sq = square(4);
```

# Functions are like first class passengers in JS

## * Function Expressions *

→ a function can be assigned to a variable

```
const cube = function (num) {
                return num*num*num;
                }

cube(3);
```

Scopes inside a function / program

global variable

local variable

```
var x;
function f1() {
    var x1;
    }
function f2() {
    var x2;
    }
```

# Array → JavaScript

Array in JavaScript can hold any type of data

let a = [ Tom, Alex, Mike, John ]

Length → 4

| Index → | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| | Tom | Alex | Mike | John |

Index [ 0 ——— length-1 ]

Splice function

```
splice(start, deleteCount, item1, item2, itemN)
```

0 | 0

elements which will be added

## Parameters

`start`

The index at which to start changing the array.

If greater than the length of the array, `start` will be set to the length of the array. In this case, no element will be deleted but the method will behave as an adding function, adding as many elements as items provided.

If negative, it will begin that many elements from the end of the array. (In this case, the origin `-1`, meaning `-n` is the index of the `n` th last element, and is therefore equivalent to the index of `array.length - n`.) If `start` is `-Infinity`, it will begin from index `0`.

`deleteCount` Optional

An integer indicating the number of elements in the array to remove from `start`.

If `deleteCount` is omitted, or if its value is equal to or larger than `array.length - start` (that is, if it is equal to or greater than the number of elements left in the array, starting at `start`), then all the elements from `start` to the end of the array will be deleted. However, it must not be omitted if there is any `item1` parameter.

If `deleteCount` is `0` or negative, no elements are removed. In this case, you should specify at least one new element (see below).

`item1, item2, ...` Optional

a, b, c

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

numbers.splice( 2, 0, 'a', 'b', 'c')

## Higher Order Functions

function which takes another function as an input or return a function from itself.

1, 2, 3, 4, 5

```
array.forEach( function() {
                          } )
```

1, 2, 3, 4, 5

```
array.forEach( function( item, index) {
                                      } );
```

pass each element

```
array.filter( function( item, index) {
                                      } )
```

true / false

return a boolean value

1, 2, 3

```
array.map( function( item, index) {
                                  return something;
                                  } )
```

# * String Interpolation *

```
let s1 = "SCB"

let s2 = 'SCB'

let s3 = `SCB`
```

```
let name = "Ajay"
let company = "SCB"
let bankId = 12345
```

```
let line = name + " works as " + company + " his bank id is " + bankId;

let line = `${name} works at ${company} his bank id ${bankId}`
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 → | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | S | t | a | n | d | a | r | d |
| 1 → | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

0 based

1 based

slice (begin, end)

(2, 5)

# Objects in JavaScript

→ group various values into a single object

```
let objName = {
    key : value,
    key : value,
    key : value,
    key : value,
    key : value,
}
```

key → string

value → can be of any type.

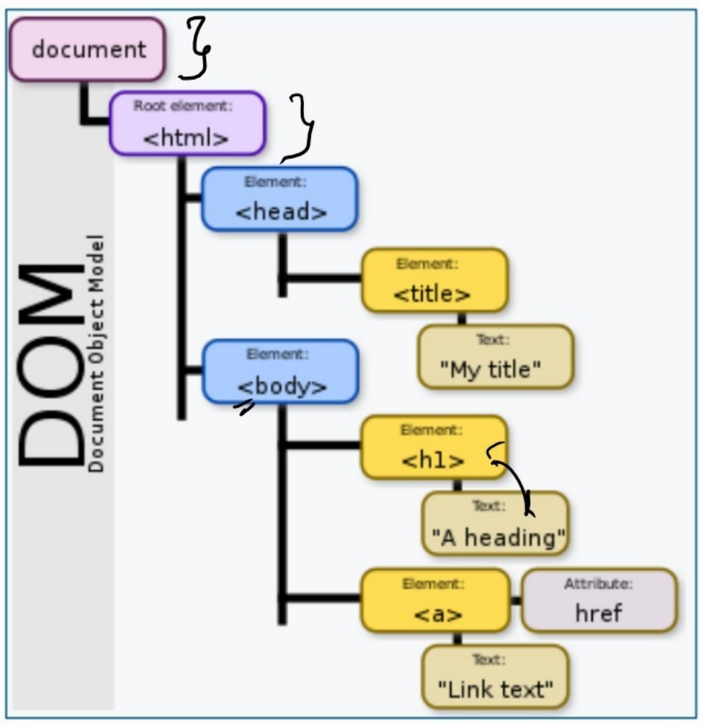
it can also be a function

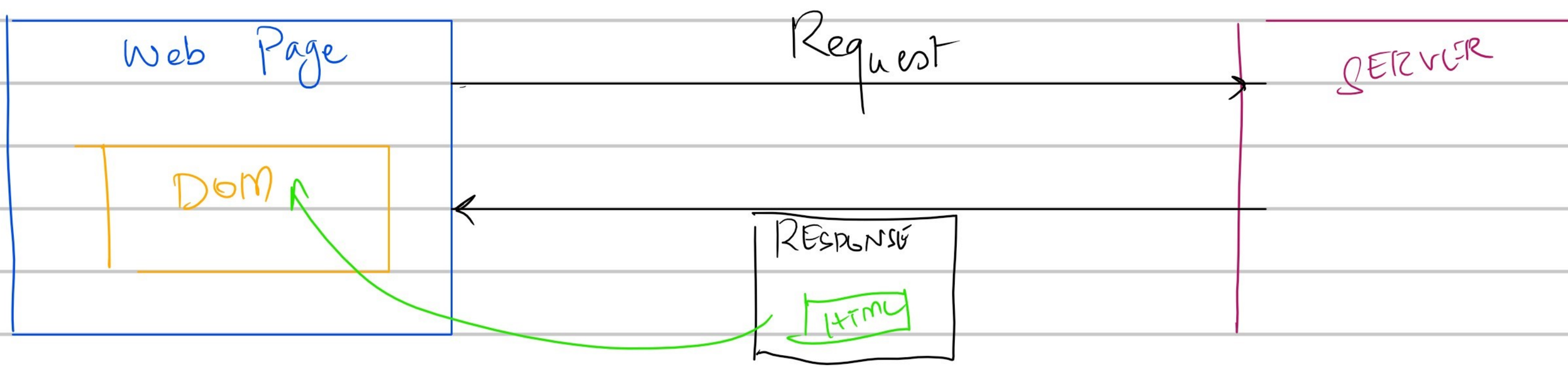
## DOM

→ Document Object Model

object which represents a web-page in a way in which elements are structured in a web page

```
<html>

<head>
    <title>My Title</title>
</head>
<body>
    <h1>A heading</h1>
    <a href="#">Link Text</a>
</body>

</html>
```



document

getElement
By Id
By Tag
By class name

getElements

querySelector

querySelectorAll

# Arrow Functions

=> ↳ cleaner way of creating traditional JS functions

```
(parameter1, parameter2, ....., parameterN) => {
    statement(s)
}
```

Arrow functions are either assigned to a variable or they are used as arguments to be passed to some function.

# Class & Object

Class → a template or a blue print which is used to describe an object

Object → instance of a class

Human (class) → Mary, Tom, Alex (objects)

# Casing

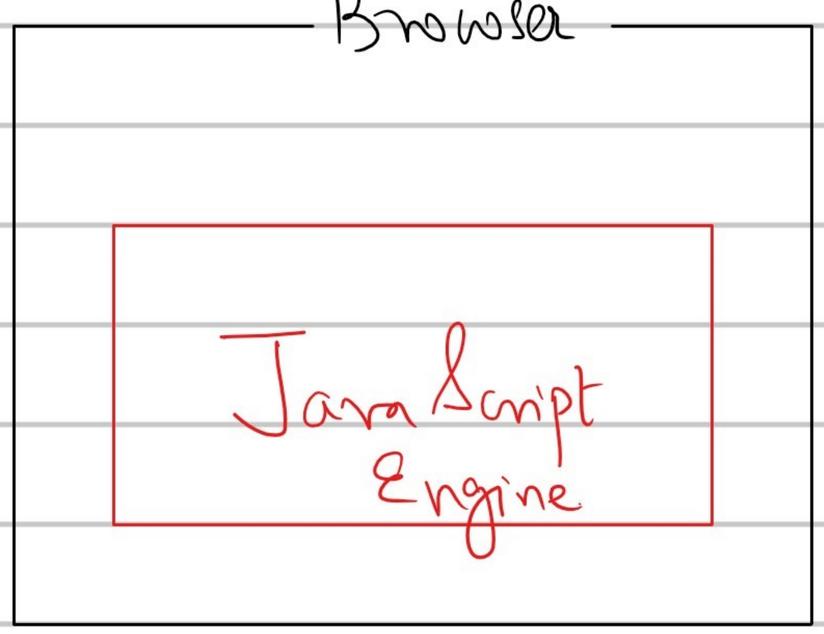
1. camelCasing → oneTwoThreeFour → variable / functions

2. Parcal Casing → OneTwoThreeFour → class

# Node

A runtime environment for executing JavaScript code outside browser

~~In~~

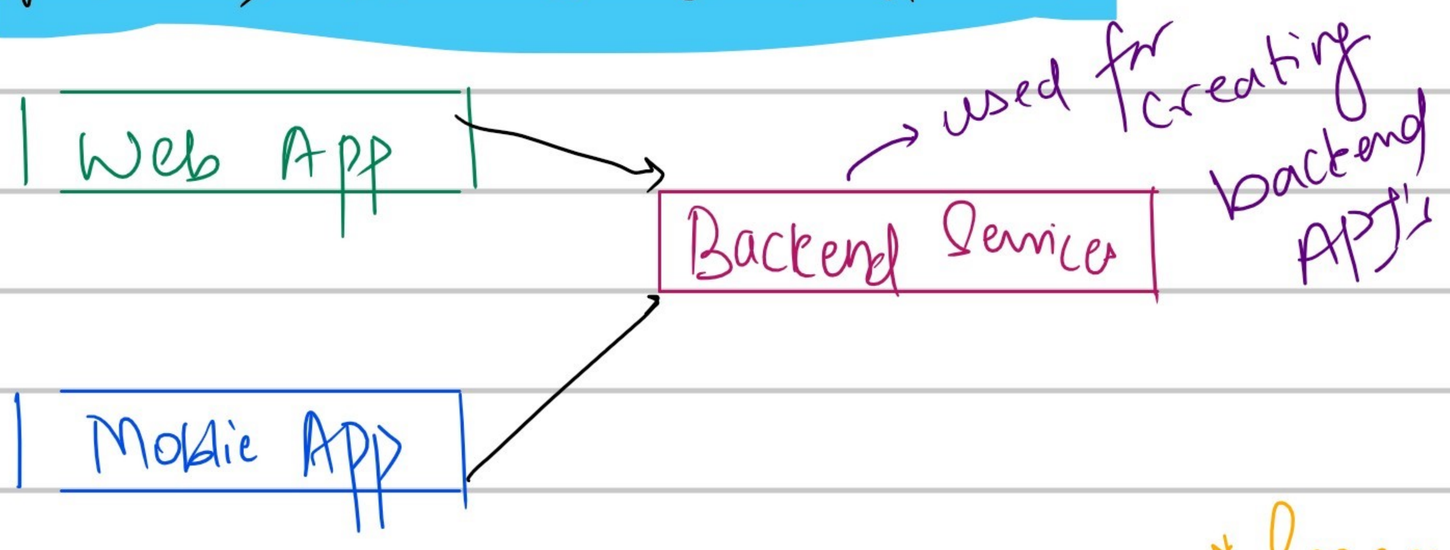



In 2008 Google open-sourced V8 → JS Engine for Google Chrome

Ryan Dahl

## * Why Node JS *

Where is Node JS used a lot?

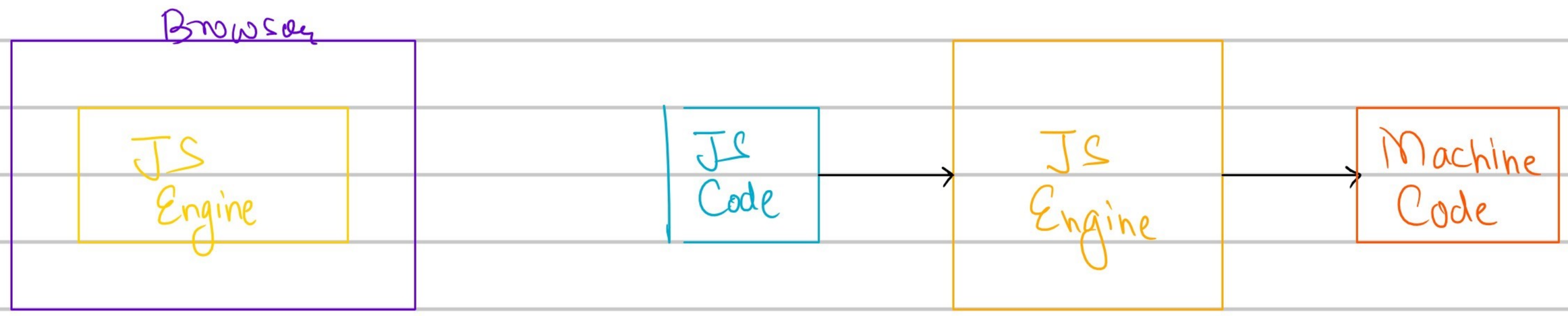



Features

* super fast
* highly scalable
* JavaScript Everywhere
* cleaner & consistent codebase
* largest ecosystem of opensource library.

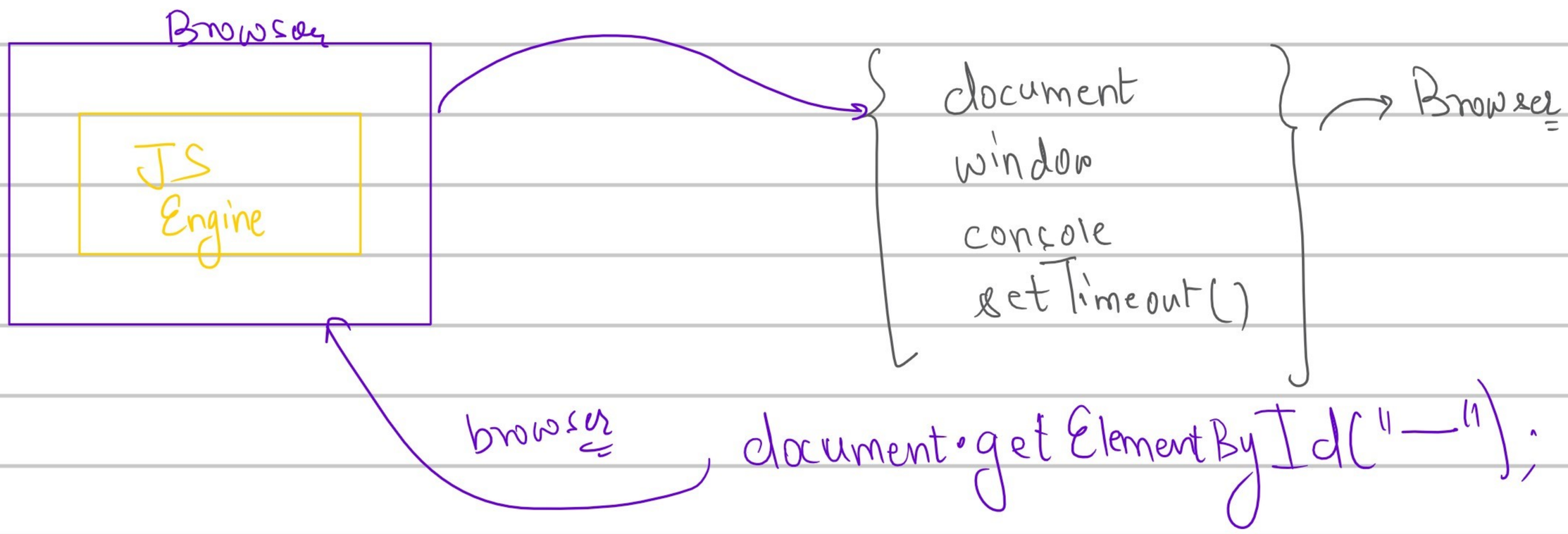
# Node Architecture

## Before 2009



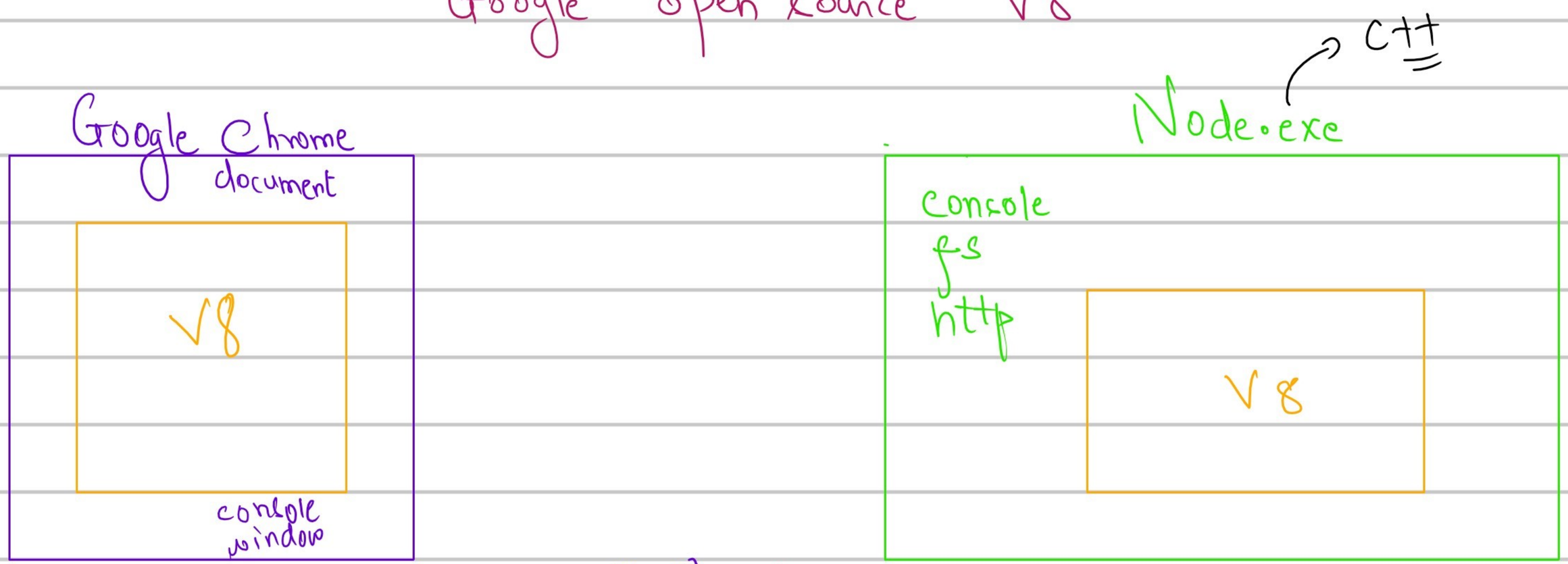

Chrome → V8
Firefox → SpiderMonkey
edge → Chakra

Runtime Environment




## Post 2009

Google open source V8




document.getElementById("p"); ✓
fs ✗
http ✗

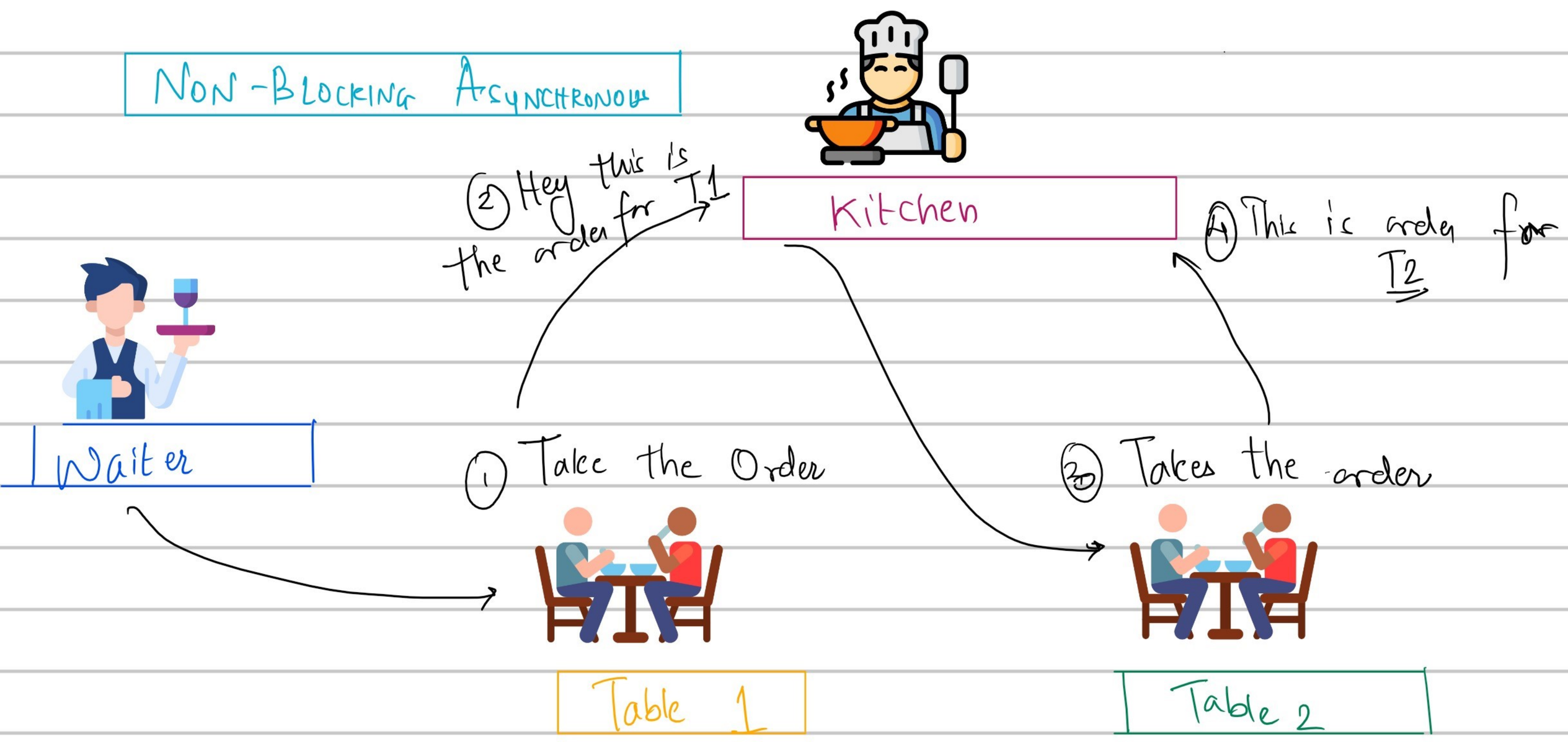
document.getElementById("p") ✗
fs ✓
http ✓

* Node is not a programming language
* Node is not even a framework like Spring Boot / Django

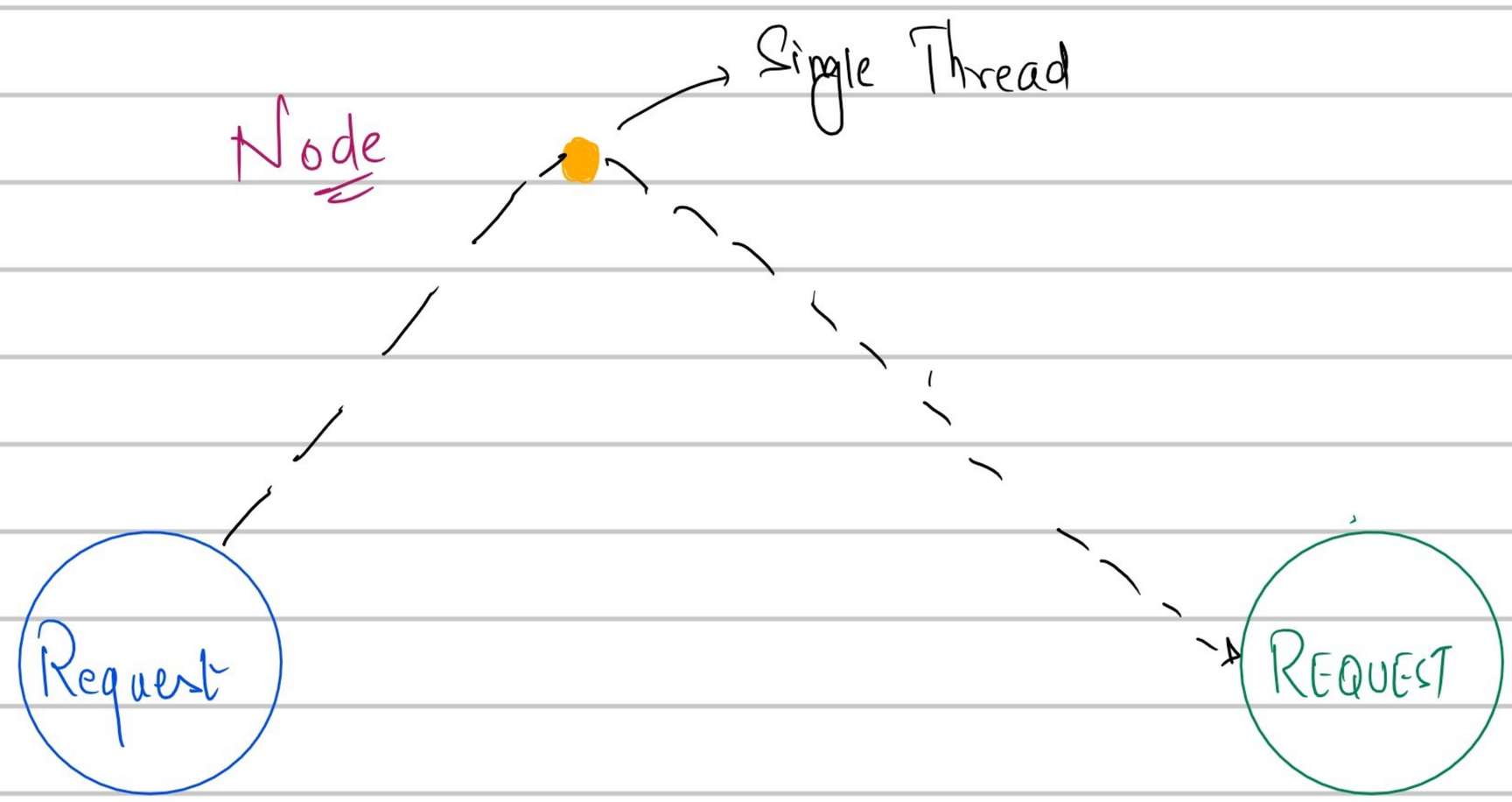
Node is a runtime environment for executing JavaScript

---

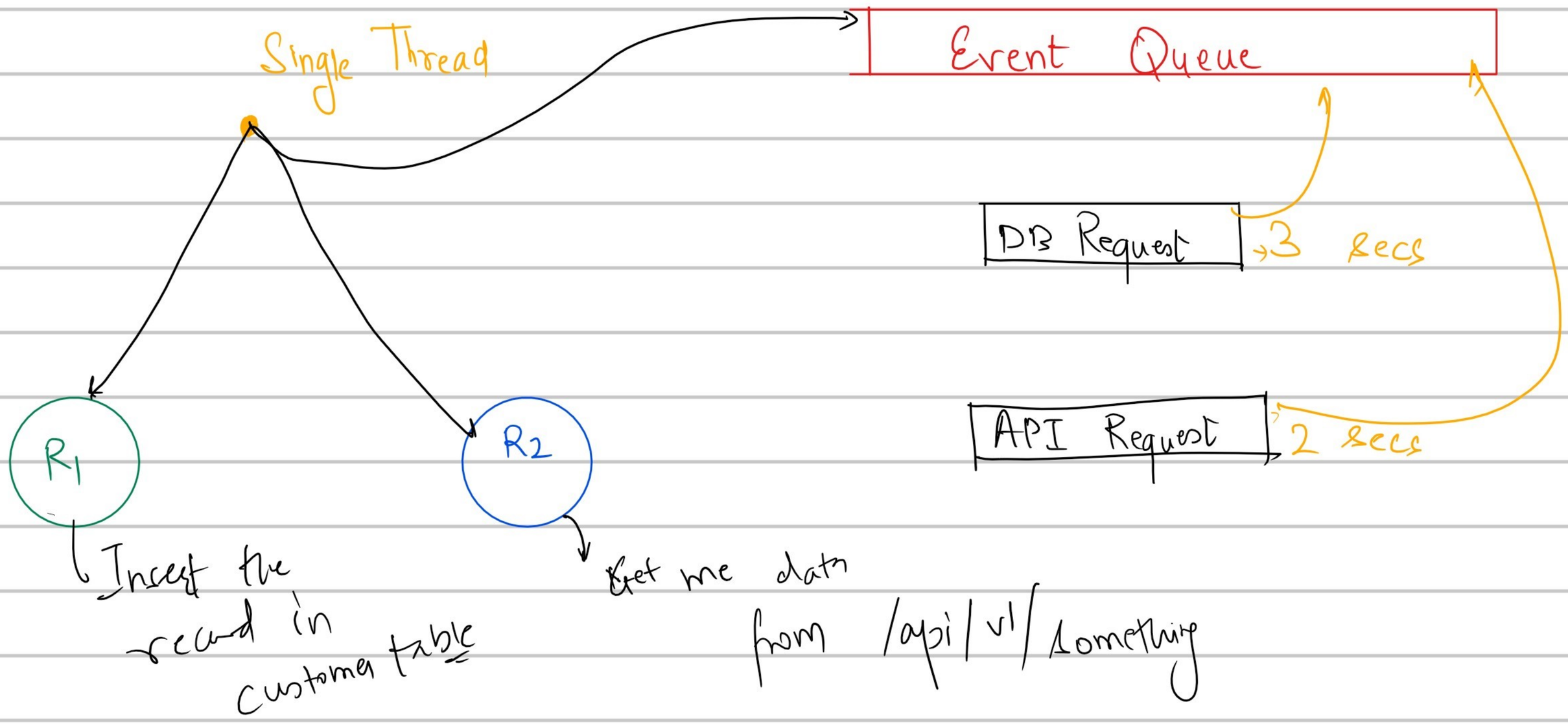
NON-BLOCKING ASYNCHRONOUS




* Node is Single Threaded *

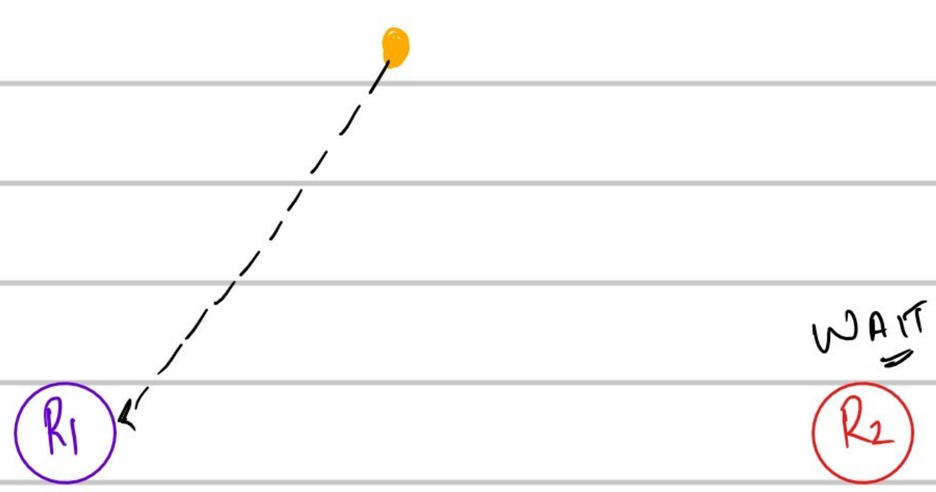


# Node applications are ASYNCHRONOUS by default.




* Node is ideal for I/O Intensive App.

* Node is not ideal for CPU Intensive App



→ Node

Core Module → http, fs, url

Local Module → user defined

Third Party Module → npm

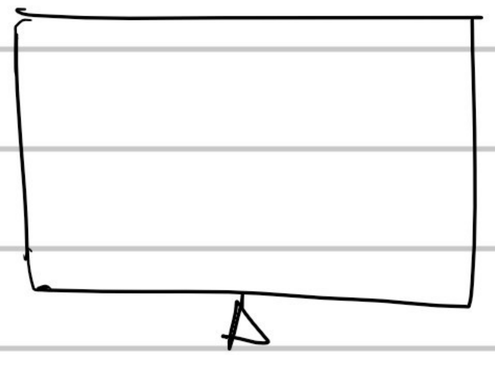
## Port

C ±32767
0

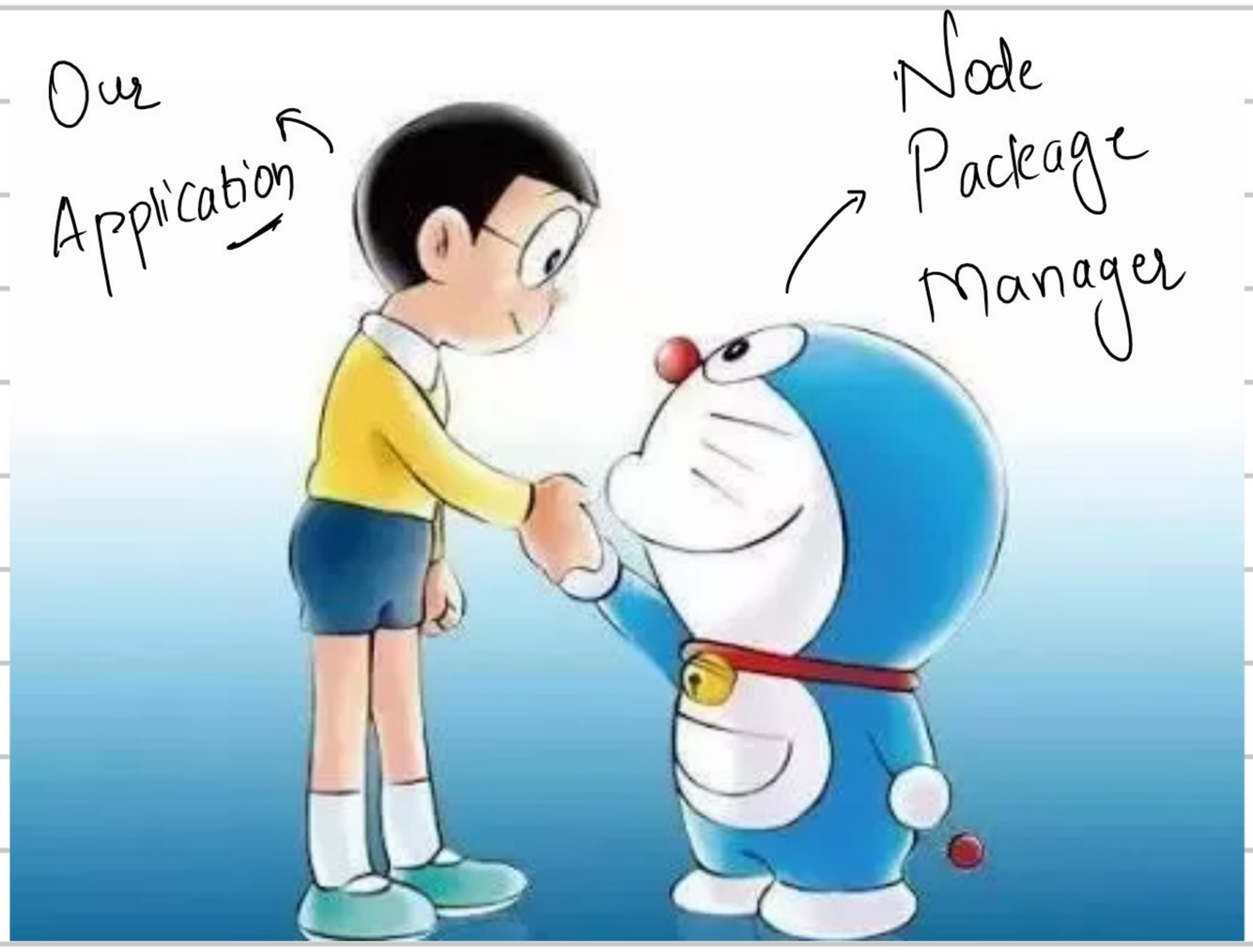
Port is a 2 byte unsigned integer which idenfies a software service on a hardware construct.

0 — 65535

0 — 1024 | 1025 — 65535



# * Node Package Manager - npm *




When in npm and for initializing a node project

my-project → npm init

npm-demo → package.json

↓ where all the meta-data of my project goes

# Installing an npm package.

| Global | Local |
| --- | --- |
| anywhere | inside project folder |
| npm install package-name -g | npm install package-name |

**Location**

C:\Users\username\AppData\Roaming\npm\node_modules

current-folder ↓ node_modules

**Version**

npm install package-name@version (x.x.x)

@latest

## Semantic Versioning

Major . Minor . Patch

6 . 3 . 0

↳ found a bug. fixed it

6 . 3 . 1

# Semantic Versioning

Major . Minor . Patch

6 . 3 . 0

↳ found a bug. fixed it

6 . 3 . 1

↳ brought some features but it doesn't change the way of working

6 . 4 . 0

↓

A major overhaul was brought it. It changed the way application/module worked

7 . 0 . 0

After npm install jquery@3.3.1

npm install underscore

package.json | index.js

index.js > ...

```
const underscore = require("underscore");
```

→ when a module is used

① Core Modules → not

② File or Folder → ./underscore → not

③ Node_modules

Version Upgrade

How node decides for update

package.json | index.js

package.json > { } dependencies > underscore

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"author": "Zartab Nakhwa",
"license": "ISC",
"dependencies": {
  "jquery": "^3.3.1",
  "underscore": "~1.13.4"
}
}
```

5
0
7

^ → do not update beyond the major

~ do not update beyond minor

Npm - Outdated - update

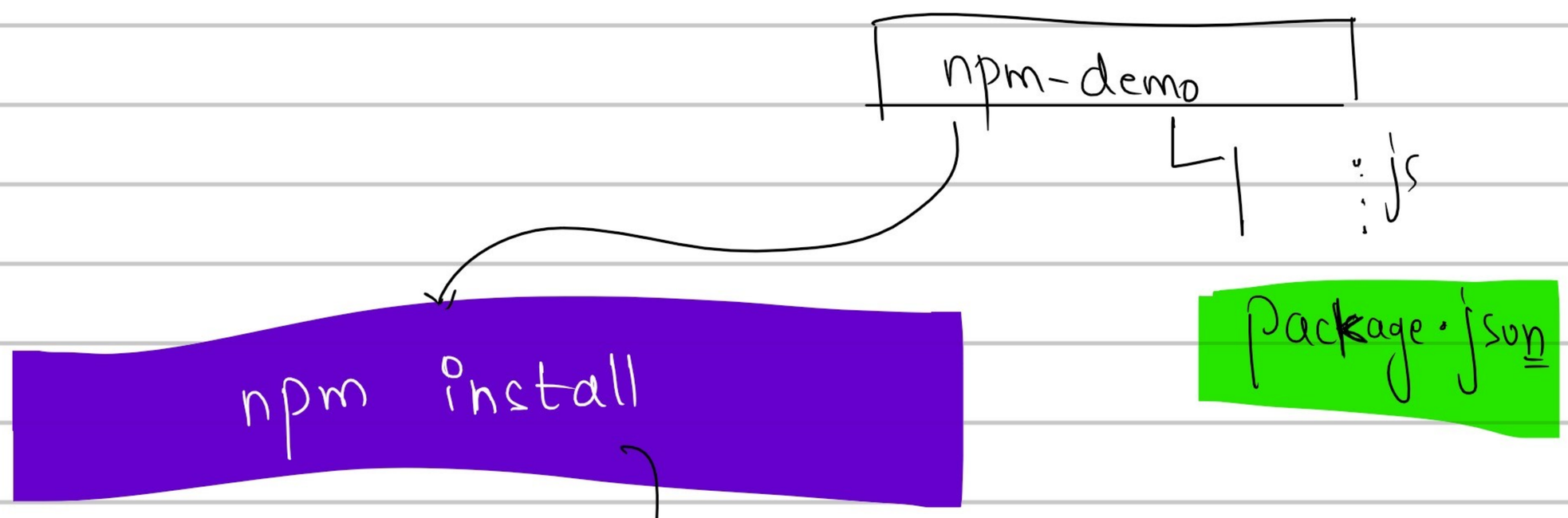
```
D:\SCB\2022\Bootcamp July 2022\NODE_NPM_JSON\npm-demo>npm outdated
Package  Current  Wanted  Latest  Location              Depended by
jquery     3.3.1   3.6.0   3.6.0  node_modules/jquery   npm-demo

D:\SCB\2022\Bootcamp July 2022\NODE_NPM_JSON\npm-demo>npm update

changed 1 package, and audited 3 packages in 931ms

found 0 vulnerabilities
```

# Uninstall a module

```
D:\SCB\2022\Bootcamp July 2022\NODE_NPM_JSON\npm-demo>npm uninstall jquery

removed 1 package, and audited 2 packages in 881ms

found 0 vulnerabilities

D:\SCB\2022\Bootcamp July 2022\NODE_NPM_JSON\npm-demo>
```

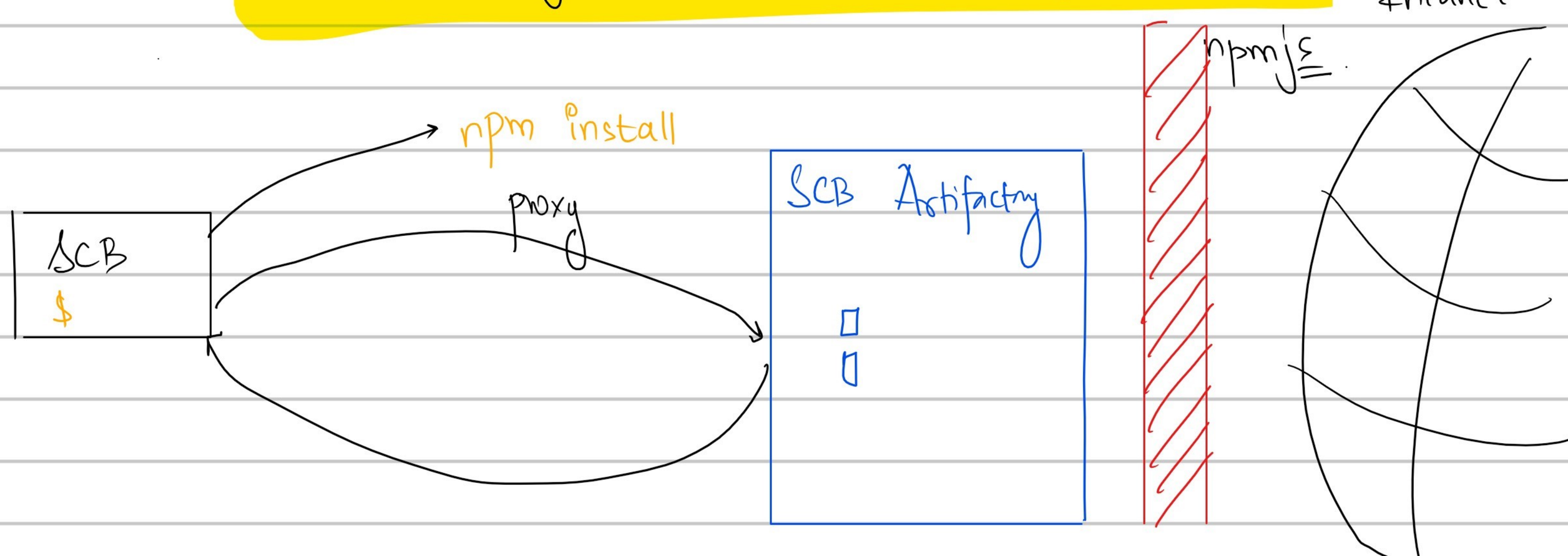
# Transport a node project

node_modules → .gitignore




download all the required libraries using package.json

# Why proxy is used in organization?



# JSON

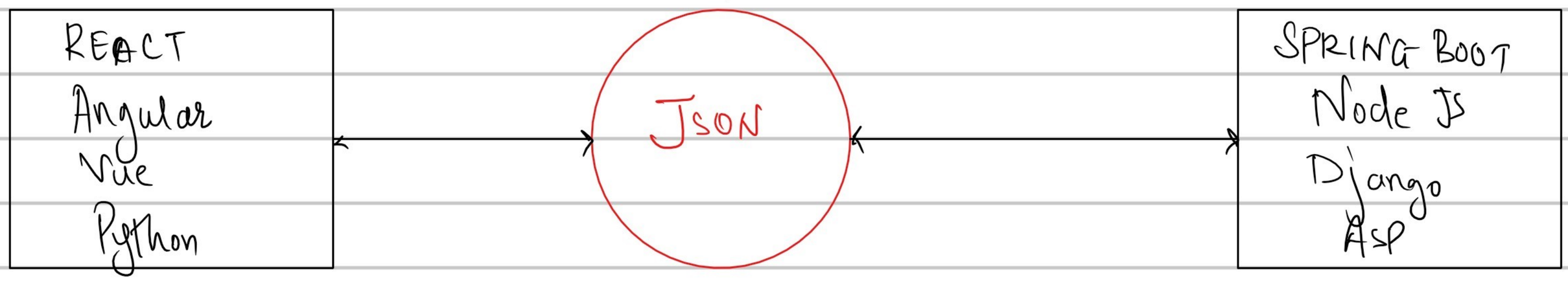
Java Script Object Notation

→ is a minimal, readable format for structuring the data

data in JSON is always in a key-value pair

JSON is

JSON is English of Programming language



# JSON

| key | value |
|---|---|

a key is always written in " " (string quotation marks)

a value can be a string, number, boolean, array, object

```
{
  "key" : value,
  ___ : ___,
  ___ : ___,
  ___ : ___
}
```

- a string
- a number
- boolean
- array
- object

## JSON → part of browser

JavaScript Object → [ stringify ] → JSON String

JSON String → [ Parse ] → JavaScript Object

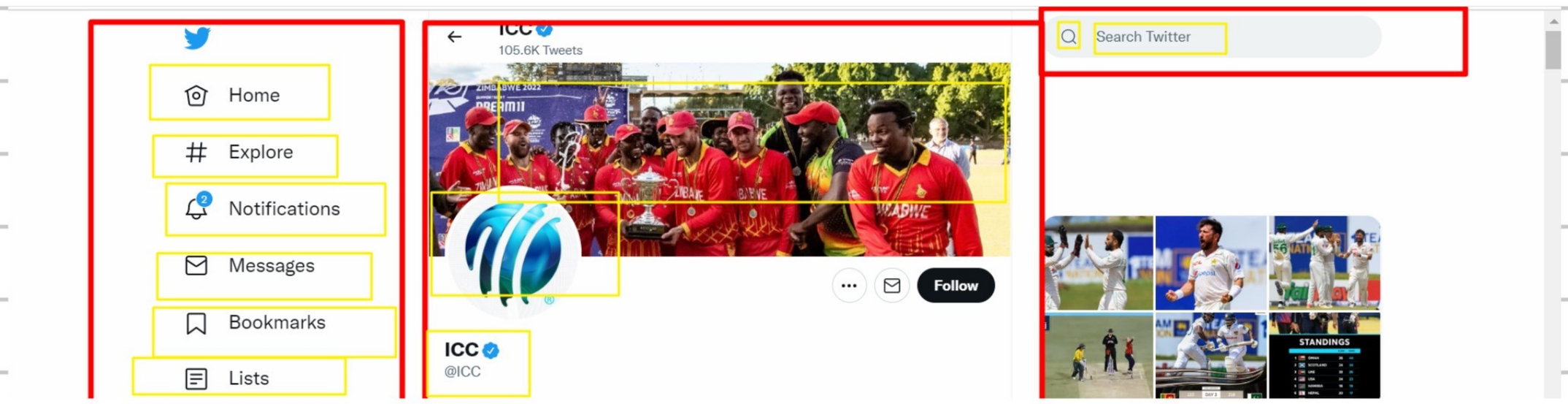
# * REACT *

JavaScript based front end / UI library.

SPA → Single Page Application's

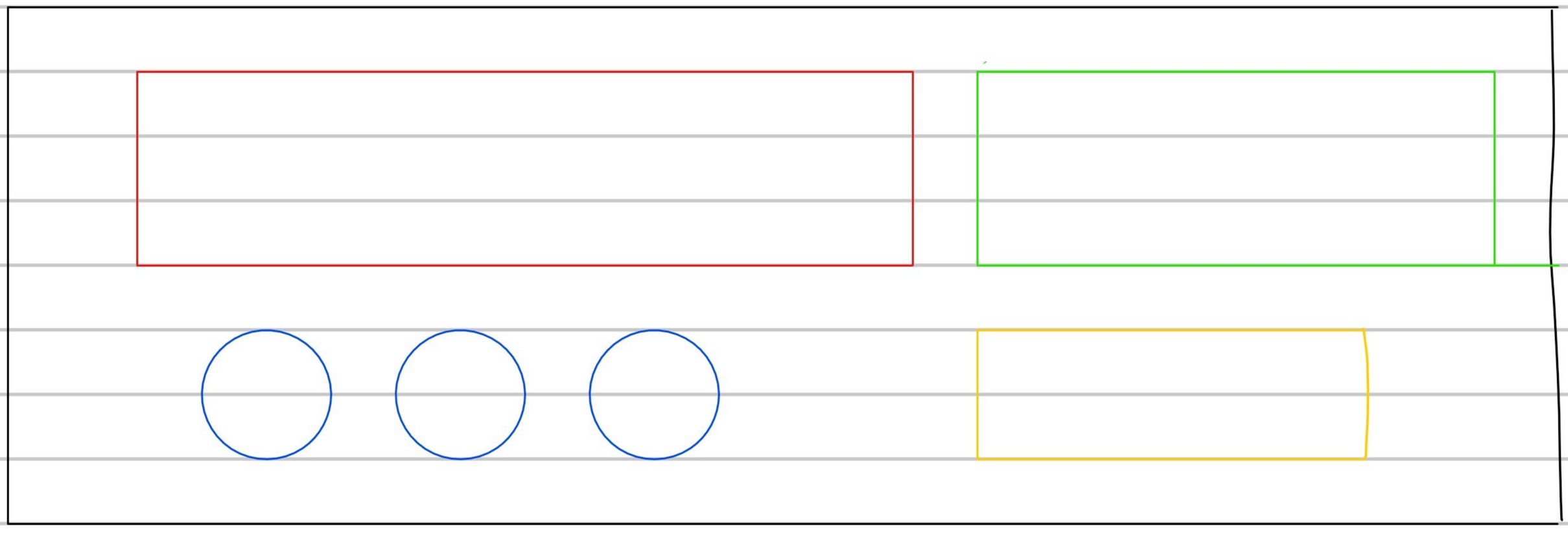
→ React was developed at Facebook

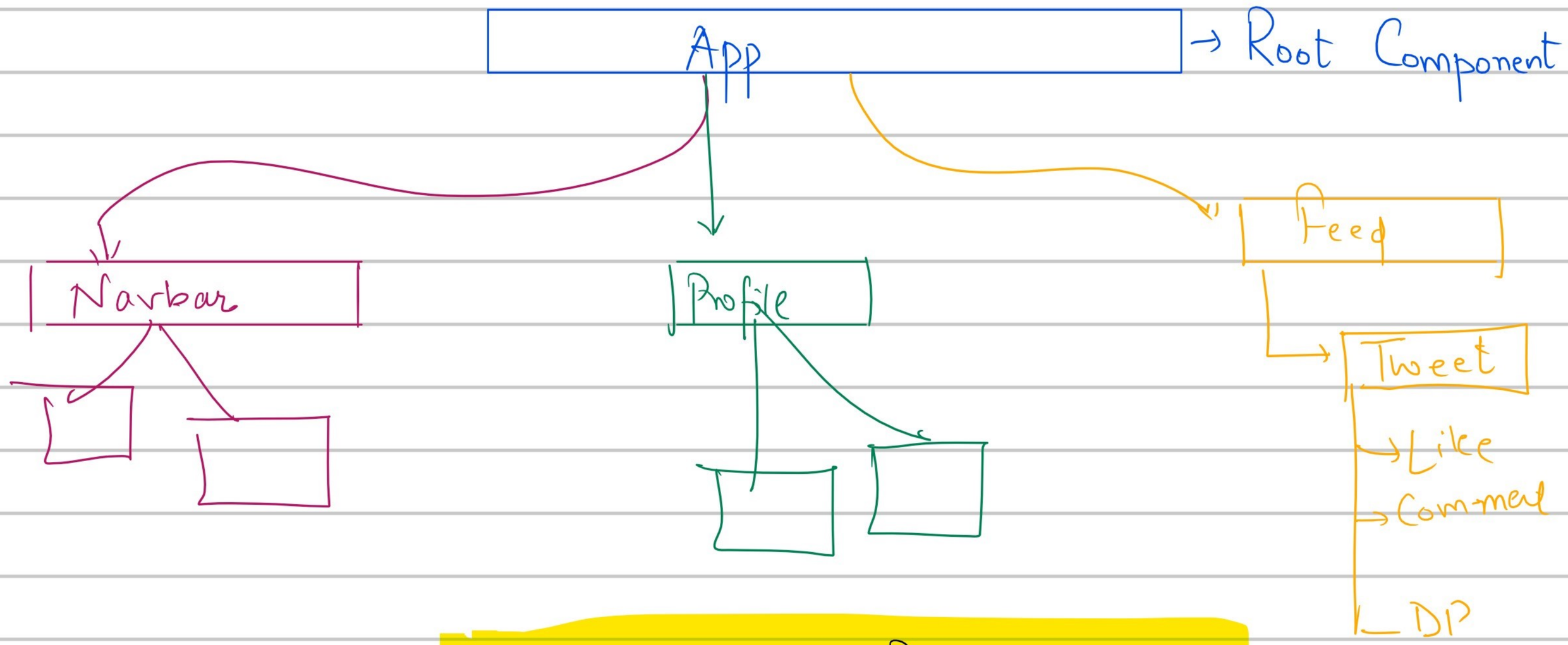
React is made up of Components!!



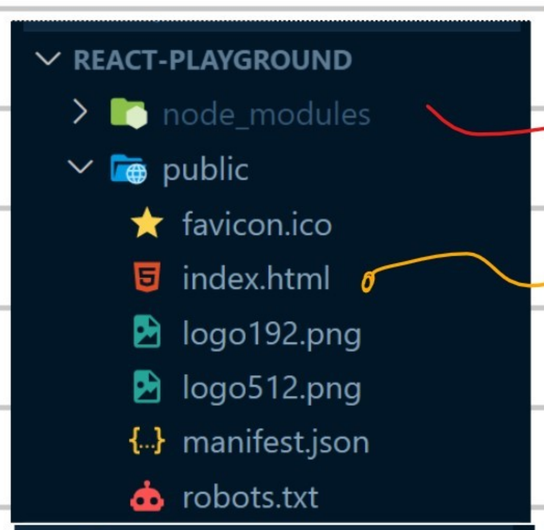

Component

Piece of UI ; who has it's own STATE & LIFECYCLE



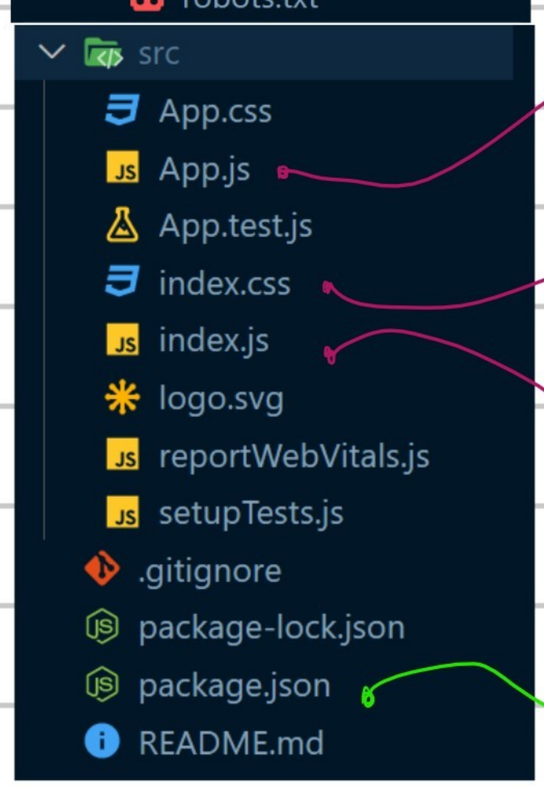



# Structure of React Application




contains the library packages.

renders the data given by index.js




Root Component → React Component.

Styles index.html

main.js file which loads the Root Component

configuration file

## * React Component *

- Class Component → Stateful
- Functional Component → Stateless

hooks → Stateful

```
class Tweet extends Component {

  state = {

  }

  render() {

  }
}
```

all the variables which forms the state of this Component; goes here

UI loads here

returns a React Component

Class Component

---

```
function FeedsPage() {

  return (

  );
}
```

UI loads here

React Component is returned

Functional Component

Go to react application folder

npm start

index.js

src > index.js > ...

```javascript
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

imported App.js

root.render(App)

App.js

src > App.js > ...

```javascript
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" /
        <p>
          Edit <code>src/App.js</code> and save to reload
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}
```

index.html

public > index.html > ...

```html
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the b
      Only files inside the `public` folder can be referenced from the HTM

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" w
      work correctly both with client-side routing and a non-root public U
      Learn how to configure a non-root public URL by running `npm run bui
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.
```

div / id → root

formance in your app,
tWebVitals(console.lo
Learn more: https://b

JSX → Java Script XML

JavaScript

HTML

const element = <h1> React </h1>

JSX is a JavaScript extension syntax used to write code in React. It binds HTML & JavaScript together.

const name = " SCB";

const e = <h1> Hello {name} </h1>

displayName() {

}

<button onClick={displayName} />

Browser → HTML ✓

JS ✓

JSX ✗ → not a valid JavaScript code

JS compiler/transpiler

JSX → Babel → JavaScript

## Imports & Exports

abc.js

```
export const xyz = 10;

export function f1() {
}

default export class Abc {
}
```

main.js

default → no {}

named → { }

```
import {xyz, f1}, Abc from './abc'
```

## Usage of React Components in other components

App.js

```
<NameComponent />
```

NameComponent.js

index.html | App.js M | HelloReactComponent.jsx 1, U

src > components > HelloReactComponent.jsx > HelloReactComponent

```
import React, { Component } from "react";

class HelloReactComponent extends Component {

    state = {

    }

    render() {
        return (

        )
    }
}
```

when some value is returned from a React render() ; you need to make sure it is a single parent

 ✓

<h1>
<h2> </h2> ✗
</h1>

```
export function HelloFunctionalComponent() {

    return (
        <div>
            <h3>Hello from Functional Component</h3>
        </div>
    )
}
```

* Why functional component *

① Functional components are easy to read & write

② It is easy to separate container & presentational Components

③ It is 100% JS

④ Can use best practices

⑤ Increases the application performance

# Props

→ Normal HTML have attributes

attribute

<img src="——" />

→ React Component have props

<NameComponent name="Zartab" />

→ prop → equivalent to attribute ;

① Attributes in HTML are only string.

In HTML ~~a~~ attribute names are pre-fixed

props can be any JS object

String, object, another component, function

*

props in React can have any names.

<NameComponent
name="Zartab" />

props are passed to

① Class Component.

→ available in class component as an object → props

② Functional Component

→ available as an arguments.

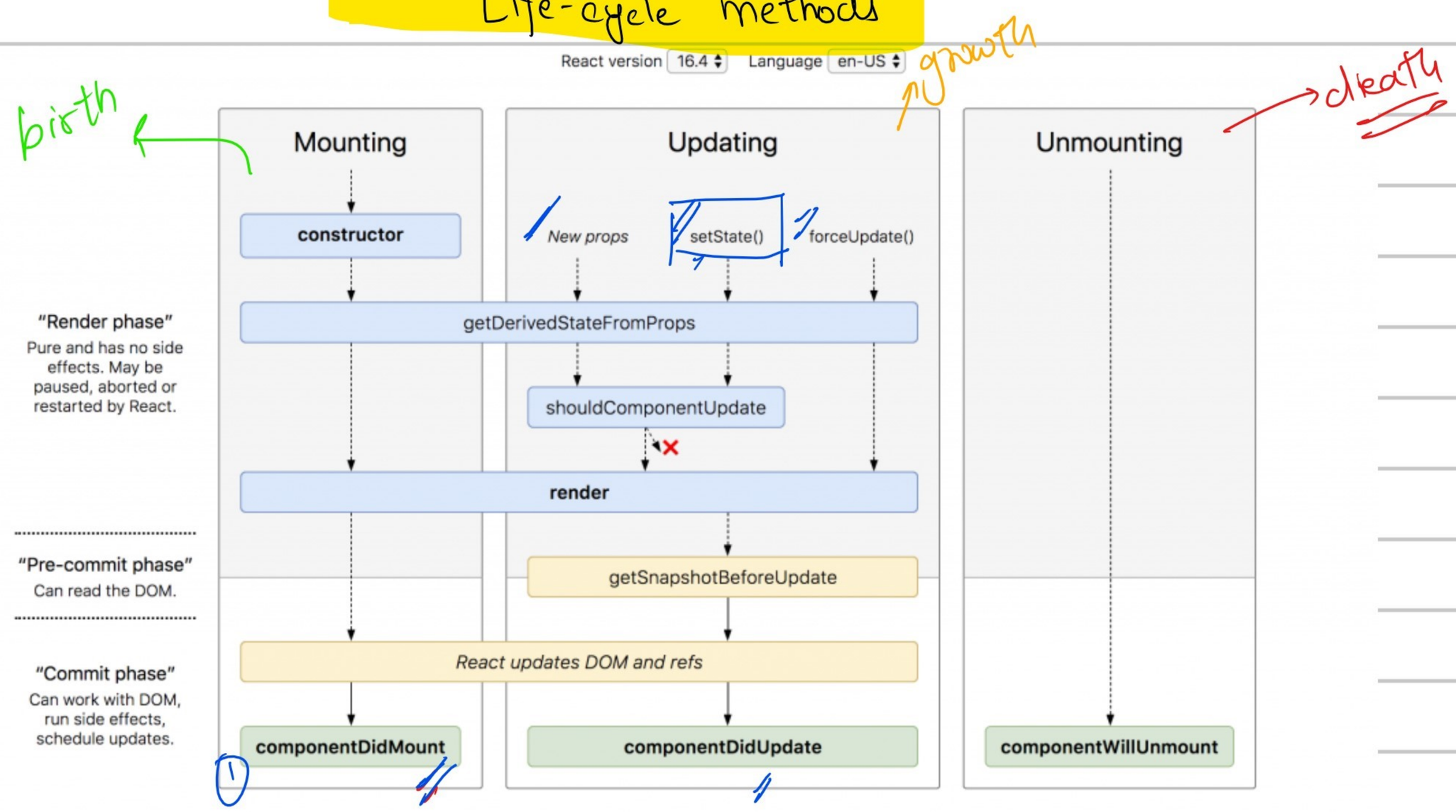
# State

State is a plain JavaScript object used by React to represent information about the component's current situations.

```
class SomeComponent extends Component{
  state={

  }

  render(){

  }
}
```

"State is like props; but it is private & it is fully controlled by the Component."

## Life-cycle methods




## TickerComponent

1. React → TC's render()
2. When TickerComponent's output is inserted in DOM, React → componentDidMount()
3. componentDidMount() → asking the browser to set up a timer & the component's tick() every second
4. Every second Browser → tick() → updates the state setState().
5. setState() → render

# Hooks & State

Hooks were introduced in React 16.8

Special type of functions, which allows us to break a class component into smaller functional components.

* In previous versions of React, if a component needed state, only way was using class component

## useState → hook

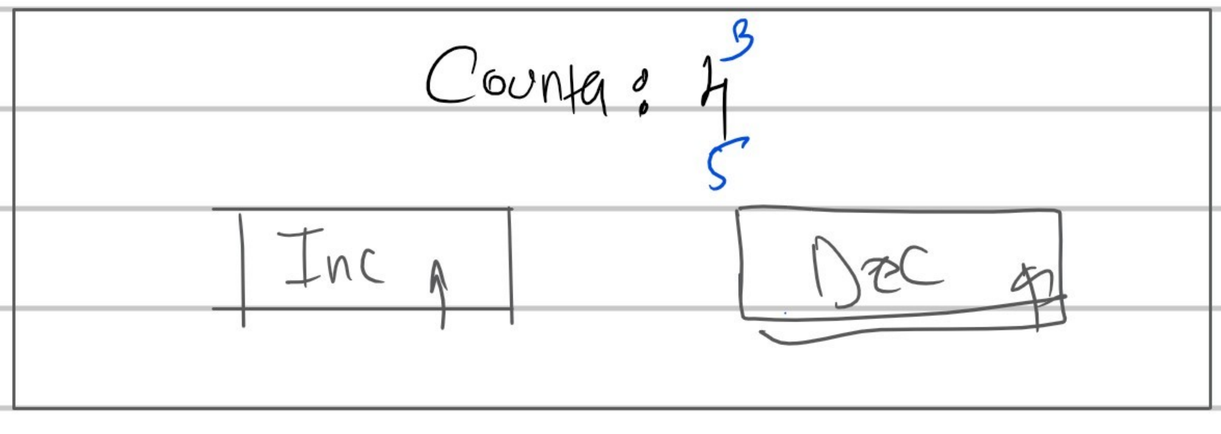
```
const [variable, setVariable] = useState(defaultValue)
```

- variable → state
- setVariable → function to update the state
- useState → hook for creating state in functional component
- defaultValue → default value for state

```
const [age, setAge] = useState(0);
const [fruit, setFruit] = useState('Apple');
const [colors, setColors] = useState([]);
const [address, setAddress] = useState({})
```

```
({ city: 'Mumbai',
   pin: 400000 }
);
```

functional



# *DOM*

DOM → Document Object Model

→ represents the UI of the application.

*Everytime application state gets updated; DOM is updated

↳ represent the change

→ DOM is represented as a tree data structure.

↳ re-rendering of the pages & its child components are time consuming operations

## *Virtual DOM*

→ When new elements are added to the UI, a Virtual DOM which is represented like a tree is created.

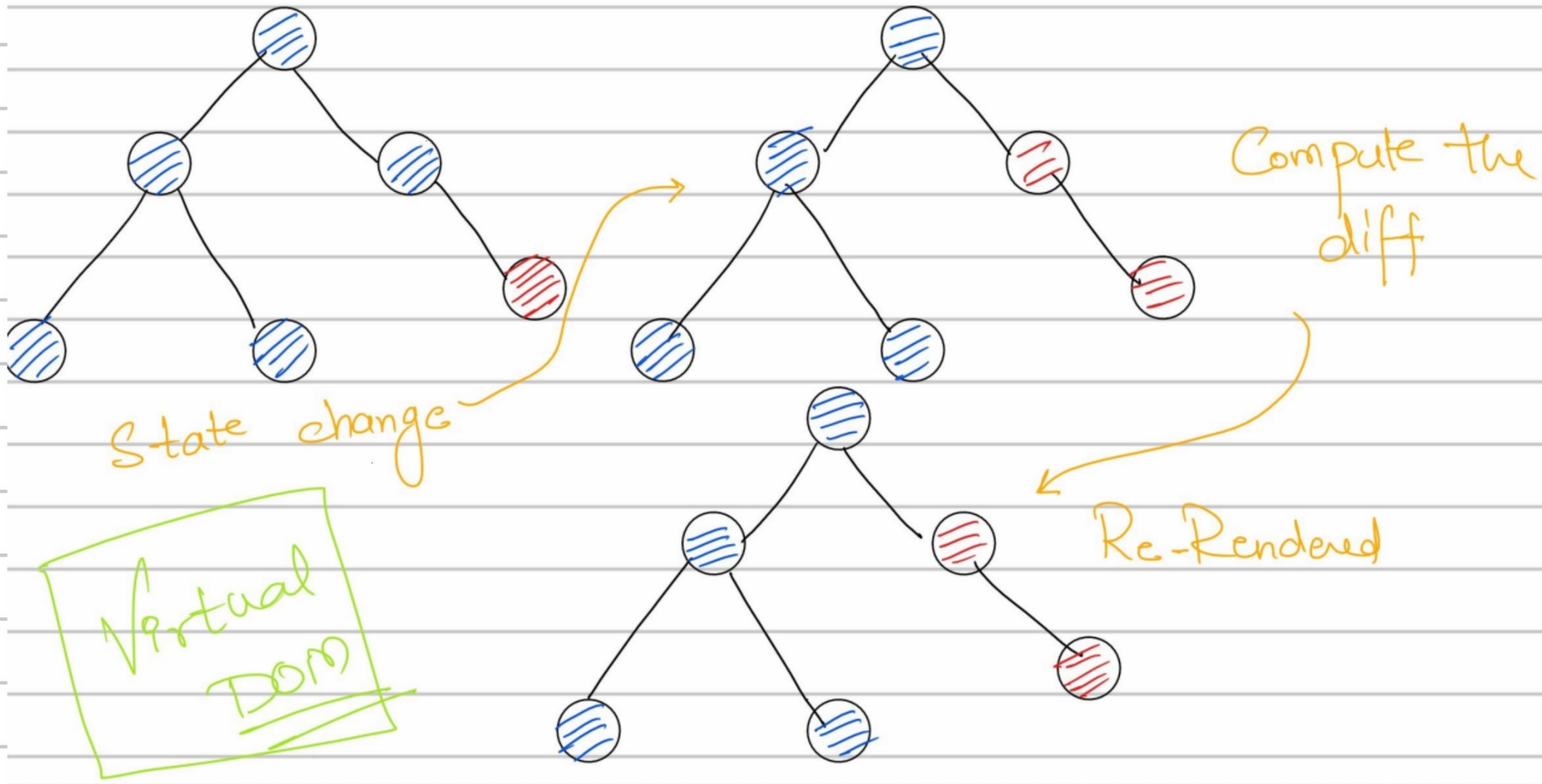
Each elements is a node in a tree.

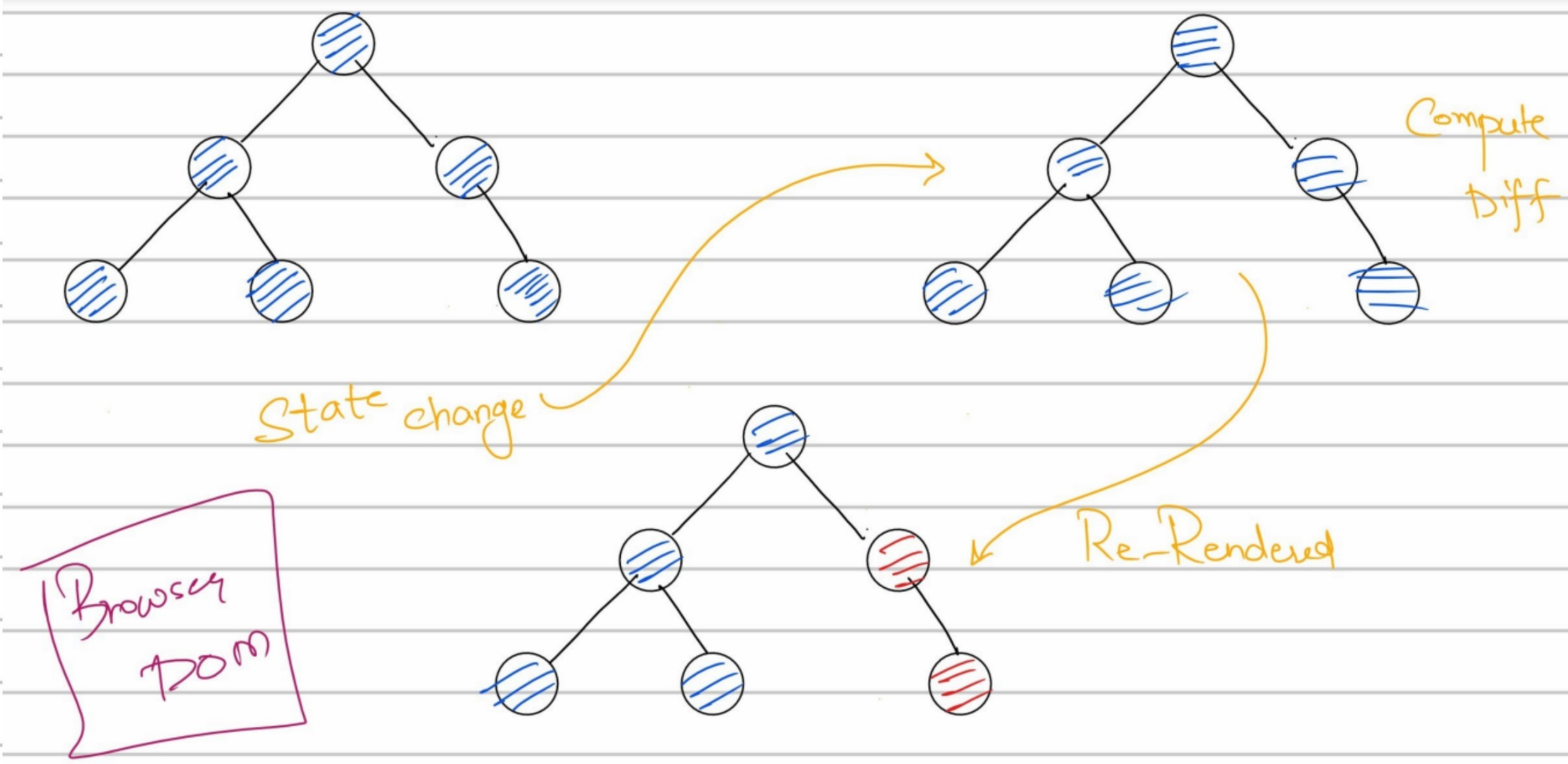
If state of these element changes; a new Virtual DOM tree is created.

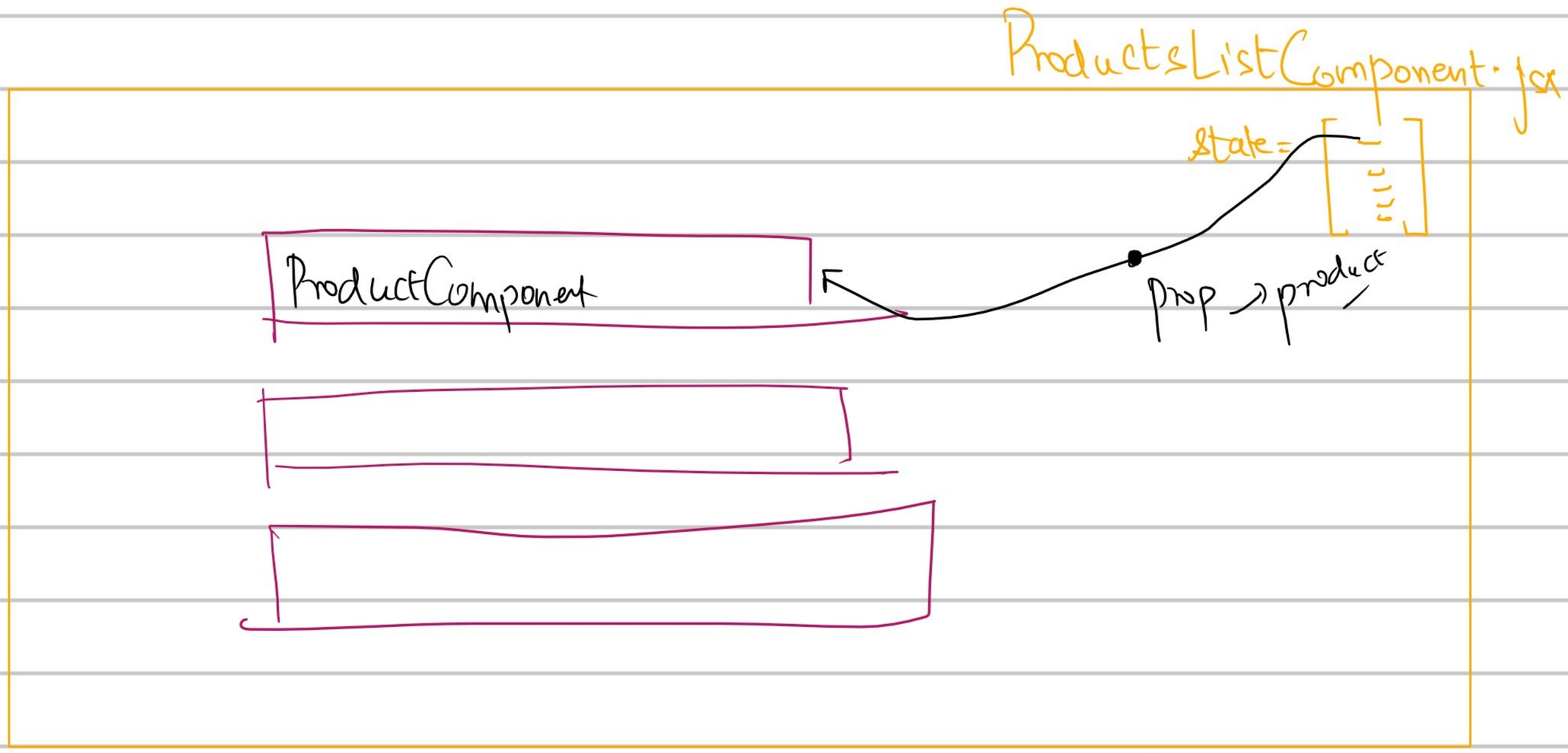
The tree is compared with previous virtual DOM Tree

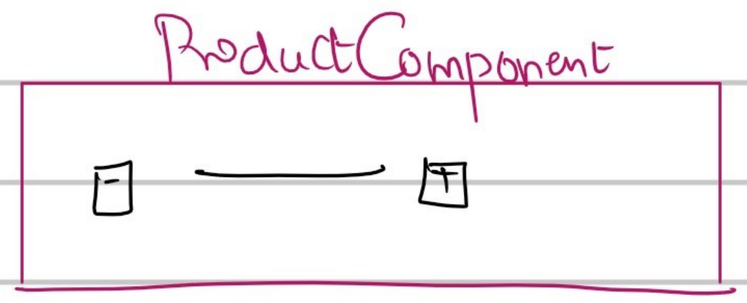
↳ Diffed

Diffing

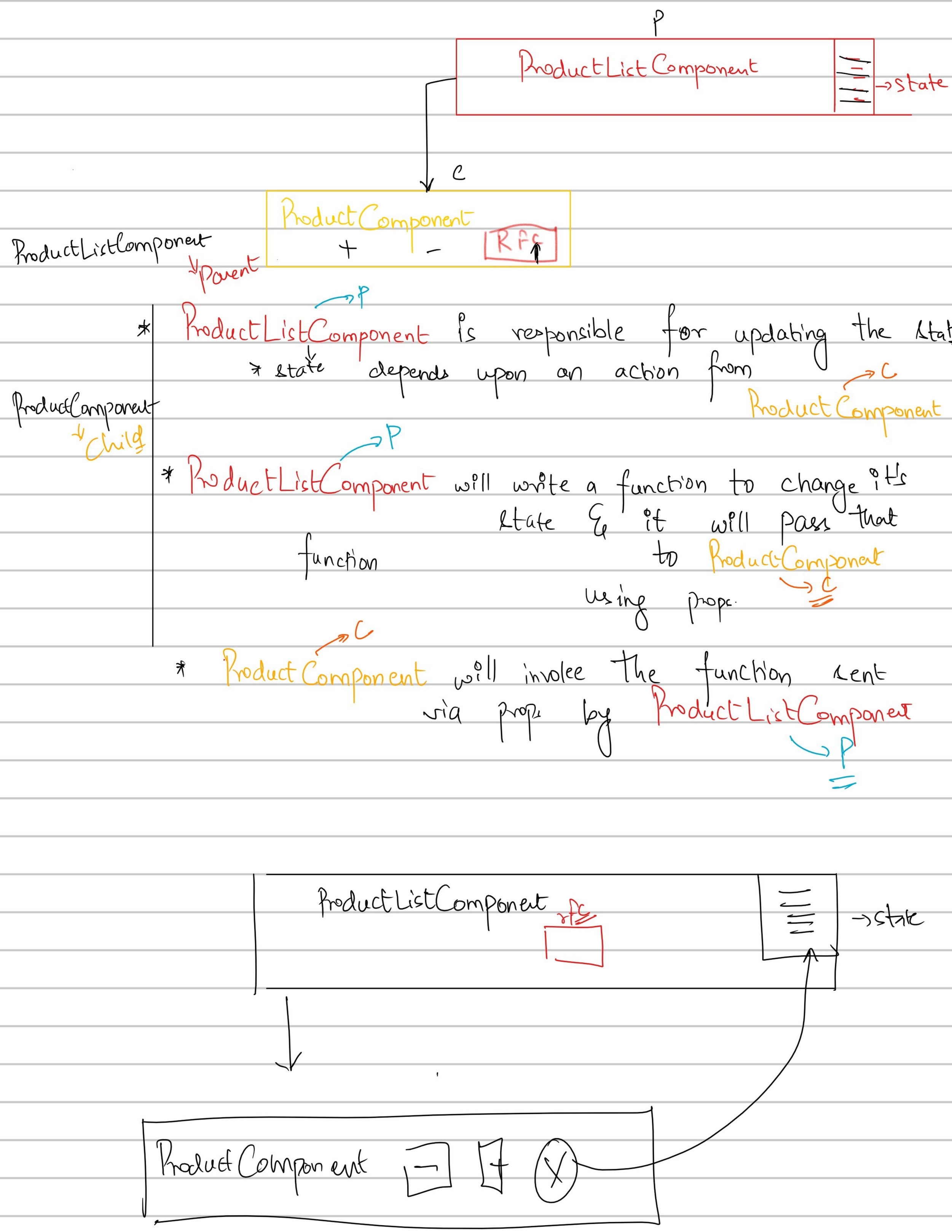

Compute the diff
State change
Re-Rendered
Virtual DOM


Compute Diff
State change
Re-Rendered
Browser DOM




"STATE of a COMPONENT should only be modified by the SAME COMPONENT"


P
ProductList Component
→state
C
Product Component
+
−
RFC
ProductListComponent
↓Parent
ProductComponent
↓Child
* ProductListComponent is responsible for updating the state
* state depends upon an action from Product Component
* ProductListComponent will write a function to change it's state & it will pass that function to ProductComponent using prop.
* Product Component will invoke the function sent via prop by ProductListComponent
ProductListComponent
RFC
→state
Product Component
−
+
X