Lecture 6

* Client->ISP->DNS(like phone book that stores IP address) and vice versa
* NsLookup.io : to view the IP address of the website
* Submarinecablemap.com: one can see all the cables that connect u to the internet

Lecture 7

* Data recd from server consists of:
* HTML : like raw materials of house..images, buttons,links
* CSS: colour of wall..how website will look like background of website
* Javascript: code that allows website to have functionality…like lightbulb

Lecture 11

* HTML decides content and structure of website
* HTML: Hypertext : pieces of text which to other documents in website
* Markup: bold, italic…done thru HTML tags like heading, paragraph

Lecture 13

* Codepen.io
* <h1>….<h6> closing tags
* <br> break line
* <!—iojfiorejijifr-->
* <https://web.archive.org/> : History of website

Lecture 16

* <!DOCTYPE html> : tells browser that code is written in HTML version
* Open VS Editor…create a new folder..add a new html file..then write ! and press enter…it generates the whole code
* To crop image in circle: <http://crop-circle.imageonline.co/>

Lecture 19

* Anchor tag: <a href=”https://...”>Hello</a>
* a=HTML element, href= HTML attribute, “https://...=link destination, hello=link text
* HTML tip: just type “!” for HTML boiler plate

Publish your website

* Pages->Branch->Main->Save

CSS

* Lots of code to write simple things as tables
* Syntax errors were common..difficult to debug..nested tables were messier
* CSS doesn’t do anything
* Only purpose is to style markup language like HTML,Xmletc

Inline CSS

To know the available color using style..google “change background color using cssmdn”

* Colorhunt.co
* Browser default css
* Everythong on webpages are boxes-chrome plugin pesticide---click on chrome 3 dots-setting-extensions-pesticide for chrome-details-allow access to file url-enable
* Pin extension on chrome
* Style before head
* Devdocs.io
* /\* width: 100px;  \*/
* width: 30%;

CSS Rules

* **Inline CSS: Requires the style attribute placed inside an HTML element.** **Internal CSS: Requires the <style> element placed inside the head section of an HTML file**.An external style sheet is **a separate CSS file that can be accessed by creating a link within the head section of the webpage**
* Selector{propery:value;}
* Selector=who, property=what, value=how
* Emojipedia
* **Class selector instead of tag selector for flexibility**
* **Class selector is shown below:-**
* <imgclass="bacon"src="https://emojipedia-us.s3.amazonaws.com/thumbs/240/apple/118/bacon\_1f953.png"alt="bacon-img">
* <imgclass="broccoli"src="https://emojipedia-us.s3.dualstack.us-west-1.amazonaws.com/thumbs/240/apple/325/broccoli\_1f966.png"alt="broccoli-img"

The style.css sheet

.bacon

{

    background-color: aqua;

}

.broccoli

{

    background-color: blueviolet;

}

ID selector

  <h1id="heading">I Love Bacon</h1>

#heading

{

    color: blue;

}

Difference between class selector and id selector

|  |  |
| --- | --- |
| **Class Selector** | **ID Selector** |
| Multiple instance | Particular ID can only be used once per page..eg id=”heading” can only be used once in the page |
| Element can have more than 1 class(refer \*) | Element can have only one ID |
| Usage: when u want to apply the same style to a group of related items | use the id to apply a specific style to a single element on your web page |
| Used frequently | Used sparingly |

* <imgclass="broccoli circular"src="https://emojipedia-us.s3.dualstack.us-west-1.amazonaws.com/thumbs/240/apple/325/broccoli\_1f966.png"alt="broccoli-img">
* .circular
* {
* border-radius: 100%;
* }

This is not possible for ID say  <h1id="heading big">I Love Bacon</h1>

HERE 2 Ids i.e. heading and big are not possible

Similarity:-

* Both are used to identify HTML elements that you want to style and CSS doesn’t care what you choose
* They are more specific than tag selectors and any of these styles will override any tag selector that are applied to the same element.
* Notice that some of these tags have predefined CSS styles that are applied by the browsers

**PSEUDO CLASS: see at this link ->**[**https://developer.mozilla.org/en-US/docs/Web/CSS/Reference--HTML**](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference--HTML) **ELEMENTS CAN HAVE DIFF STATES**

**:active : left get CSS change based on the state like hovering or not hovering**

Common is hover class..we can get to change the color of image on mouse hover

img:hover

{

    background-color: gold;

}

Favicon: fav icon:

Favicon.cc

Click on pesticide…then using contro; button and mouse hover…one can see the properties displayed below

* Div: special HTML element that allows us to divide the contemt on the website. Div doesn’t do anything unless css is used with it..content division element allows to split into diff containers or boxes
* Span and div are both **generic HTML elements that group together related parts of a web page**. However, they serve different functions. A div element is used for block-level organization and styling of page elements, whereas a span element is used for inline organization and styling
* BOX MODEL OF CSS
* CSS DISPLAY PROPERTY
* BLOCK
* INLINE
* INLINE-BLOCK
* NONE

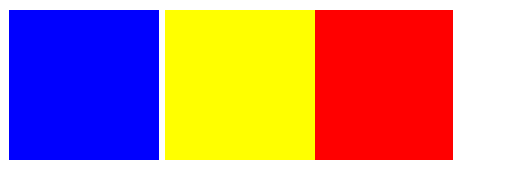
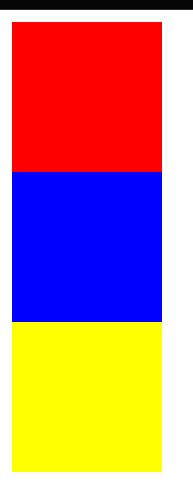
Common Inline Elements

* Spans
* Images
* Anchors
* The above 3 do not take up the entire block space/ screen space and allows adjacent placement of items..hence called inline elements..problem is we cant change the width..it is fixed..in the example below we were able to chnge the width of {p} (though doesn’t allow any other element to sit on the same line)but we were not able to change the width of span{}…paragraph is a block element
* 
* One solution is that we can use the inline property for block element like paragraph to let other elements sit beside it but now we lost the ability to change its width
* Now we can also use block in inline element to change its width but then it doesn’t allow other elements to sit on the same line as shown in image below
* so to give the best of both worlds…inline block elements can be usedthe use of none property is shown below
* visibility:hidden can also be used..but then it reserves the space as shown in image below
* CSS STATIC AND RELATIVE POSITIONING
* 1 content is everything…ie. Height and width are determined by content
* 2 order comes from code
* 3 children sit on top of parents
* Eg<div>
* <h1> a programmer</h1>
* </div>….color of h1 will be on top..say if div is of red colour

Position of layout: <https://leannezhang.medium.com/difference-between-css-position-absolute-versus-relative-35f064384c6>

A better tutorial to understand the positioning concept: https://developer.mozilla.org/en-US/docs/Web/CSS/position

* Static
* Relative
* Absolute
* Fixed



.red

{

height:100px;

width:100px;

background-color:red;

position:relative;

display:inline-block;

left:200px;

}

.blue

{

height:100px;

width:100px;

background-color:blue;

display:inline-block;

right:100px;

position:relative;

}

.yellow

{

height:100px;

width:100px;

background-color:yellow;

position:relative;

display:inline-block;

right:100px;

}

Same thing can be achieved by absolute positioning:

.red

{

height:100px;

width:100px;

background-color:red;

position:absolute;

display:inline-block;

left:200px;

}

.blue

{

height:100px;

width:100px;

background-color:blue;

display:inline-block;

position:absolute;

}

.yellow

{

height:100px;

width:100px;

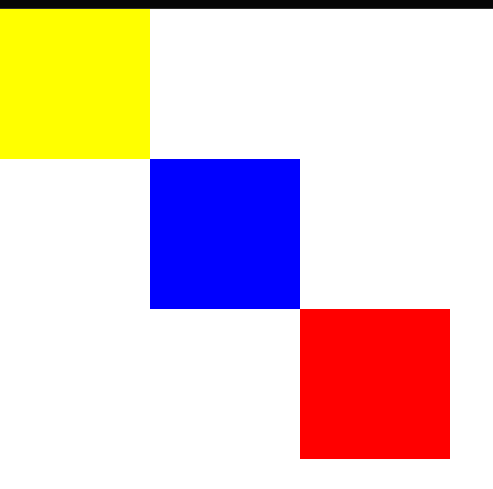
background-color:yellow;

position:absolute;

display:inline-block;

left:100px;

}

body{

margin: 0;

}

.red

{

height:100px;

width:100px;

background-color:red;

position:absolute;

display:inline-block;

top:200px;

left:200px;

}

.blue

{

height:100px;

width:100px;

background-color:blue;

display:inline-block;

left:100px;

top:100px;

position:absolute;

}

.yellow

{

height:100px;

width:100px;

background-color:yellow;

position:absolute;

display:inline-block;

}

Done by absolute positioning



body{

margin: 0;

}

.container

{

position:relative;

width:300px;

height:300px;

background-color:grey;

}

.red

{

height:100px;

width:100px;

background-color:red;

position:absolute;

right:0;

}

.blue

{

height:100px;

width:100px;

background-color:blue;

display:inline-block;

top:0;

left:100px;

position:absolute;

}

.yellow

{

height:100px;

width:100px;

background-color:yellow;

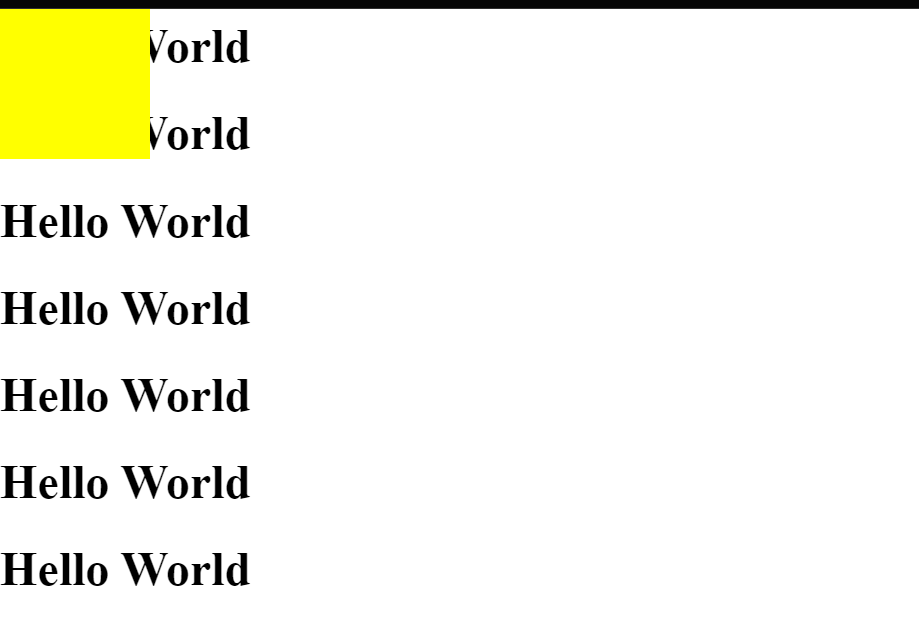
position:absolute;

display:inline-block;

top:0;

}

Using position:fixed would keep element fixed on screen while scrolling also



.yellow

{

height:100px;

width:100px;

background-color:yellow;

position:fixed;

display:inline-block;

top:0;

}

Centering elements with CSS

text-align: center;

works for full width block elements..inline block

say if we change the code to h1

{

    margin-top: 0;

    width: 10%;

}

Then ouput isn’t what we would expect

So in this case all we need to do is:-

h1

{

    /\*margin-top: 0;\*/

    width: 10%;

    margin:0auto0auto;

}

Font Styling

* Serif and Sans Serif
* 5 font family:
* Cursive:not used freq..low readability + doesn’t look gr8..handwritten style
* Fantasy: avatar style
* Inherit
* Monospace: used for coding..same space b/w letters
* Sans-serif
* Serif
* Font-family:serif
* Font-family:verdana, sans-serif
* The above means that if verdana not in users system then sans-serif shall be used
* Better use wb safe fonts that most OS will be able to render :CSS Web safe fonts
* Go to “CSS Font stack” : It displays the % chance that OS will render<https://www.cssfontstack.com/>
* Say “helvetika” font..particularly for Mac..then if Windows is there then fall back is required
* To be more picky..ie same font shud be displayed in every system then FONT EMBEDDING can be used
* <https://fonts.google.com/> : here whole bunch of font family is present
* So go to this website…add them to your selection..include their link in index.html…add in cSS as well and Voila !

97: Understanding typography and how to choose a font

Of all fonts..2 large families

* Serif and sans-serf feet at the end..inspired by marble carvings..letterhead for legal company. It has sub-family as:-

Old Style, Transitional, Modern and Slab-Serif..one can tell which is the oldest family by looking at the thickest and thinnest part of the letter..this width differenece gets more and more in the newer family

[**https://www.flaticon.com/**](https://www.flaticon.com/)plenty of stock images

[**https://giphy.com/**](https://giphy.com/)

**CSS Sizing**

 font-size: 90px;

Doing the above is fine but once the user decides to increase or decrease the size then it remains unaltered. Preferences->font size-very large/very small..So in this case % is used as a fix for dynamic sizing.

Font-size:100%; 16px hence for 90 px…font-size should be 562.5% (90\*100 / 16)

* Em is also used(used in text for italic)..here used differently
* 1em=16px=100%
* Now suppose font size in parent is set to 100% and then css..h1 tag also font size set to 100% then the font size increases gigantically..like 200%. To avoid this instead of em..rem is used as a unit which means to ignore the previous styling and focus on current

Margin:auto; to keep alignmrntcenter

float: left;

clear: left;

below is opposite of above

float is used for wrapping text around certain element..dont use for positioning

 line-height: 2;

to stylize button..refer this s

ite: <https://css3buttongenerator.com/>

To add a line break to your HTML code, you use the **<br> tag**

**Website Solution**

margin: 100px0;

* 100px: top bottom….0px: left right
* .intro
* {
* width:30%;
* margin:auto;
* }
* The obove code places the intro for 30% screen and by mentioning it auto, it keeps it in centre.
* To print “…” instead of----- to bifurcate sections…
* hr{
* border: dotted#EAF6F66px;
* border-bottom: none;
* width: 4%;
* margin: 100pxauto;
* }
* Add <br> to add newline
* margin: 40pxauto60px;
* The above means 40px top…left nd right is auto and 60 px for bottom
* text-decoration: none;
* this would remove underline from anchor tag

BOOTSTRAP

* a front-end library
* front end lang: HTM:, CSS, JAvascript
* Backend: .NET, Ruby, PHP, NodeJS, Java, SQL
* Bootsrap: bunch of code that is reusable
* Developed by twitter to bring consistency across websites…now open-source
* It is RESPONSIVE: responds to the view-port..ie mobile, desktop, etc…adaptive layout
* Access to PRE-STYLED ELEMENTS
* Use **codeply** to write code
* <button class="btnbtn-primary">Hello World</button>
* <button class="btnbtn-dark">Hello World</button>
* <button class="btnbtn-outline-dark">Hello World</button>
* Predefined classes that have already been styled

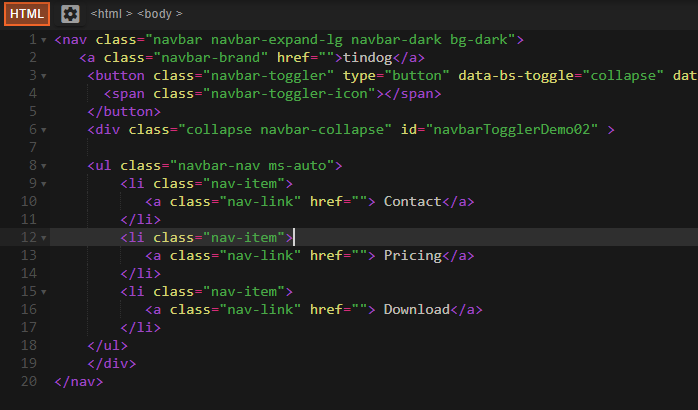
Head over to getbootstrap.com..whole page is dedicated to install…copy paste cdn link from there.. "https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">

* Downloads bootsrap file if user doesn’t have on browser
* Intro about CDN—if server is in US..and I access website from UK then,..it goes thru undersea cables..increaeslatency..if that is the only location where page is present..it can create a delay of more than few millisec or more..so here CDN comes..instead of hosting the website at only single loc..we have a bunch of loc where it is stored and can accessd..it looks for loc that is of the shortest dist to get the website delivered..bootstrap have loaded their bootstarpcss on maxcdn which has a whole no of routing points across the world. Relatively quickly…if browser doesn’t have bootsrap then goes to CDn..generally browsers have it in cache…this was one way
* Other way to use bootstrap is to copy paste their starter template. Its got all the declaration,bootrapcssetc..also has javascript and jquery (req for few componenets of bootstrap to work like dropdown menu)..so one can download the starter template..everthing in the index.html to work
* Third way is to download the CSS and javascript source code to include in website..so instead of pointing URl on internet one can point it to a relative URL that will be included with your website..the downside ois that this negates any caching that the browser might have already..so it ha sto download all the files which can increase latency

65. WIREFRAMING

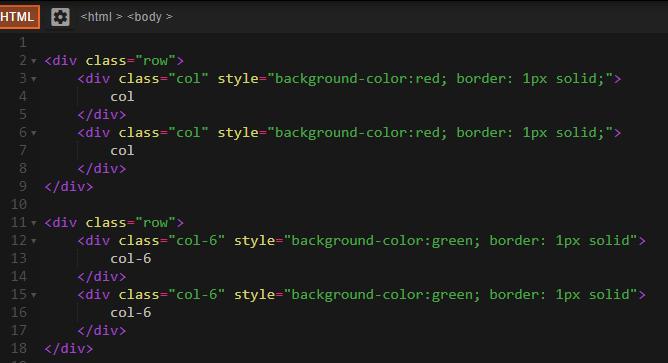
* Low fidelity representation of your website..done with pencil etc..before writing any code..sharing it with people..settle with design before going to implement…meant to be very simple
* Opposite is MOCKUP-high fidelity representation..like going to future to show the screens as to how it will work in future..u end up choosing font,icon,coloretc which bogs u down..so classic workflow is to look at other ppls websites (**https://www.awwwards.com/)**and then see the UI patterns as well of others([**https://ui-patterns.com/**](https://ui-patterns.com/)**)**
* next step is to create a simple sketched wireframe for planning the layout..then create a mockup(photoshop/illustrator)to creataphoto realistic versions of your future website. Then to add the extra mile, one can also add the PROTOTYPe which is basically the animated version of your website.
* A really good resource for looking for inspiration on all of these fronts is a website called DRIBBLE, which is a place where designers put up their portfolios.[**https://dribbble.com/**](https://dribbble.com/)**..**Cool thing about dribble is that ne can even search dribble by colour.
* To create a wirefrsme..use just pencil and paper…a website called [**https://sneakpeekit.com/**](https://sneakpeekit.com/)which has printout for mobile, browser, tablet etc. in order to wireframe your website. A bit more advanced, industry specific tool is [**https://balsamiq.com/**](https://balsamiq.com/)

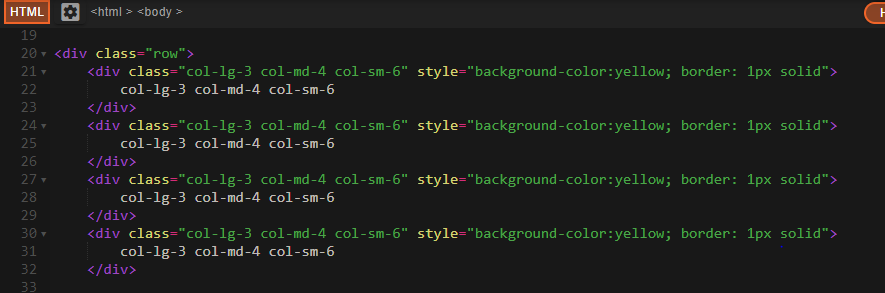
**THE BOOTSTRAP NAVBAR**

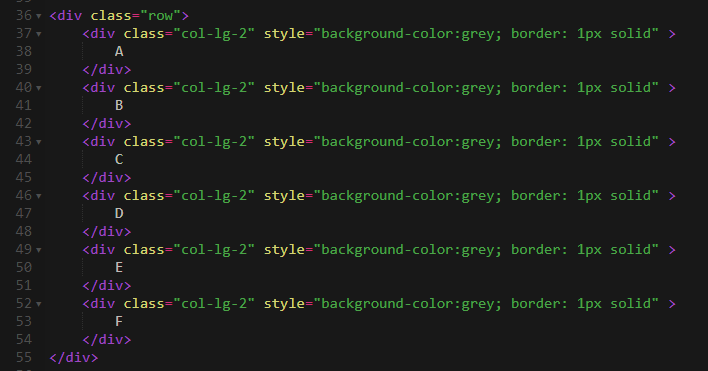
* Bootsrap is a library that contains reusable bits of code
* This lirary is made by third party…originally by Twitter but since then many contributors are there
* Bootsrap-components-navbar(<https://getbootstrap.com/docs/5.3/components/navbar/>)
* Bootstrap-utility-spacing: one see documentation on margin and spacing
* Before using codeply, ensure that HTML and BOOtstrap both are added
* 
* Navbar-expand-lg is written to align horizontally
* Navbar-dark: dark theme selected as bar
* Bg-dark: dark colour selected as background. Bg-warning: yellow, bg-danger:redetc
* Ms-auto: written to add space between brand name and menu items. M:margin and s:space so it takes maximum possible space b/w brand and menu items
* Toggle is used for responsive layout design: choose the size of the navigation bar

**WHAT WE’LL MAKE: TINDOG**

**BOOTSTRAP GRID SYSTEM**







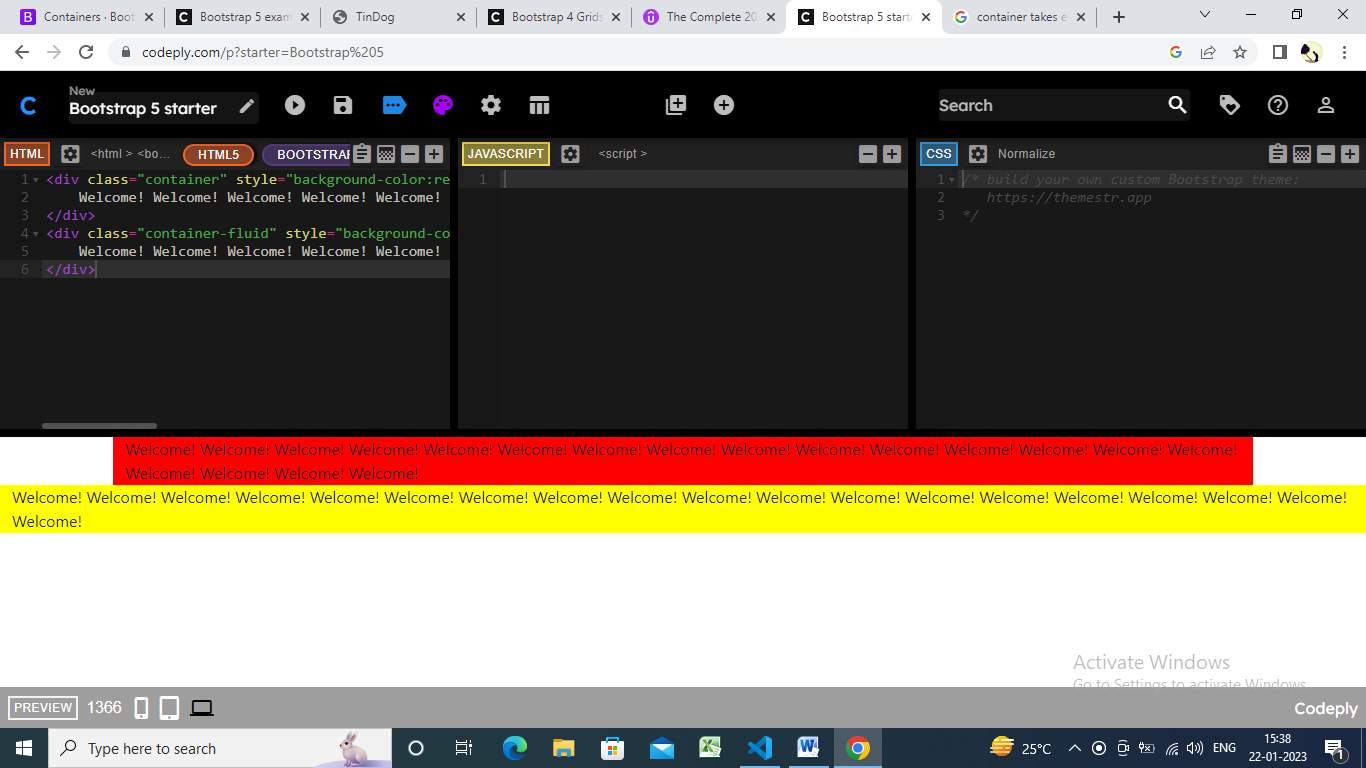
* The whole screen is divided in 12 units..Herelg: desktop, bg:tablet, sm: mobile…lets say we wish to see 4 icon n desktop mode then 12/4=3..so we write col-lg-3

Adding Grid Layout to website

* To Target HTML element we target name of element say body{}, to target class we add . in front of class name, for ID we add #
* <navclass="navbarnavbar-expand-lgnavbar-dark bg-dark">
* To make thenavbar header have the same red color as the title, we remove bg-dark">
* To embed the fonts: Ubuntu and Montserrat
* Go <https://fonts.google.com/>

Containers

* Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport.
* Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**.
* Auto adaptive as per screen..this leaves space around.Our default .container class is a responsive, fixed-width container, meaning its max-width changes at each breakpoint.
* Next type is fluid container…they take entire screen…Use .container-fluid for a full width container, spanning the entire width of the viewport.



Both the tags ([<div>](https://www.geeksforgeeks.org/div-tag-html/)and [<section>](https://www.geeksforgeeks.org/html-section-tag/#:~:text=Section%20tag%20defines%20the%20section,other%20section%20of%20documents%20needed.)) are used in the webpage, <section> tag means that the content inside relates to a single theme, and <div> tag is used as a block part of the webpage and don’t convey any particular meaning.

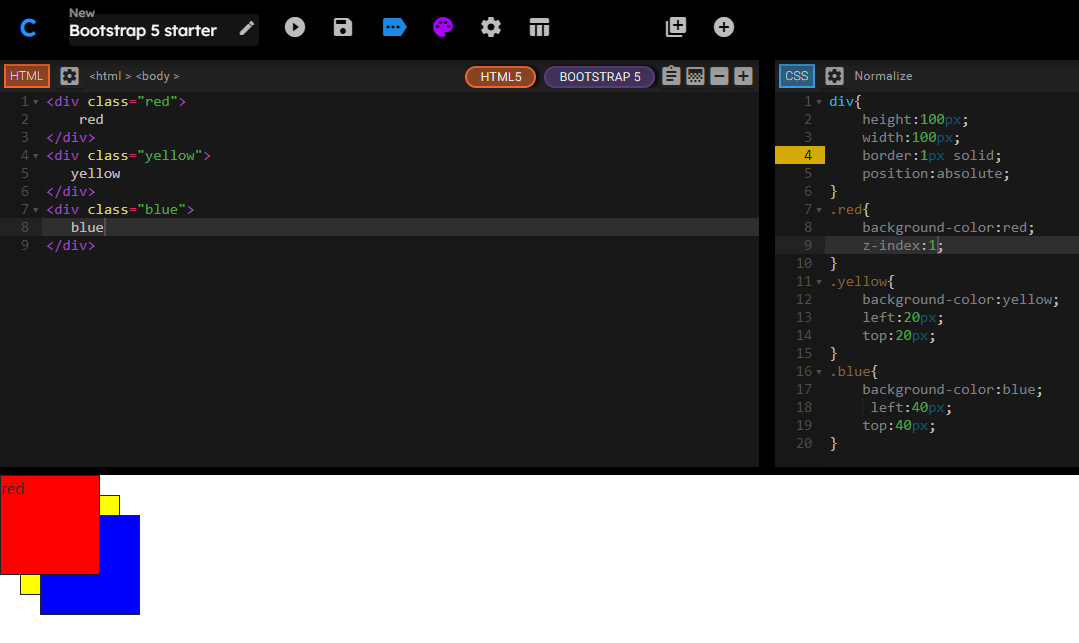
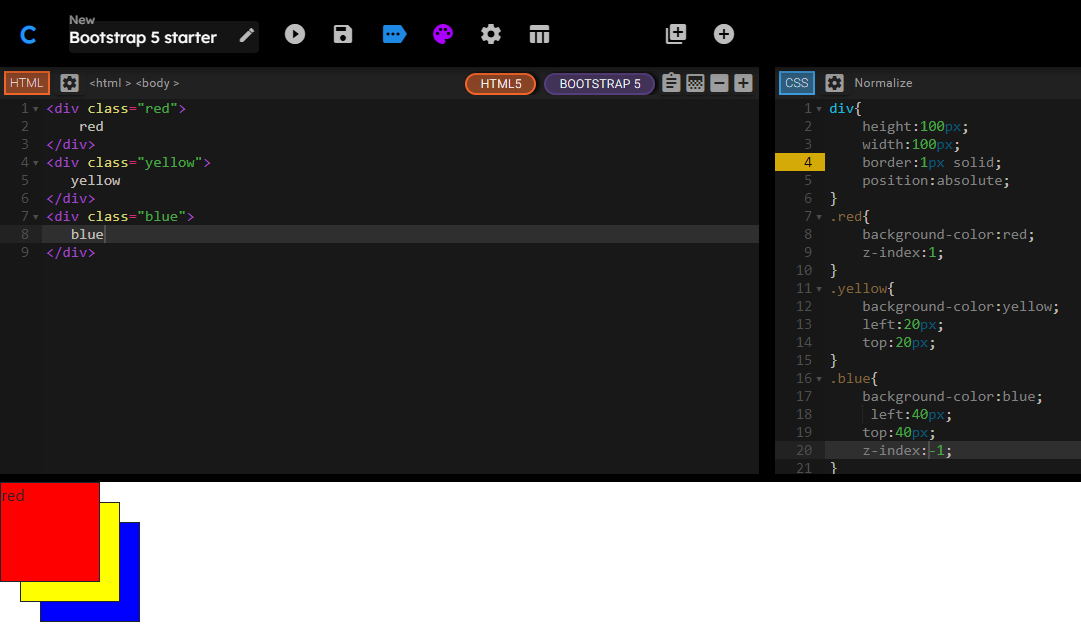
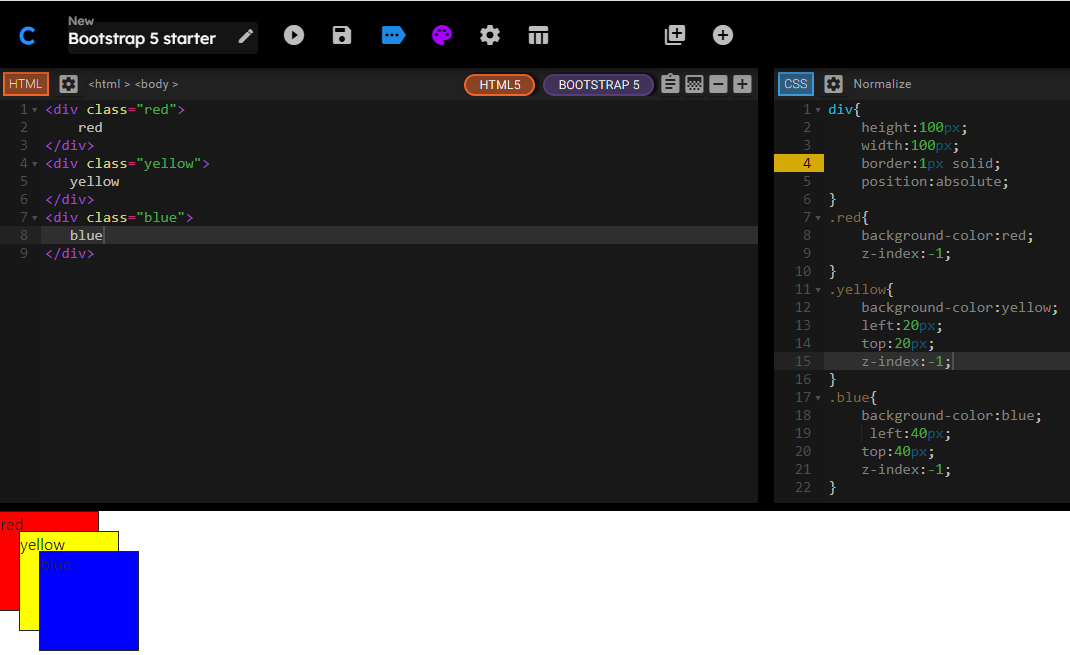
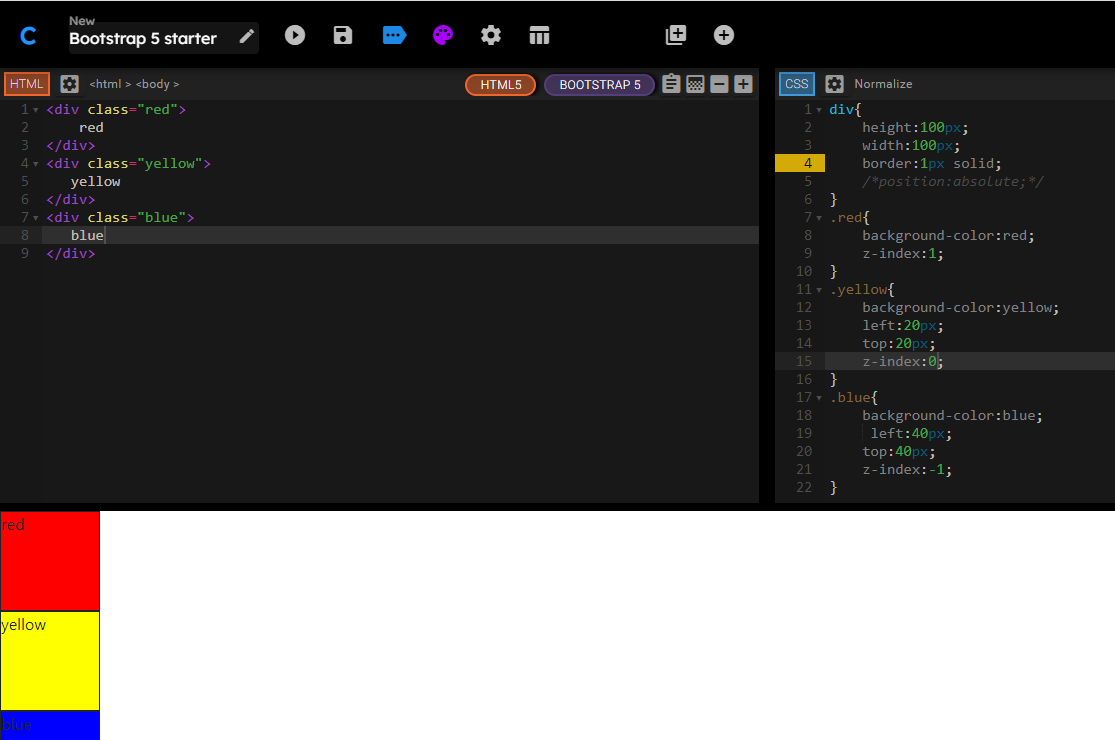
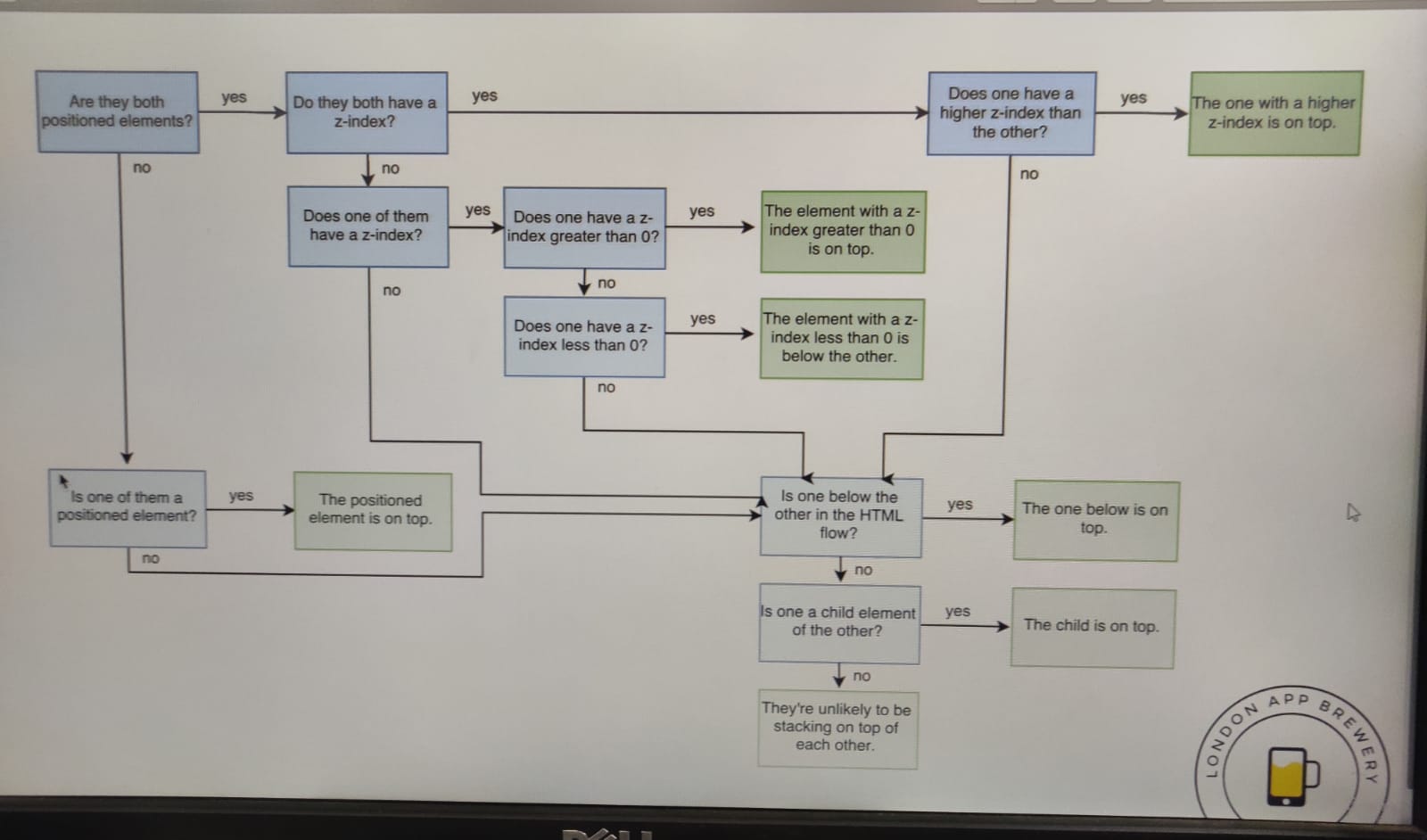
**BOOTSTRAP BUTTONS AND FONT AWESOME**

* Icons on button come from website fontawesome (<https://fontawesome.com/>)
* Use CDN
* <script src="https://kit.fontawesome.com/8d02909fe6.js" crossorigin="anonymous"></script>
* And in button add…
* <iclass="fa-brands fa-apple"
* The code should actually look like this:-
* <buttontype="button"class="btnbtn-dark btn-lg"><iclass="fa-brands fa-apple"></i> Download</button>
* **STYLING OUR WEBSITE**
* To rotate image:-
* .title-image
* {
* width: 60%;
* transform: rotate(25deg);
* }
* **BOOTSTRAP CAROUSEL**
* Line height:Set the line height for different <div> elements:
* .testimonials-image
* {
* width: 10%;
* border-radius: 100%;
* margin: 20px;
* }
* Width:10% makes it smaller, border-radius:100% makes it round, margin-20% gives space

BOOTSTRAP CARDS

* <https://getbootstrap.com/docs/5.3/examples/pricing/>
* <https://bootsnipp.com/> (SEE login screen)

CSS Z-INDEX AND STACKING ORDER

* 
* 
* Z-index default is 0
* 
* Stacking order and z-index is complicated…z-index only **works if elements are positioned..**if not then elements doesn’t get stacked..like in img below
* 
* Z-index doesn’t work in static positioned elements or if no position mentioned coz static is the default position
* 

Code to rotate image

.title-image

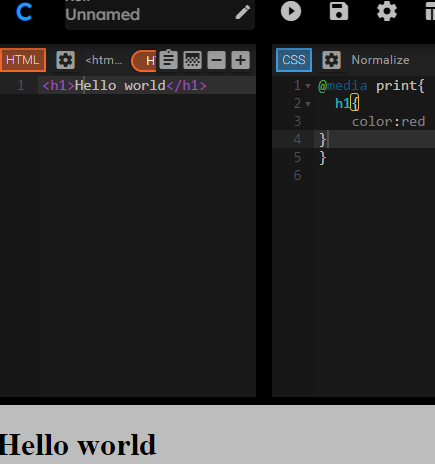
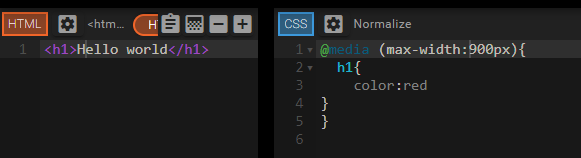
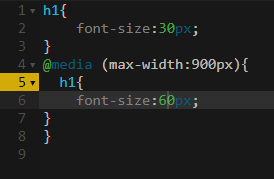
{

    width: 60%;

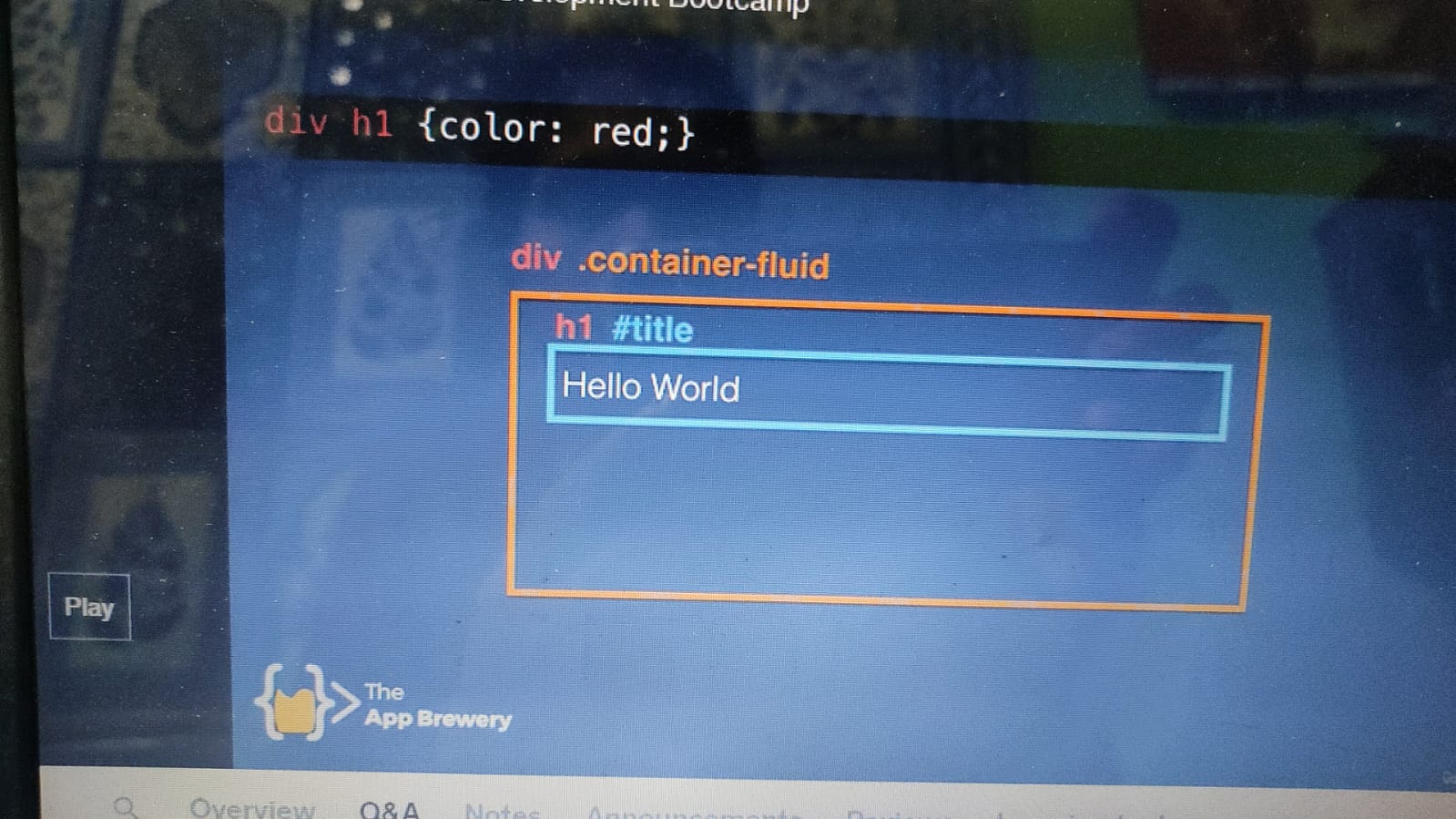
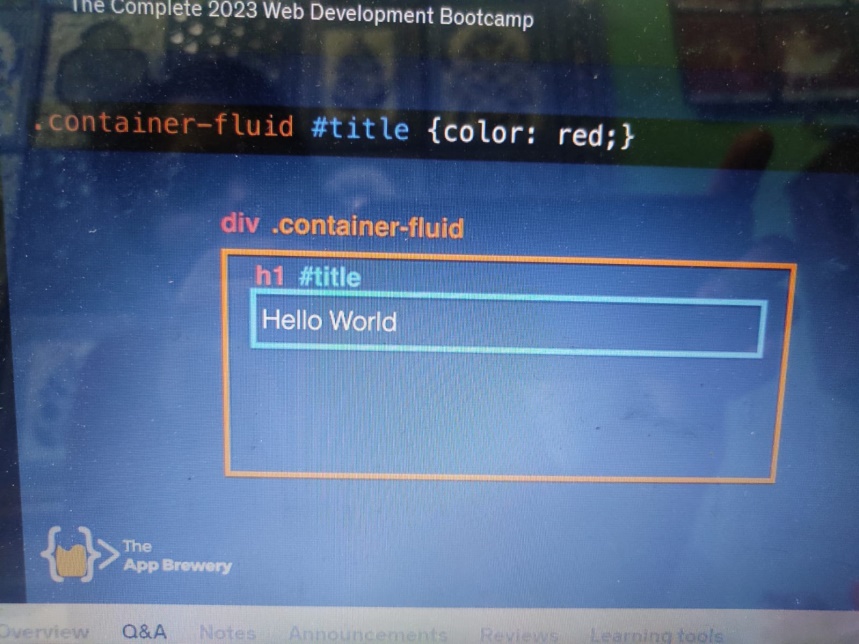
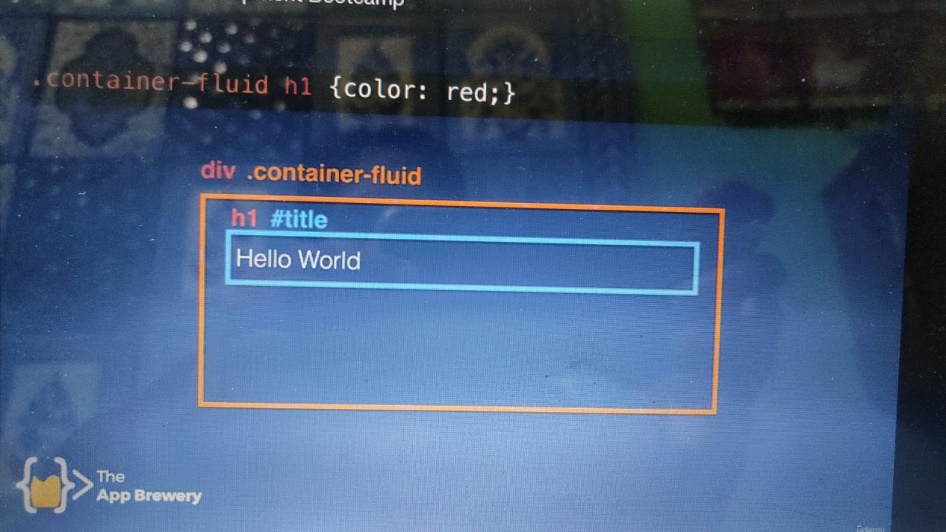
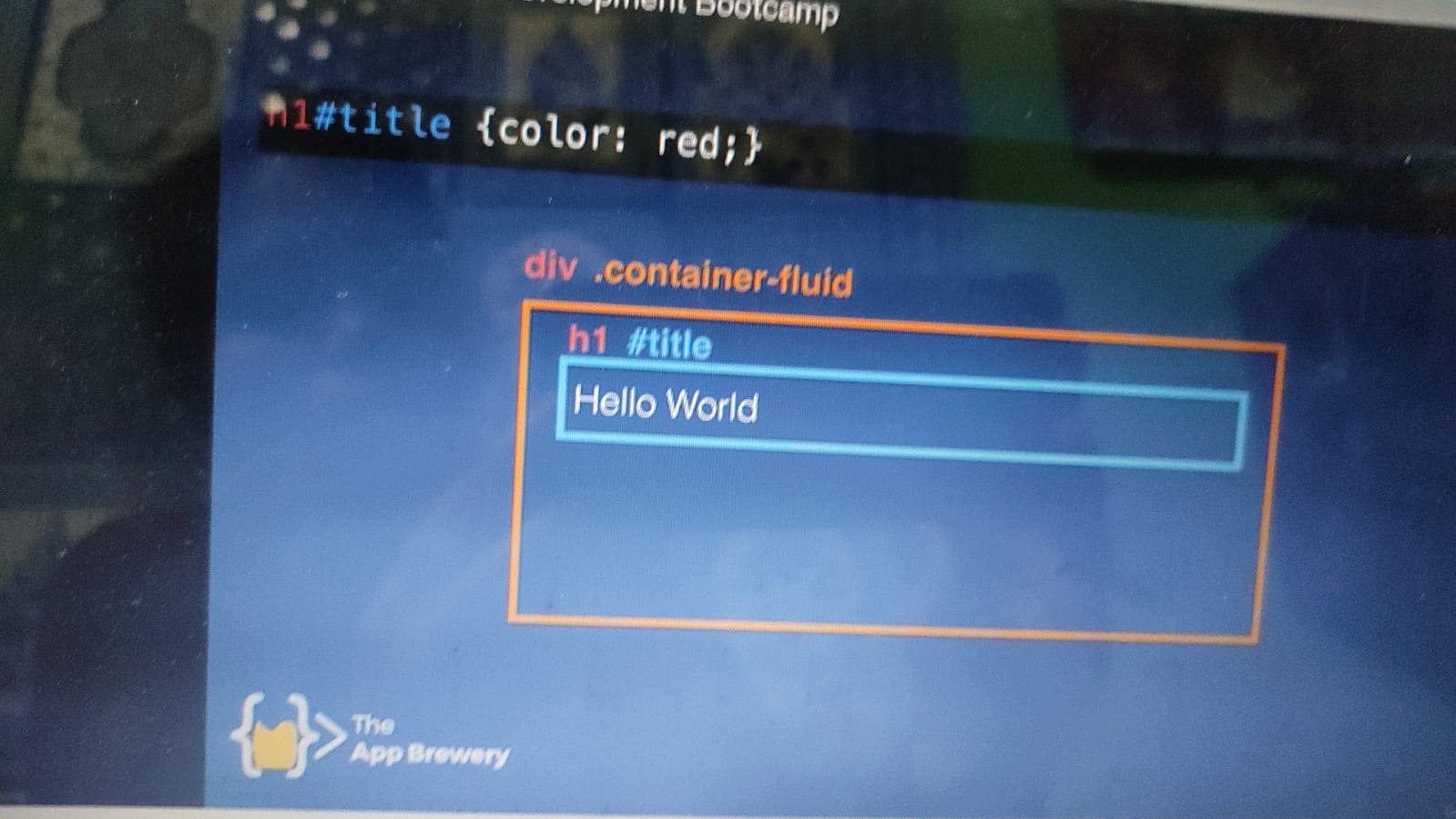
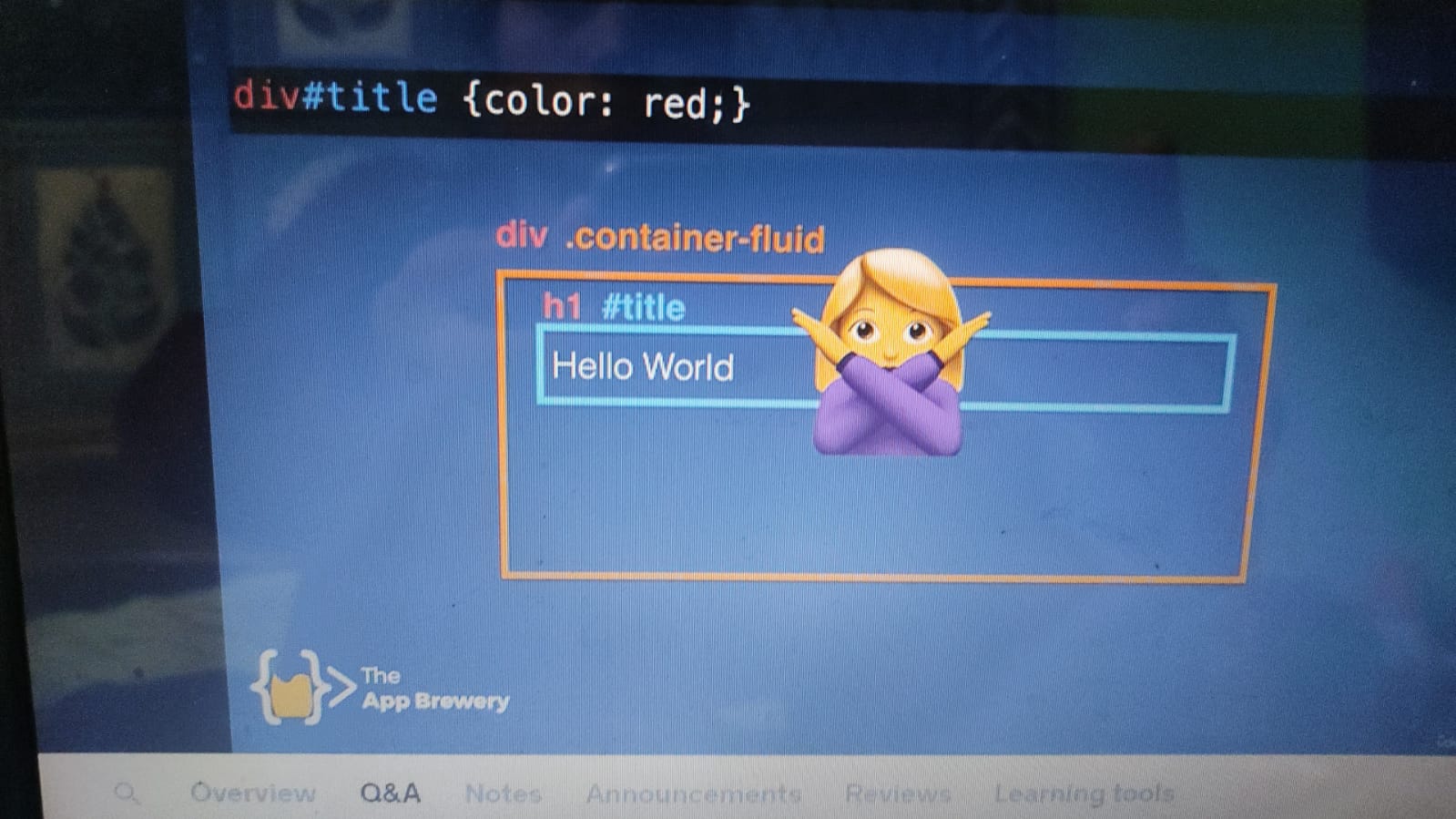
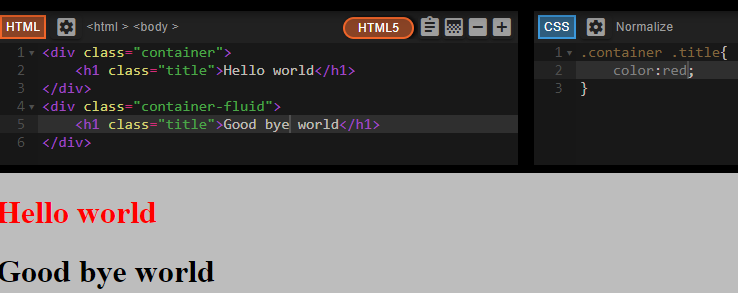
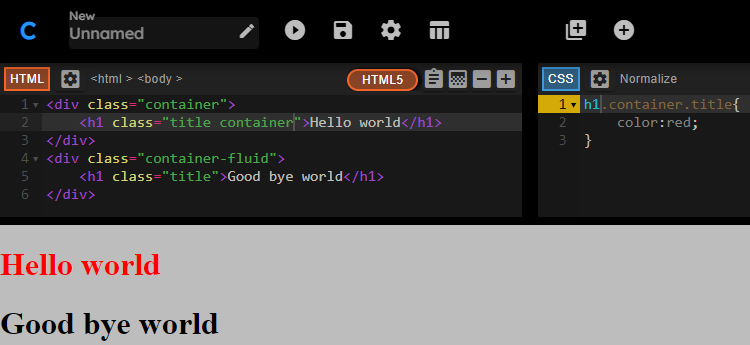
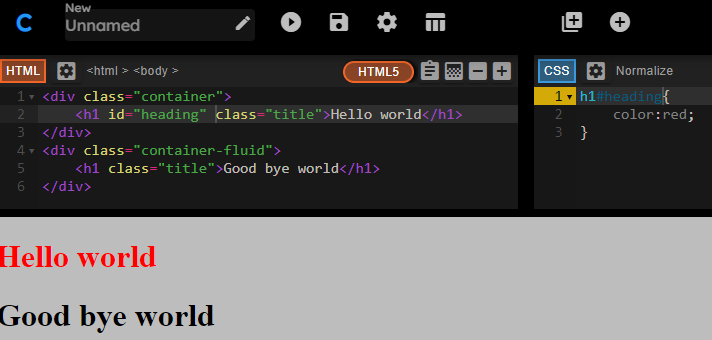
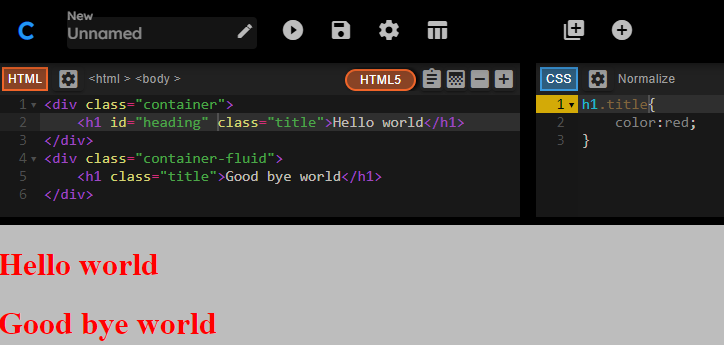
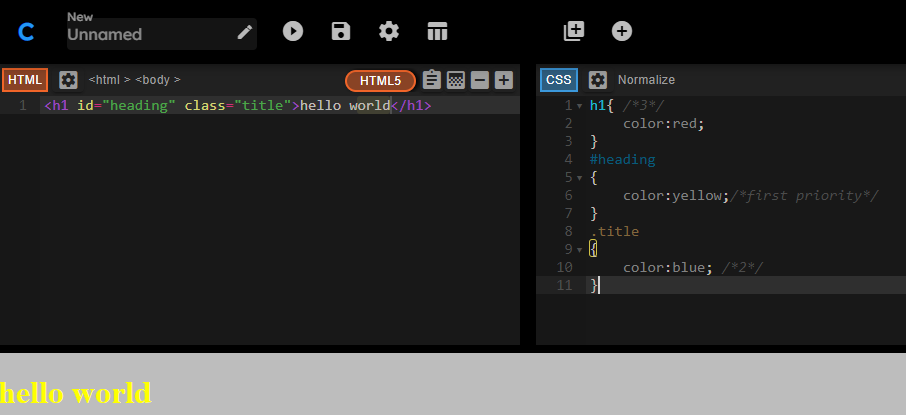
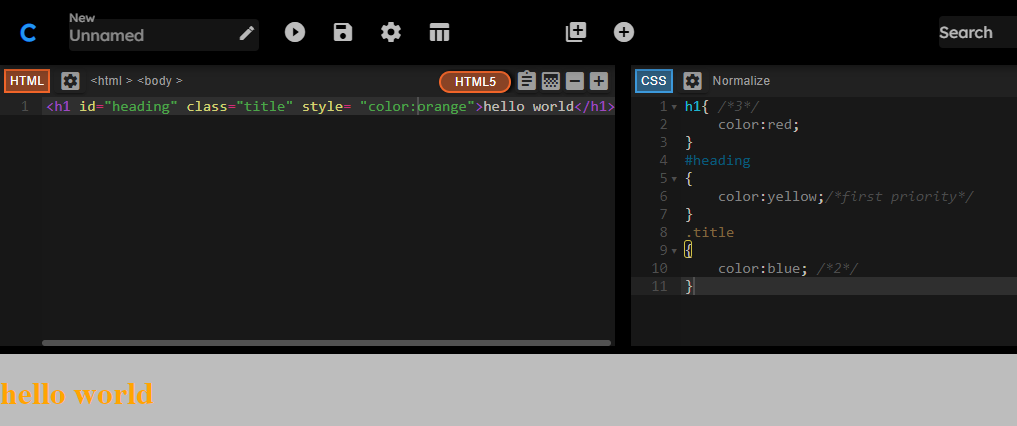
    transform: rotate(25deg);

}

**Media Query Breakpoints**

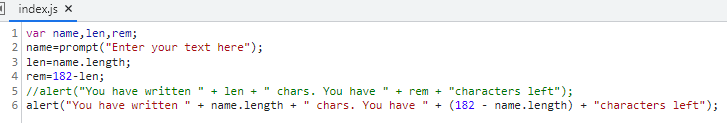
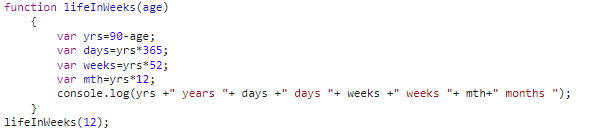
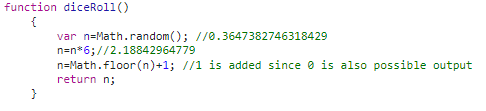
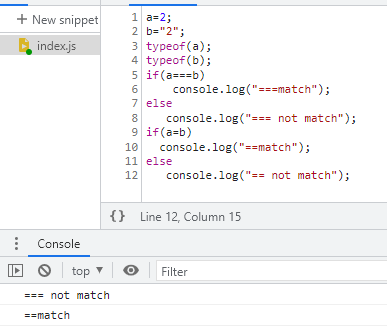
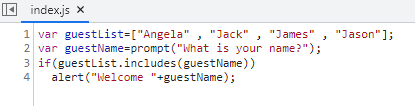
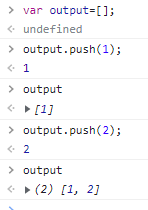
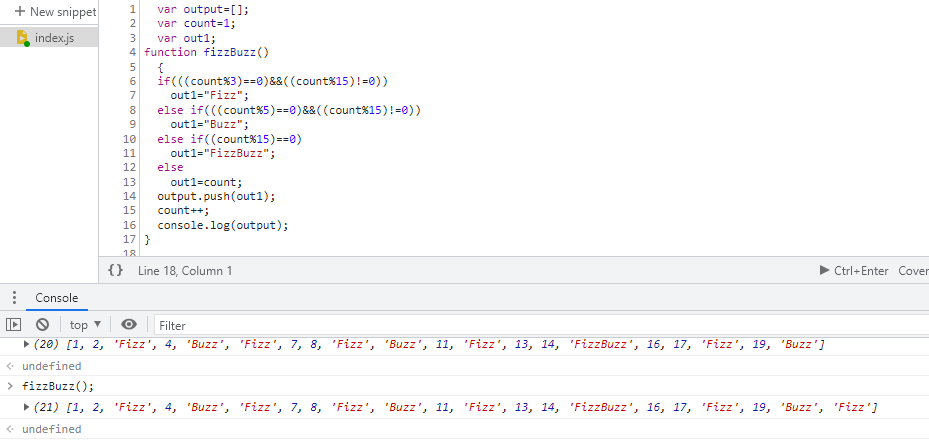
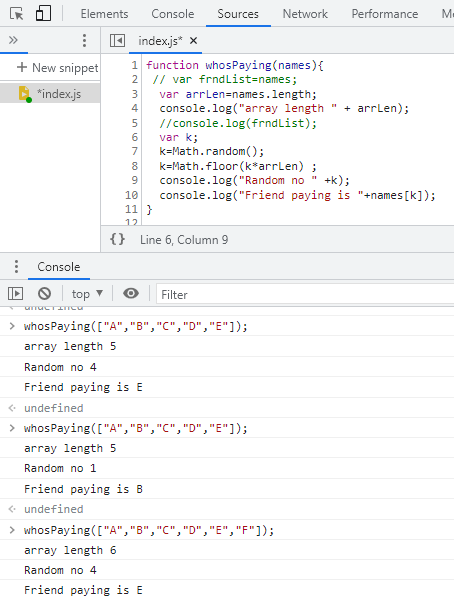
* To check if website is mobile friendly or not..click on this website <https://search.google.com/test/mobile-friendly>
* If we want text color “red” in certain conditions say when my sit eisprinted..Media Query is used for this purpose
* 
* In this case, color of text is Red only when print commamd is given.
* Syntax: @media <type><feature>
* Eg @media screen(min-width:900px){ //do something}
* 
* In this case the font shall change to color red if size of screen is 900px or less say like tablet or phone..it shall be red
* 
* If statement is @media(min-width:900px and max-width:1000px) then it gets triggered only when viewport is greater than 900px but less than 1000px
* The screen size can be seen by clicking Inspect->performance..the screen size gets mentioned in the upper right corner
* <aclass="nav-link"href="#footer"> Contact</a>
* This basically takes the cursor to the footer section of the screen

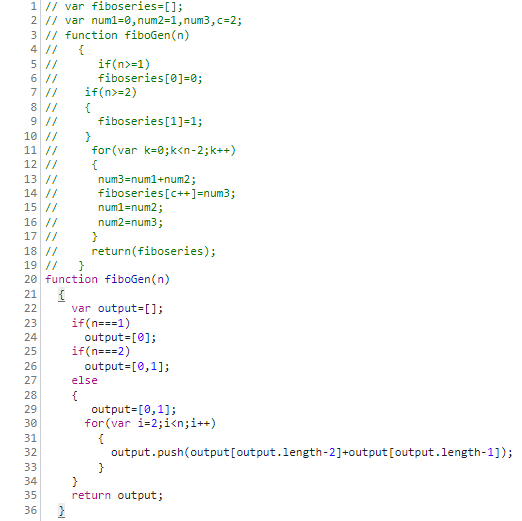
CODE REFRATORING

* #title.container-fluid{….}…hierarchial selector
* h1,p{color:red}
* 
* All h1 inside div gets affected
* 
* Element with id “title” that is inside parent with class container-fluid shud have red
* 
* All h1 that is child with element with class fluid-container shud be red (read from right 2 left)
* Combined selectors…. Selelector1.selector2{…} or Selelector1#selector2{…}
* Hierarchial then use space..if on same level then use dot
* 
* H1 that has id title shud be red..read from left to right
* 
* No div having id title..we have id with child title
* Diff b/w element.class and element .class
* 
* 
* 
* 
* Preference is first given to id then class then HTML element
* 
* 
* Inline styles are more specific hencemore priority
* <https://www.dailyui.co/> to practice UI design
* <https://collectui.com/> all UIs are collected
* <https://www.canva.com/> online graphics editing platform

JAVASCRIPT

Javascript is a scripting language used by programmers across the world **to create dynamic and interactive web content like applications and browsers**. JavaScript is so popular that it's the most used programming language in the world, used as a client-side programming language by 97.0% of all websites.

* EARLIER CALLED livescript
* Lets say for a play, script wud be Javascript and actors shall be HTML
* Javascript is an interpreted lang whereas java is a compiled prog language
* Interpreted Lang: Javascript, Python, Ruby
* Compiled: Java, C/C++, Swift
* It is the lang that poers the web
* To open Javascript console on browser..press F12 or ctrl+shift+I then goto console
* Lets say you wish to enter hello world then type
* alert("Hello");
* alert("world"); hold Shift after first statement and press enter before typing the 2nd line else only hello shall be executed after tping enter
* Better way is to go to Sources->Snippet->New Snippet
* Console:for testing single line but Sources for whole code
* Here alert is the function, hello is msg and ; denotes end
* <https://github.com/rwaldron/idiomatic.js/>
* Ctrl+L to clear console
* typeof(1.2); returns number
* Click Refresh and then click Empty cache and hard reload else the variable decl gets stored from previous program in memory
* VARiable name var not allowed,
* var name="angela";name.length
* 
* alert(prompt("Enter your tweet: ").slice(0,140)); this code limits the user to enter only 140 char
* var name=”Angela”;
* name=name.toUpperCase();
* name=name.toLowerCase();
* ask user for name..but if they send insmall then capitalize only 1stchar
* varuname=prompt("Enter your name: ");
* alert(uname.slice(0,1).toUpperCase() + uname.slice(1,uname.length));
* alert(uname.slice(0,1).toUpperCase() + uname.slice(1,uname.length).toLowerCase());
* Math.floor(x); for rounding
* Life in weeks challenge
* 
* Math.pow(height,2);math.round(var);
* Var n=Math.random();..any no b/w 0 to 0.9999999999999999 (16 dec places) doesn’t reach 1
* Lets say we gen random no by rolling a dice
* 
* **Software-generated random numbers only are pseudorandom**.
* = is not equal to=== in javascript
* 
* ===also checks datatype but == doesn’t check that
* 
* Program that prints nos from 1 to 100. For multiples of 3,print “Fizz” instead of the no and for multiples of 5, print”Buzz”, if both multiples of 3 and 5, print “FizzBuzz”
* <https://blog.codinghorror.com/>
* 
* Push always pushes item to the end of the array. pop is used to pop the last item from array… arrayname.pop();
* 
* Friend paying for dinner
* <https://www.jitbit.com/alexblog/249-now-thats-what-i-call-a-hacker/>
* 



* Inline css: An inline CSS uses the style attribute of an HTML element.<h1 style="color:blue;">A Blue Heading</h1>
* Internal css:An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

h1   {color: blue;}

cssshud always be put at the top in head section but with javascriptitshould be placed at end right before closing body tag

Inline javascript:

<bodyonload="alert('hello');">

Internal javascript:

<body>

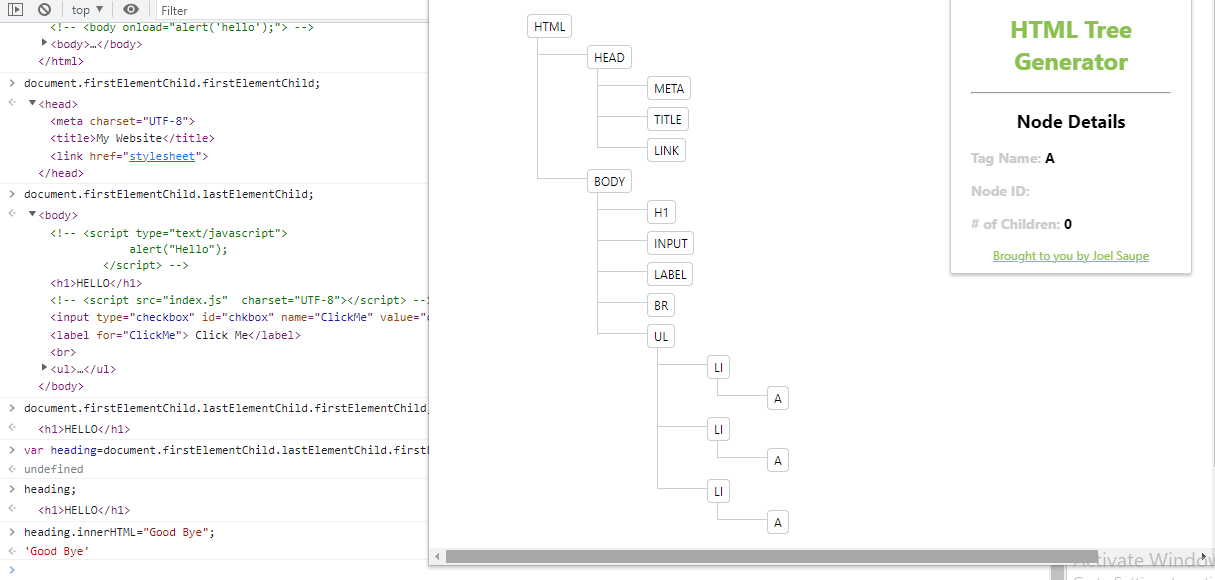
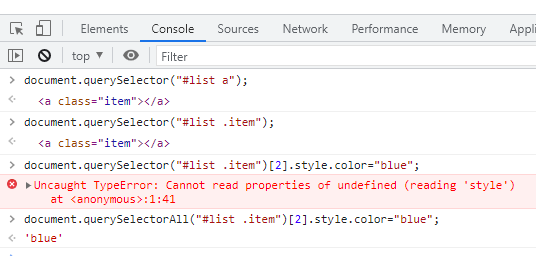
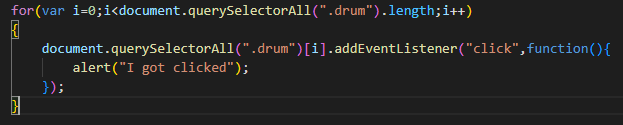
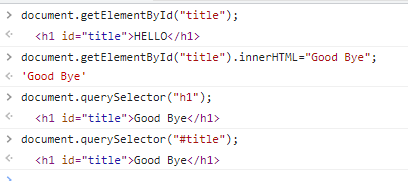
<scripttype="text/javascript">

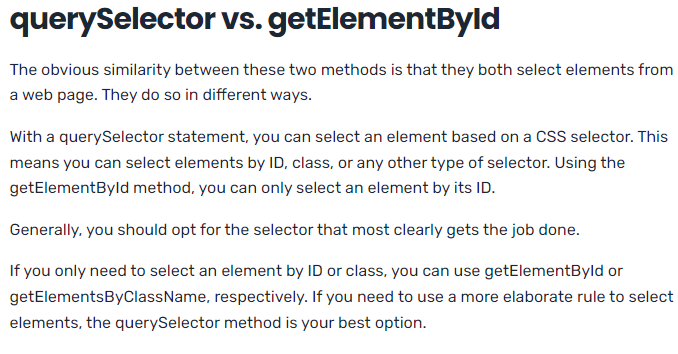
    alert("Hello");

        </script>

 <h1>HELLO</h1>

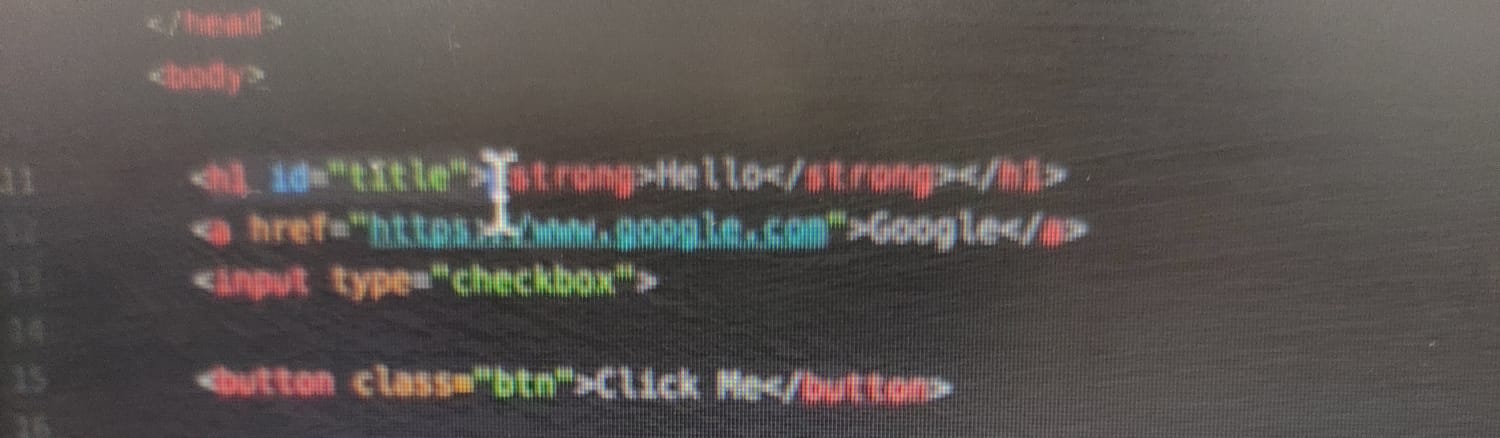
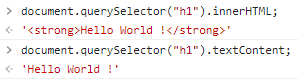
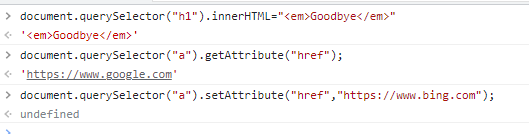
</body>

* DOCUMENT OBJECT MODEL
* It catalogs the webpage into individual objects that we can manipulate
* HTML tree visualizer is a free plugin
* 
* In the above code we manipulated <h1> to read “goodbye”from”hello”
* Object inside DOM have properties(innerHTML, style,firstChild) and methods(click(),appendChild(),setAttribute())
* 
* Query selector allows more complex quries since id, class, elemt tag name etc can be targeted
* Getelementmerthods are more broad and difficult to target individual object without targeting individual html
* 
* 
* **getElementById matches the id attributes to find DOM nodes, while querySelector searches by selectors**. So for an invalid selector e.g<div id="1"></div> , getElementById('1') would work while querySelector('#1') would fail, unless you tell it to match the id attribute (e.gquerySelector('[id="1"]')

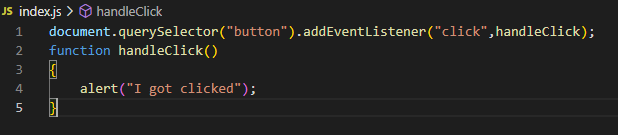
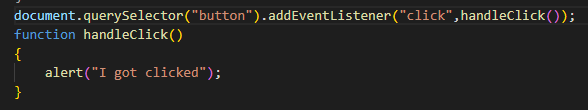
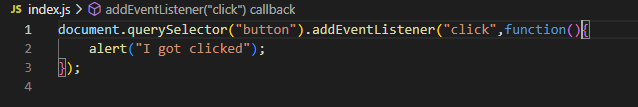
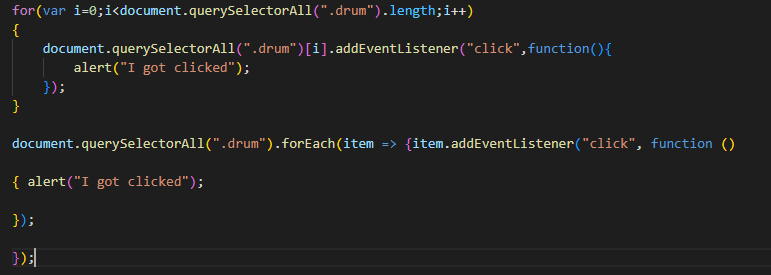
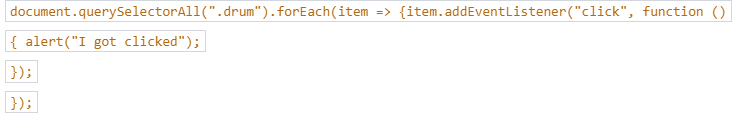
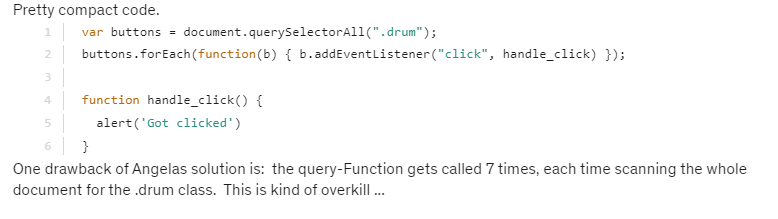
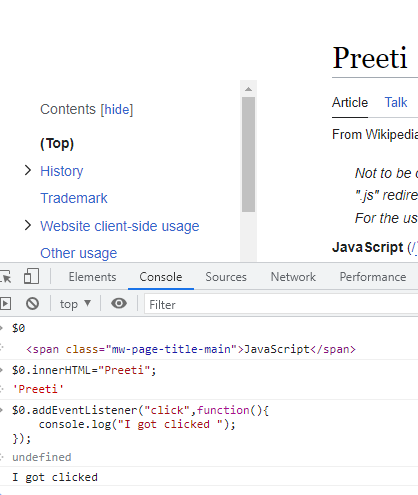
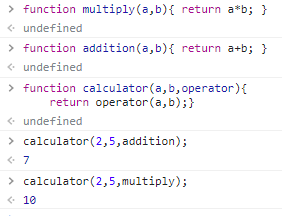
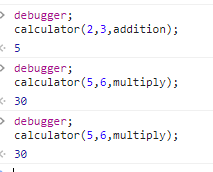
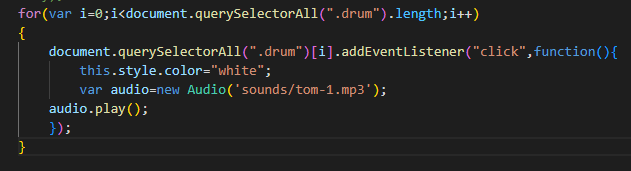
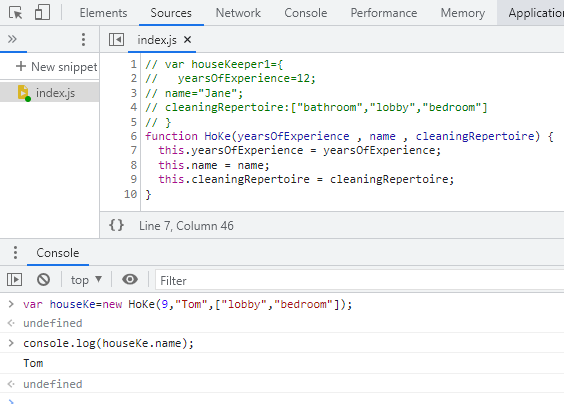
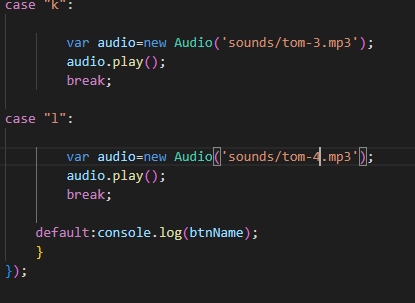


* 

innerHTML: The innerHTML property sets or returns the HTML content (inner HTML) of an element. It gives literally everything inside the elements tags

* 
* 
* 
* When writing, HTML also has go inside quotes else it considers a s a javascript code
* document.querySelector("h1").innerHTML="<em>Goodbye</em>"
* Manipulating attributes of element eg class, href, a, srcetc..everyhthing that goes inside tag other than tagname is an attribute
* 
* to read: <https://www.kirupa.com/html5/finding_elements_dom_using_querySelector.html>

Adding Event listeners to button

* 
* 
* What this code does is:it fins the first button in our code “w drum” button and adds event listener to this button to listen the click event listener to run the code
* 
* The below image isn’t right:Code runs until it hits script tag then it finds the index.js and then it will this code. If we had the parenthesis, then this wud have beena straight up method call, it gets called straight away when event listener is added..hence we drop the () and add the name of the js function as an input so that it can be called at a later time..basically waiting for the click event to happen before we call the handle click function
* Common way is to write it simply as an anonymous function
* 
* 
* 
* 
* HIGH ORDER FUNCTIONS AND PASSING FUNCTIONS AS ARGUMENTS
* 
* 
* High order functions are functions that can take other functions as inputs.
* 
* After debugger, shift+enter to write function call then step into
* 
* JAVASCRIPT OBJECTS
* Constructor name is not camelcased.its first letter needs to be in capital
* 
* 
* Free website building tools: themeforest, wix,neog.camp,rocketcareer
* JQUERY
* Javascript library that ill prevent our fingers from breaking
* John Recce invented it.
* Document.queryselector(“h1”) in jquery is jQuery(“h1”) or $(“h1”);
* CDN: <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
* $() =jQuery()
* $("h1").css("color","red");
* Place the cdn link to jquery just above the link to js(placement matters)
* 
* 
* B4 wevefinishedloadingjquery librry,index.js might like access it so to check if jqury library isready by the above image. So better is to place it at the end

HOW MINIFICATION WORKS TO REDUCE FILE SIZE

* On copy pasting the link link of jquery<https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js> in the browser, we can see that it is basically the minimized version of javascript only.
* There is a site [~~https://www.minifier.org/~~](https://www.minifier.org/)where one can miify all the js and csscode..actual link <https://www.toptal.com/developers/javascript-minifier>
* $(“h1.title”) or$(“#header h1”)..aparts from this jqury allows
* Document.querySelectorAll(“button”)=$(“button”);

MANIPULATING STYLES WITH JQUERY

* $(“h1”).addClass(“big-title margin-50”);
* $(“button”).hasClass(“margin-50”); gives true or false

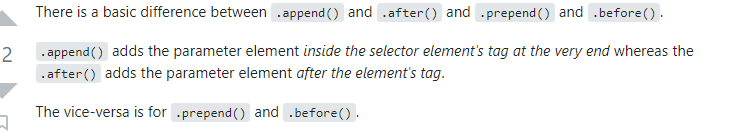
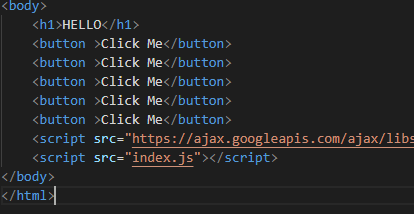
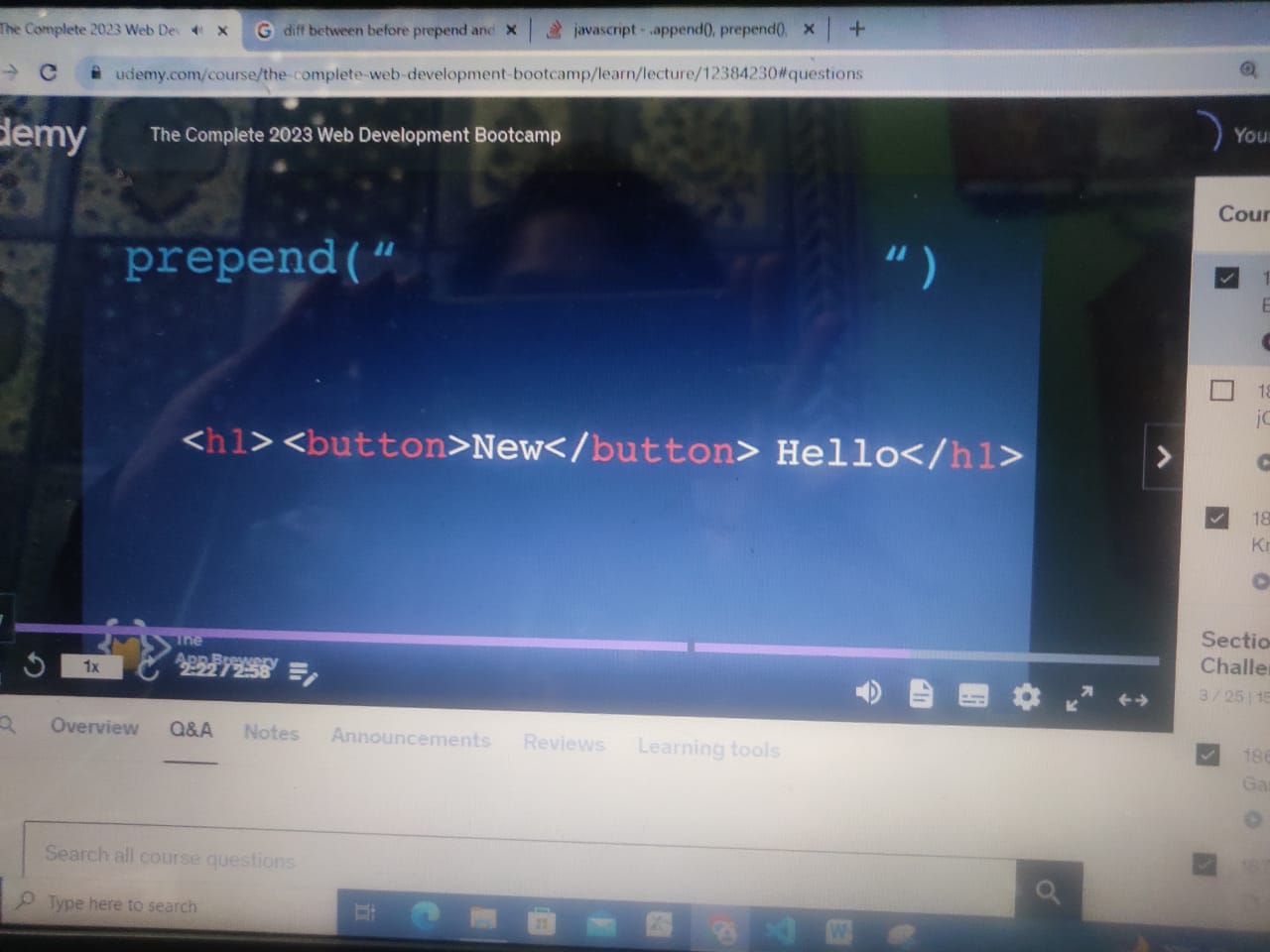
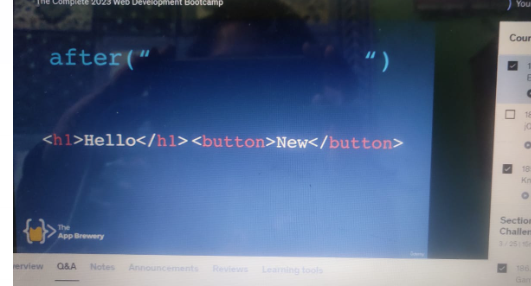
MANIPULATING TEXT WITH JQUERY

* $(“h1”).text(“Bye”); or $(“button”).text(“click me”);
* Instead of innerhtml..we will be using $(“button”).html(“<em>Hey</em>”);

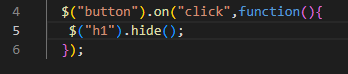
MANIPULATING ATTRIBUTES WITH JQUERY

* Console.log($(“img”).attr(“src”));
* $(“a”).attr(“href”,”https://www.yahoo.com”);

ADDING/REMOVING ATTRIBUTES WITH JQUERY

* $(“h1”).before(“<button>New</button>”); $(“h1”).after(“<button>New</button>”);
* $(“h1”).prepend(“<button>New</button>”);
* $(“h1”).append(“<button>New</button>”);
* $(“button”).remove();
* 
* 
* WEBSITE ANIMATION WITH QUERY
* 
* 

WEBSITE ANIMATION WITH JQUERY

* 
* $("h1").toggle();
* $("h1").fadeOut();
* $("h1").slideUp();
* $("h1").slideDown();
* $("h1").slideToggle();

Animation

  $("button").on("click",function(){

   $("h1").animate({opacity:0.5});

  });

It is to note that inside the animate, b/w curly braces, u canonly add the cssrulesh having numeric value..like 0.5 etc..cant change color

$("button").on("click",function(){

   $("h1").animate({margin:20});

  });

You can chain

$("button").on("click",function(){

   $("h1").slideUp().slideDown().animate({margin:20});

  });

Node.js

* Create backend using javascript
* Superfast+scalable and fast applications
* Ebay, uber, twitter etc use node.js as backend
* Javascript locked behind bars..ie. inside browser..only interact with browser, req info from browser, add event listener to components in browser..we r not able to write javascript code beyond the boundaries of the broswr…cant reach out of the browser and reach uers copyter toget access to users file system etc..
* It allows us to take jacsrcipt outsi of browser and interact with the h/w..we can use jaac inside website to give funct and behaviour like..animation or dropdown menu..allows to interact directly
* We can run node.js on our own computer as well as someones else comp or server
* Result can be sent back to client..happens everything on server rather than client side

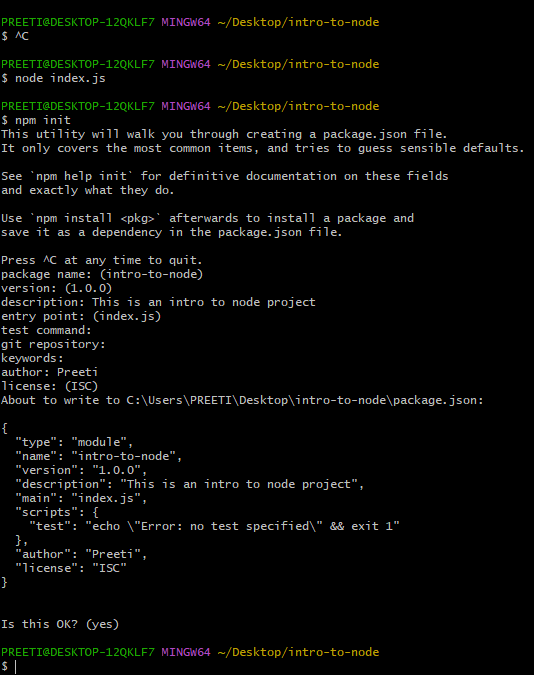
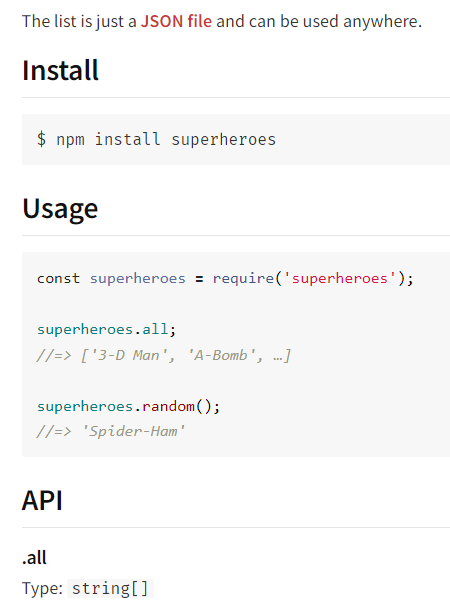
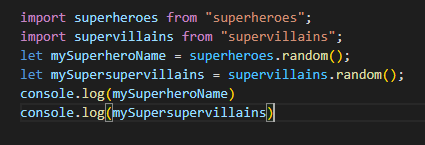
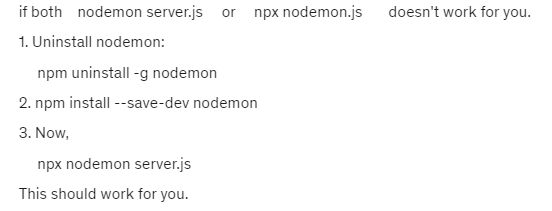
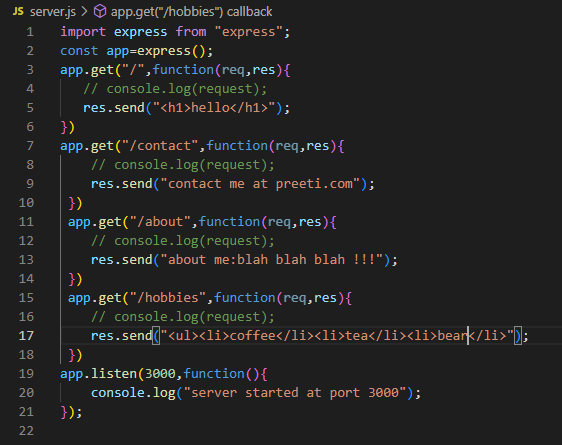
REPL=READ EVALUATION PRINT LOOP

* Allows to execute code in bytesize chunks..to access just write node inside command line and hhit enter
* > means inside REPL
* Instead of typing console in full just write con and press tab…press tab 2 times if necessary
* Type console. To see result for console
* To exit the REPL..either say .exit and enter or hit ctrl+c twice

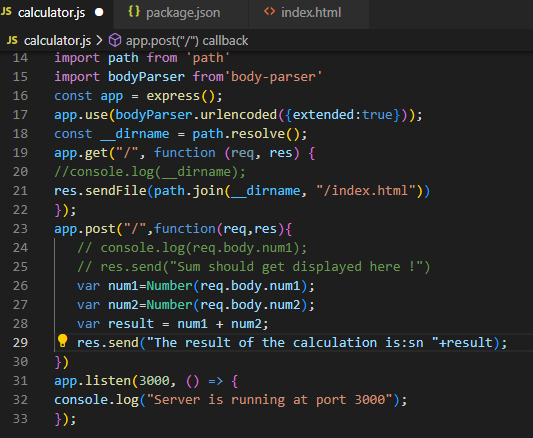
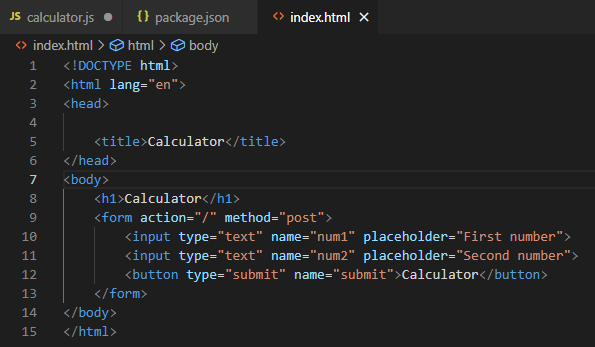
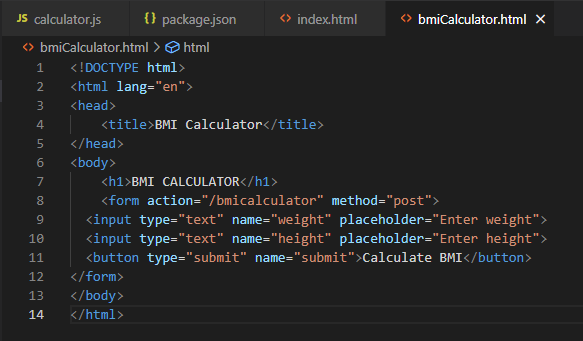
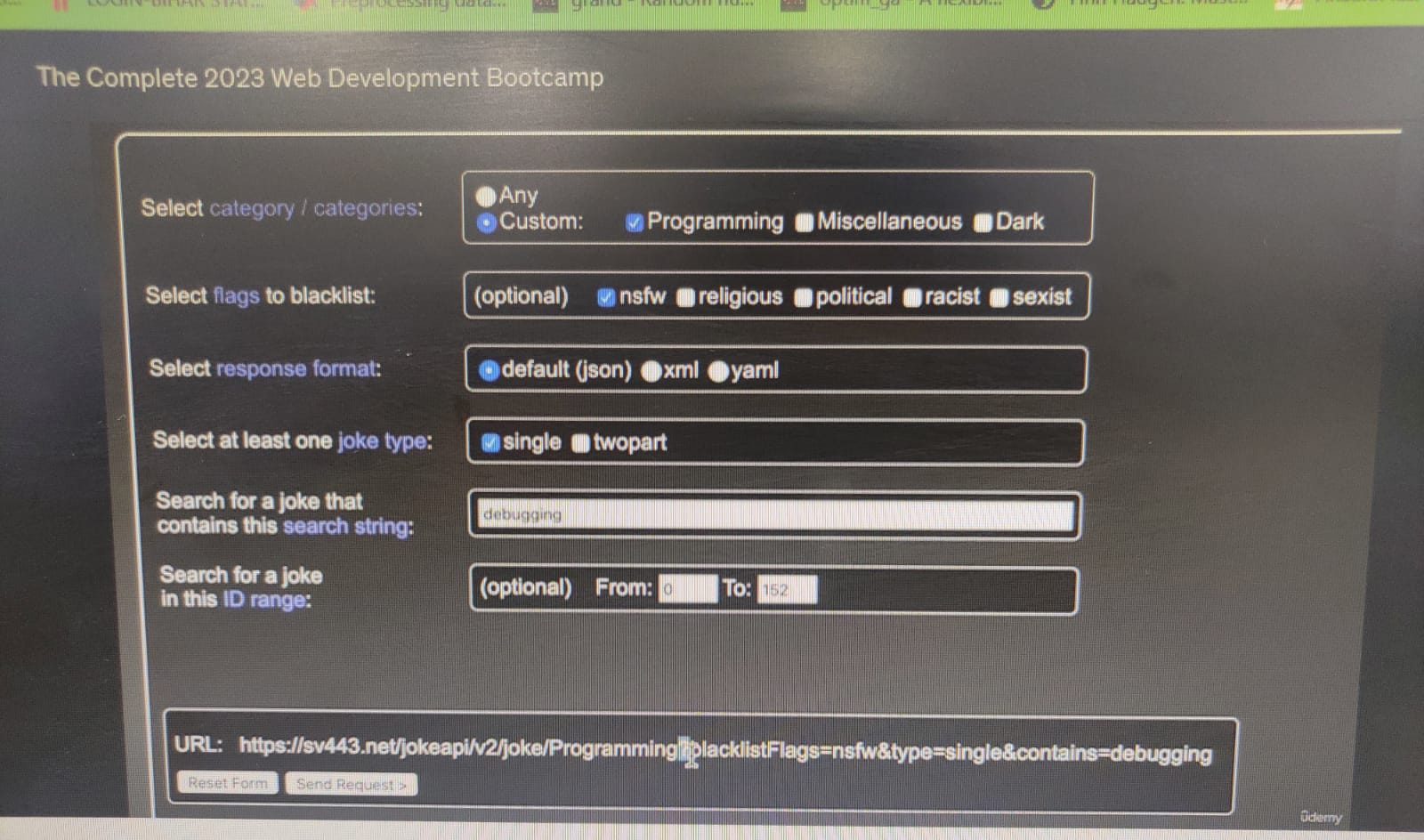
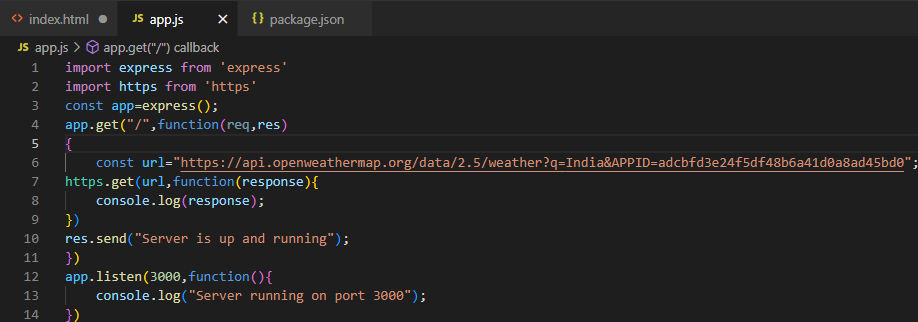
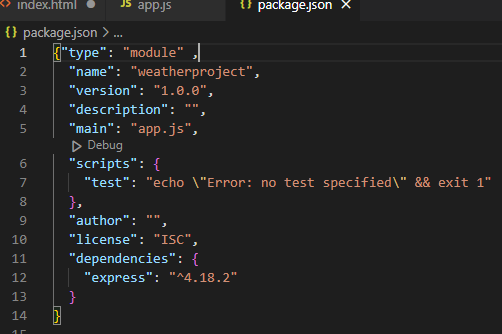
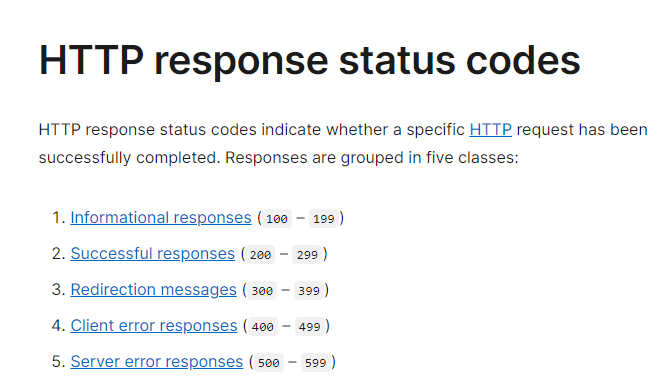
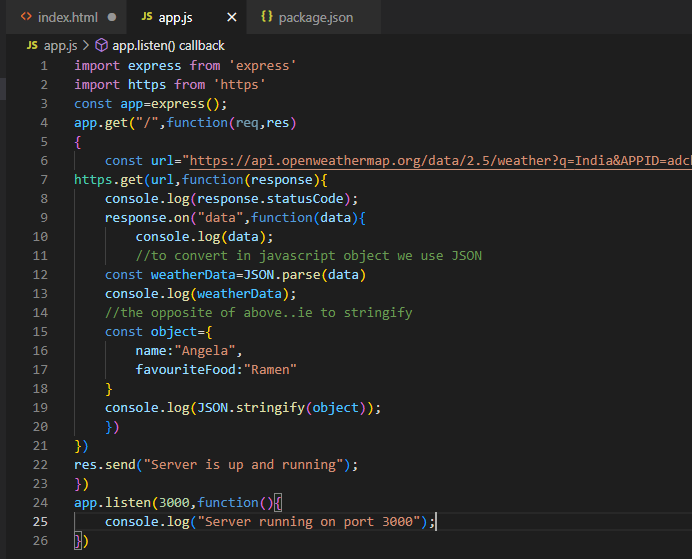
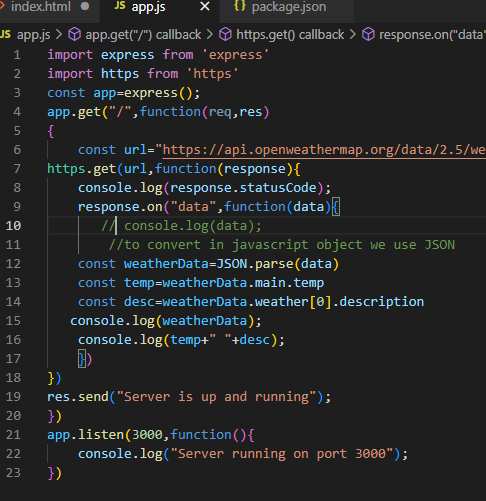
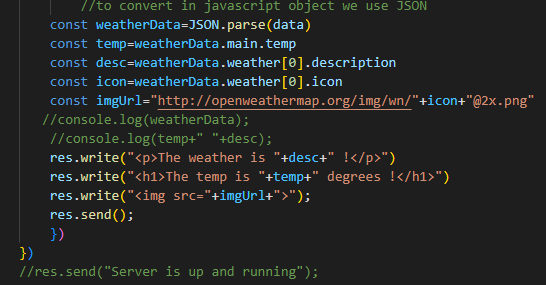
HOW TO USE NATIVE REPL BUNDLES

* <https://nodejs.org/api/fs.html>

NPM package manager

* Pckake manager for external modules. Worlds largest oll of these pack of code
* 
* <https://www.npmjs.com/>
* 
* JSON stands for **J**ava**S**cript **O**bject **N**otation
* JSON is a lightweight format for storing and transporting data
* JSON is often used when data is sent from a server to a web page
* JSON is "self-describing" and easy to understand
* (var and let are both used for variable declaration in javascript but the difference between them is that **var is function scoped and let is block scoped**. Variable declared by let cannot be redeclared and must be declared before use whereas variables declared with var keyword are hoisted)
* 
* EXPRESS
* Node framework..just as jquery is to javascript, express os for node
* It is a bunch of code writtenby others that add extra fetures, organize and structure code built with web appl structured with node
* Allows to interact directly with hw of computer..we can create dektop appl…
* 
* Press ctrl+c to quit server and then node.js to run
* url in browser: <http://localhost:3000/>
* instead of typing ctl+c and then node.js to restart te server again and again, install nodeman.io..utility that u can install by npm that will monitor 4 changes in sourcecode and restart
* type npm install –g nodeman in hyperterminal
* the above wont work..try as below
* 
* 
* CALCULATOR PROGRAM
* This runs code on server unlike uptil now where all code worked on client browser side. This shall be done using node.js
* All cal done on server side only
* To open visual studio from terminal, just type code.. write code dir name to open dir

On terminal, type (1) npm init..then couple of ok’s..then npm install express( "type": "module", to added in package.json…then (2)“Install express body-parser” on terminal then (3) “npx nodeman calculator.js”

* Res.sendFile(\_\_dirname)..gives file path of current path no matter where it is hosted
* List of HTTP status codes..200=OK..400=client error
* To tap into form data, install npm package called bodyparser…npm install body-parser
* app.use(bodyParser.urlencoded({extended:true}));
* why use bodyparser? Allows to go any of routes and parse into req.body and this is the parsed version of the http request
* by using bodyparser, we were able to parse http req that we get and by using urlencoded we get access to form data..and then tap into each as if they were proprtes of object body
* 
* 
* BMI CALCULATOR CODE
* 
* 
* APPLICATION PROGRAMMING INTERFACE
* Set of commands, functions, protocols and objects that programmers can use to create software or interact with an external system. API shall act as a bridge bw your server and someone else’s server
* 4 important aspects:-
* Endpoints: every API that inetreacts with external API
* Paths
* Parameters
* Authentication
* 
* The parameters come after “?” separated by “&”. They are in key value pair in the form of key=value
* API AUTHENTICATION AND POSTMAN
* <https://home.openweathermap.org/>
* JSON=JAVASCRIPT OBJECT NOTATION
* It is in readable format and can also collapse to take little space
* Browser addon: json viewer pro: free to download and it prettifies the code in tree, chart structure etc
* <https://httpstatusdogs.com/>
* 
* 
* 
* 
* 
* We can have only one res.send but we can have multiple res.write
* 
* (The name attribute is for submitting a form element to the server; many elements may share the same name (e.g. radio buttons, which must have the same name within the set). The id attribute is for uniquely identifying any element (not just form elements). It must be unique throughout the entire document.)
* GIT
* Git init, then git status shows untracked files in red..inside working dir but it is not yet in staging area…to add to staging area…use git add filename..then on typing git status..it turns green. Next is to type git commit –m “message”….then type git log to see the commits made….better way to add files is git add . which brings all files to staging area
* Working directory->staging area(git add)->local repository(git commit)
* Use git checkout filename to revert the last position in local repository
* Use git status to see the changes…to see the changes done.. use git diff filename

GITHUB:Remote repository

* Open a new repository..then go to git
* Git remote add origin url(origin is name of your remote as convention..paste theurl)

Eg $ git remote add origin https://github.com/vpreeti/Story.git

* git push -u origin master this pushes local repo to remote repository using u flag which links your remote and local repo and then we push it to remote…master is the main branch of all our commits (origin=remote,master=name of branch)
* in github goto insights->network to see the progress
* ls –a toseehidden files

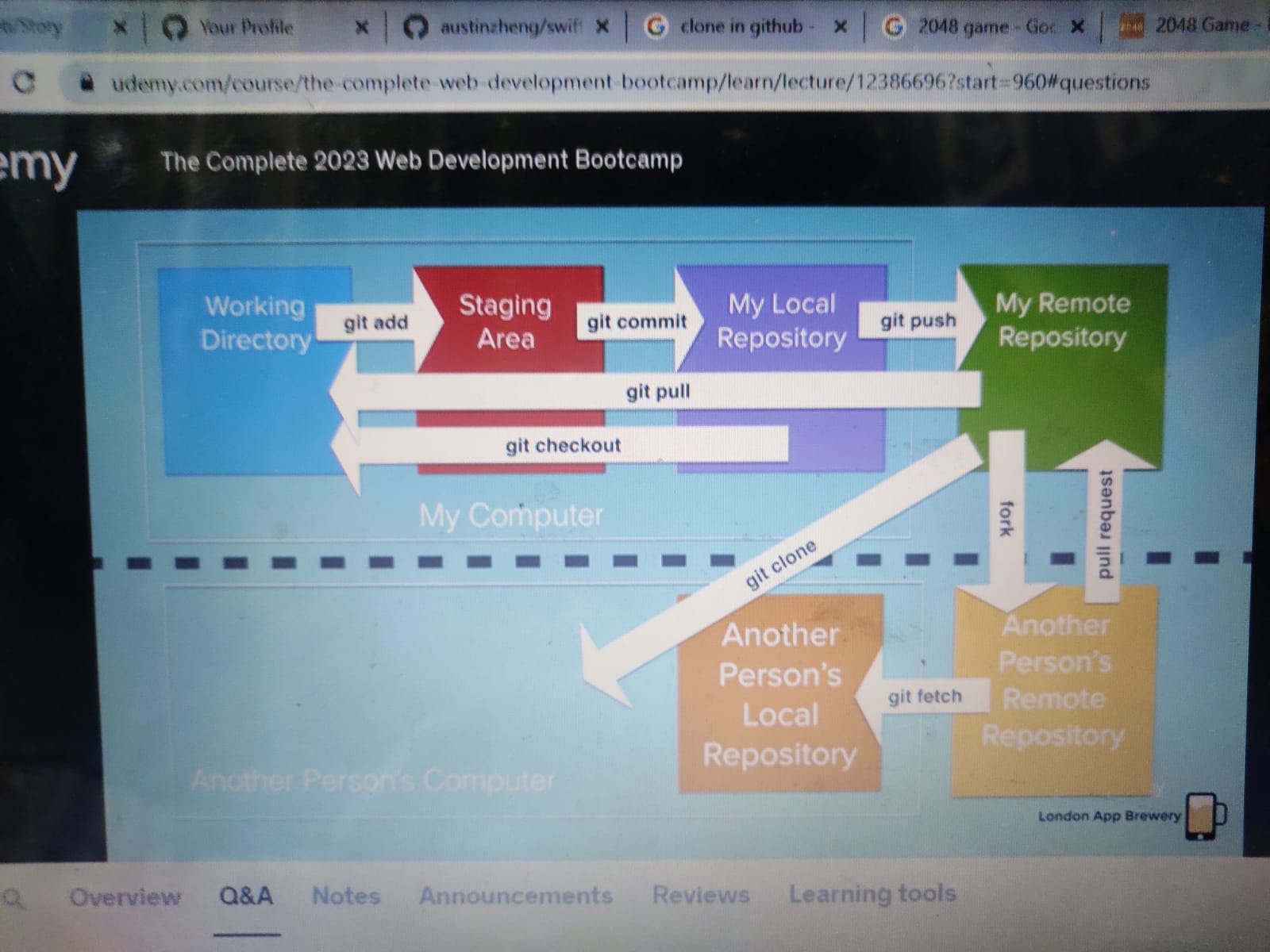
GIT IGNORE

* touch .gitignore..this file can be seen in ls-a
* 1. Git init 2. Git add . 3. Git status
* To remove files fro staging area… git rm --cached –r .
* Go to gitignore file and place the file to be ignored in separate lines
* Use # to comment, \*.txt

CLONING

* To pull from remote to local repository..pulls all commit and version
* Git clone httpscode

BRANCHING AND MERGING

* Git branch alien-plot
* To change the branch..write git checkout alien-plot
* Merge brach alien-plot
* :q! to exit editor
* Fork creates a completely idependent copy of git repository. Git clone creates a linked copy that will continue to synchronize with the target repository
* New person after forking will clone that repository locally. If she adds features then after commit changes she can PUSH to her remote repository. If she wants the owner to incorporate the changes then she can do by pull request(if not havin access) owner can the approve pull request and merge
* 
* Write npm I body-parser…..i is shortcut for install

TEMPLATE

* EJS: Embedded Javascript Templating

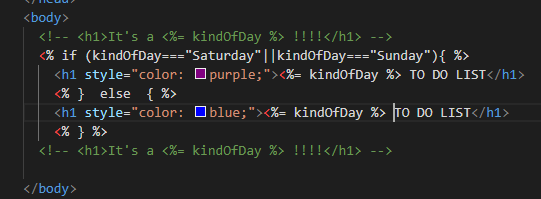
### Go to <https://ejs.co/> and then goto Using EJS with Express

1. npm install ejs

* app.set('view engine', 'ejs');
* make a folder named”views” and inside it make a file “list.ejs”
* <h1>It's a <%= kindOfDay %> !!!!</h1>

In app.get function,write:-

res.render("list",{kindOfDay:dayName});

* To beautify code in visual studio, use Shift+Alt+F
* 
* Logic should be in Javascript file so use it sparingly
* Diff bw var and let is that if we use var inside curcly braces,,say if or for then this var is global and can be accessed elsewhere also but if wedeclare it as let then it is local…always use let
* Express Routing Parameter
* app.get('/users/:userId/books/:bookId', (req, res) => {
* res.send(req.params)
* })
* <https://expressjs.com/en/guide/routing.html>

Eg   app.get('/compose/:someText', function(req, res) {

    //res.send(req.params)

    console.log(req.params.someText);

  })

* What is NPX?
* NPX stands for **Node Package eXecute**. It is simply an NPM package runner. It allows developers to execute any Javascript Package available on the NPM registry without even installing it.
* Express,ejs,requests are all libraries
* <https://lodash.com/>

## **Why Lodash?**

Lodash makes JavaScript easier by taking the hassle out of working with arrays, numbers, objects, strings, etc.Lodash’s modular methods are great for:

* Iterating arrays, objects, & strings
* Manipulating & testing values
* Creating composite functions
* Install: Npm i lodash