MEAN Stack Notes

Technologies

1. GIT
2. HTML
3. CSS
4. Javascript
5. ES6+
6. Typescript
7. Angular
8. MySQL
9. MongoDB
10. Node.js & Express.js
11. DevOps
12. Micro Front end
13. Case study.

GIT:

It is a version controlling system.

It helps to collaborate each others work in a team.

There will be two repositories in the GIT

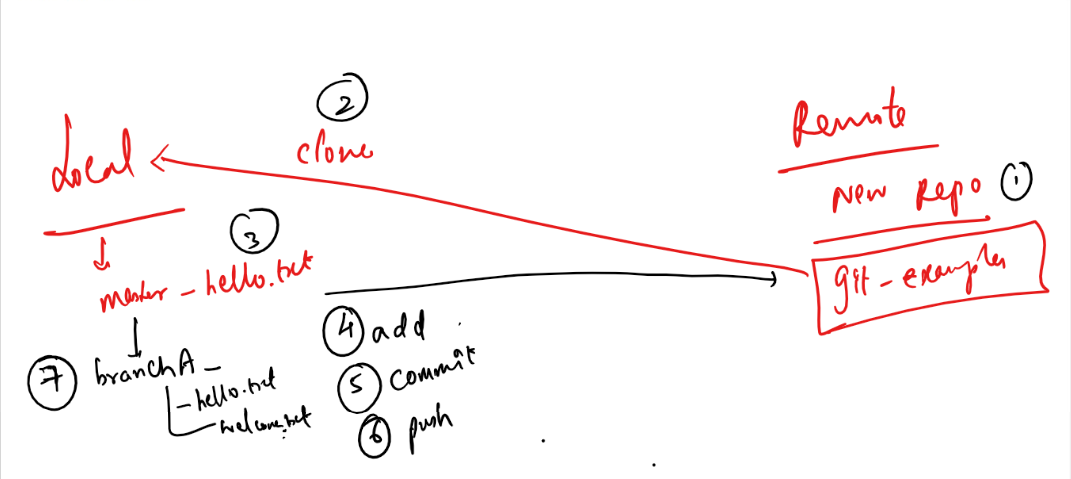
1. Remote: Will have access to everyone
2. Local: Local to the particular individual

Git provides commands to update each others work

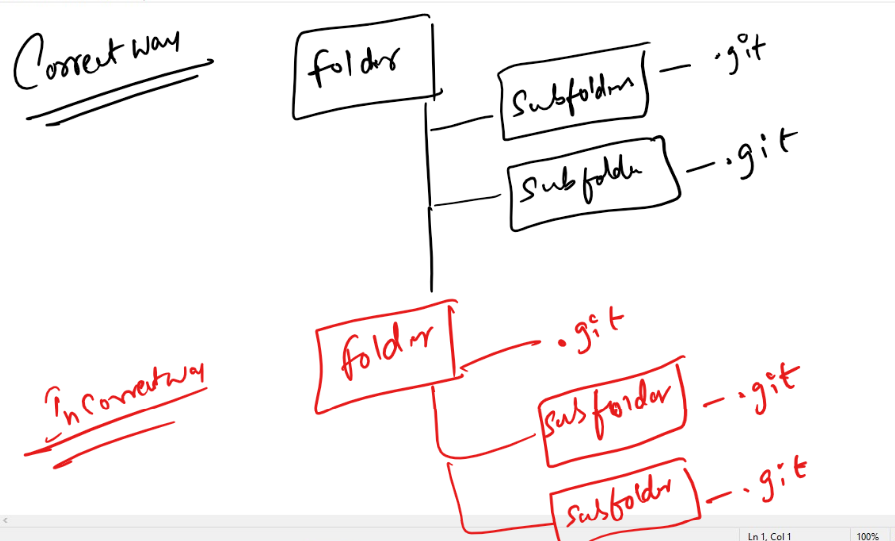
1. push: It tries to update the remote repository from the local repository
2. pull: It tries to update the local repository from the remote repository
3. clone: It creates a copy of remote repository in the local repository, it is done for the first time before you can enter push/pull
4. merge: It is used to manually merge the updates in the local repository from the local branch
5. add: It is used to add the changes to the staging area, only the changes in the staging area can be committed and pushed
6. commit: it is used to commit the changes, it creates one unique id which is used at the time of merging
7. status: It is used to see the changes done in the local repository, it shows the changes in the staging area and the changes not in the staging area

Git branches:

It is a pointer of particular commits, by default git gives you one branch called master/main branch, but every individual must work in custom branch not in the master branch

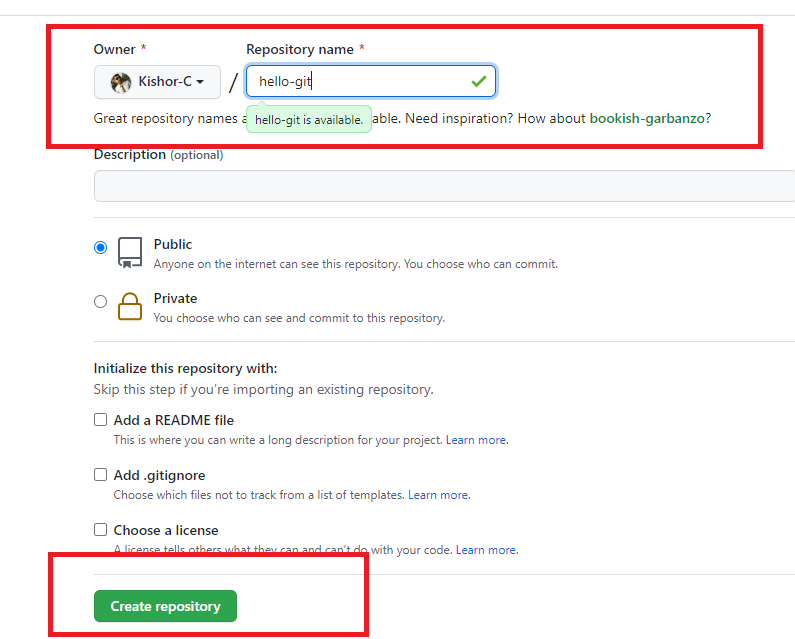


You must have a single .git folder which represents the git repository in the parent directory, but nested repositories you must not have ie.., an enclosing folder having .git and the sub folder also havening .git.

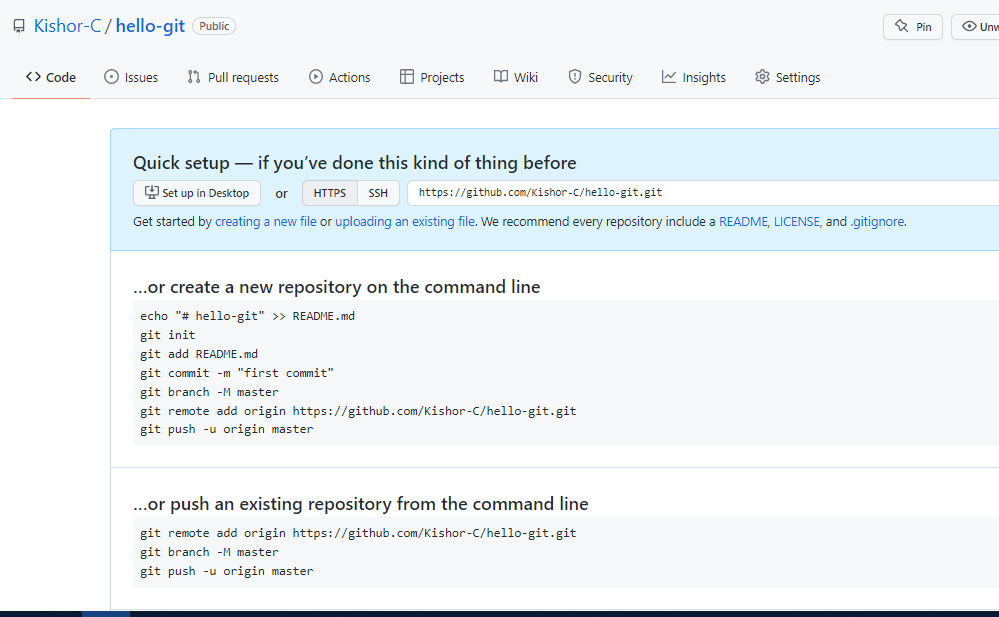


Steps to work with GIT

1. Create a GIT account & login
2. Create a remote repository & name it

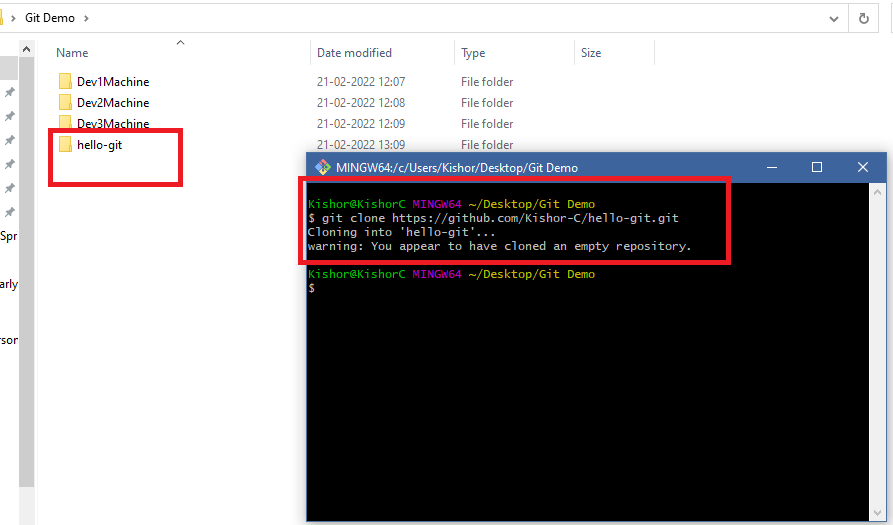


1. After you create the repository you will see the empty repository in the git website



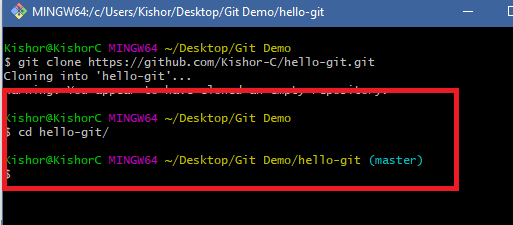
1. Clone the Remote repository in the local machine

Command: git clone URL



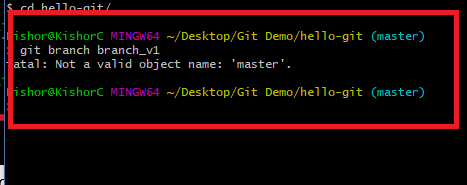
Note: You don’t see the branch name in the GIT bash, because you need to navigate to hello-git

1. Navigate to the local repository cloned



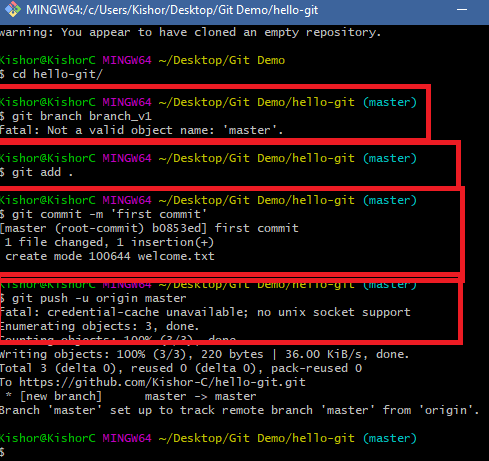
1. Create a custom branch when you want to do any changes/updates

Command: git branch branch\_name

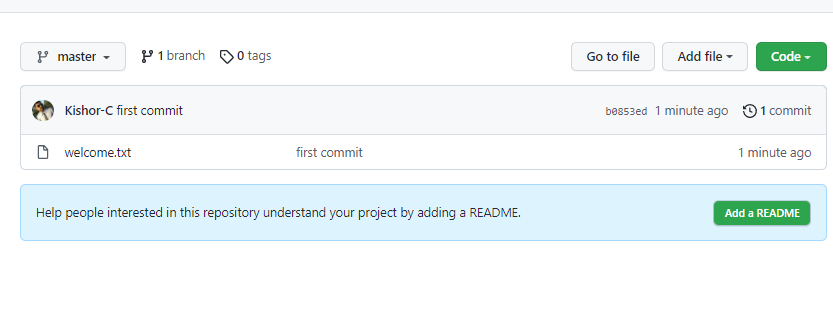
Command: git checkout branch\_name

Note: You may get this fatal error, when there are not commits in the master, i.e., if its empty repository you can’t create branch

To avoid this in the master branch create 1st commit

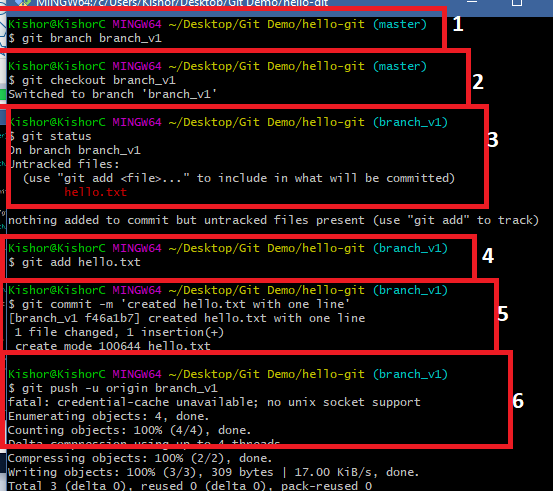


In the Remote you can see that first commit

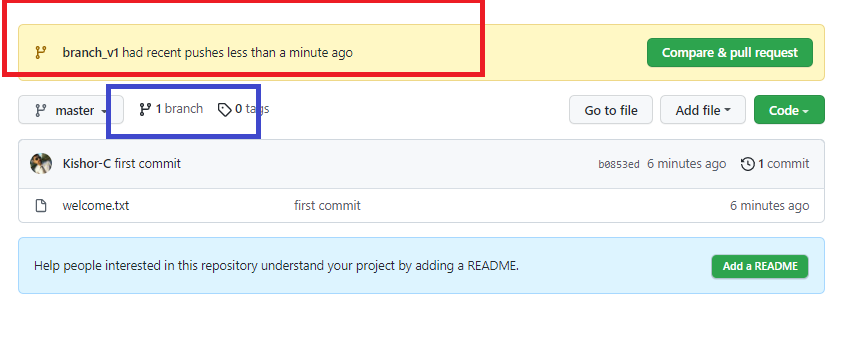


Note: It doesn’t ask you to create pull request, as we pushed master branch

1. Now we can create a custom branch and push that branch to the Remote repository

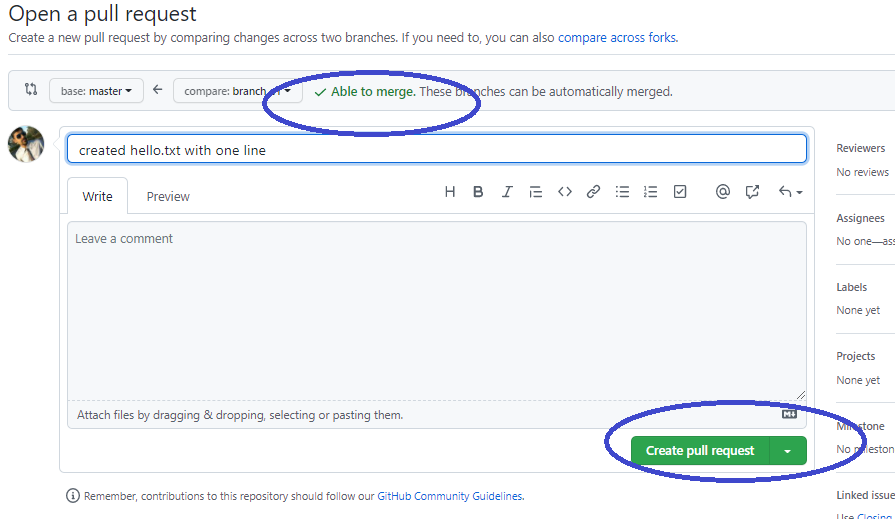


1. Now you can see a pull request in the Remote repository

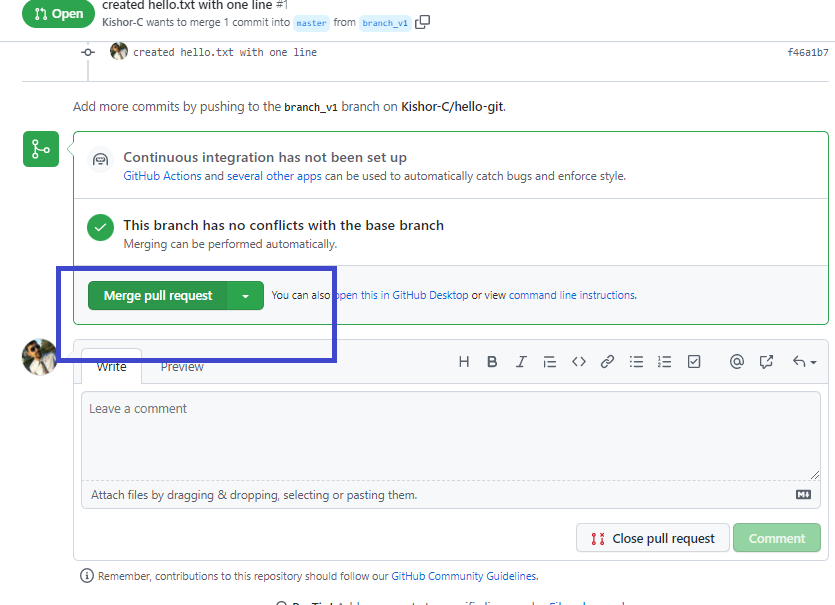


Note: Sometimes you may not see the recent push branch with Compare & Pull request, in that case you click on the branch highlighted in blue color

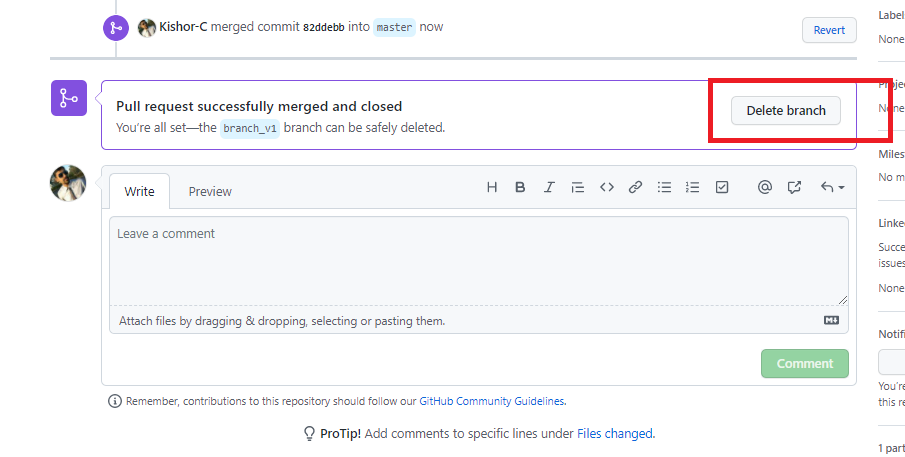
1. You can click on compare & pull request to update the remote or close pull request when there’s a conflict, in either of the case you must delete custom branch in remote repository.



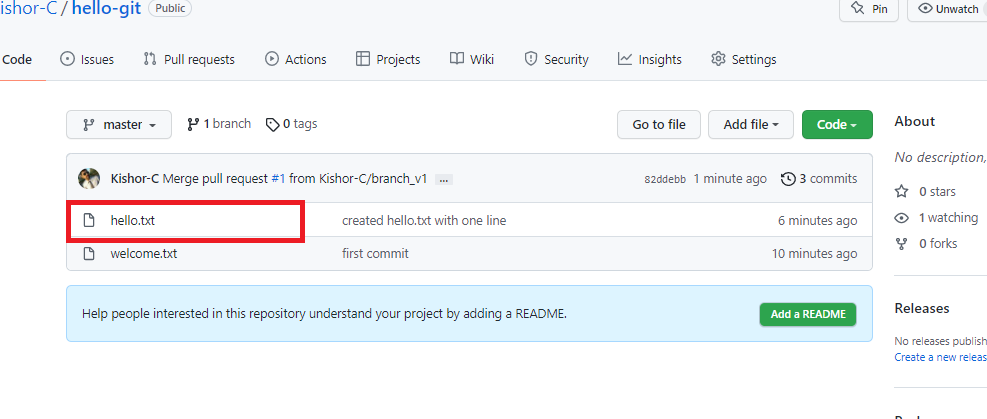
Merge the branch & delete



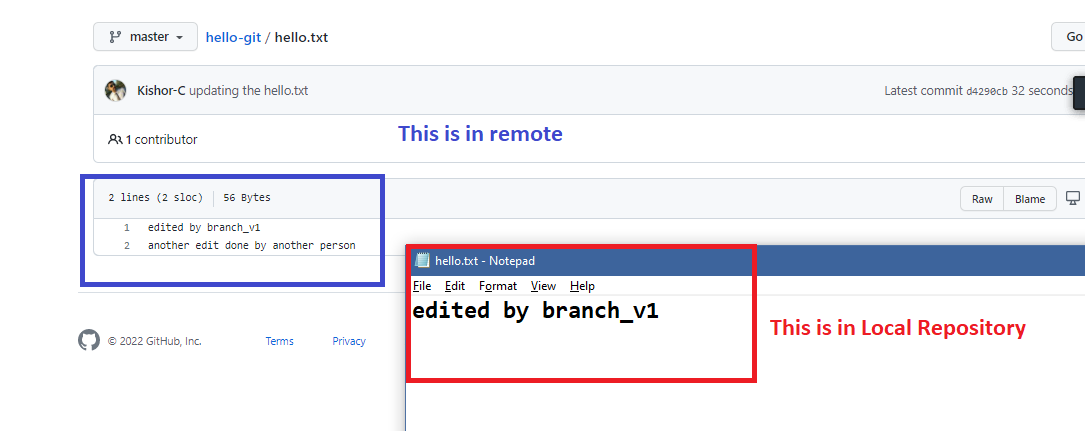
Delete branch



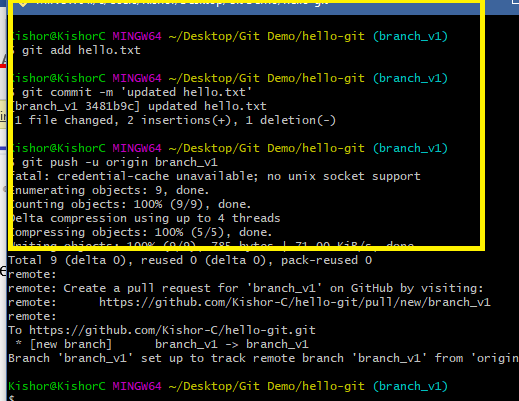
1. You can see the changes in the remote



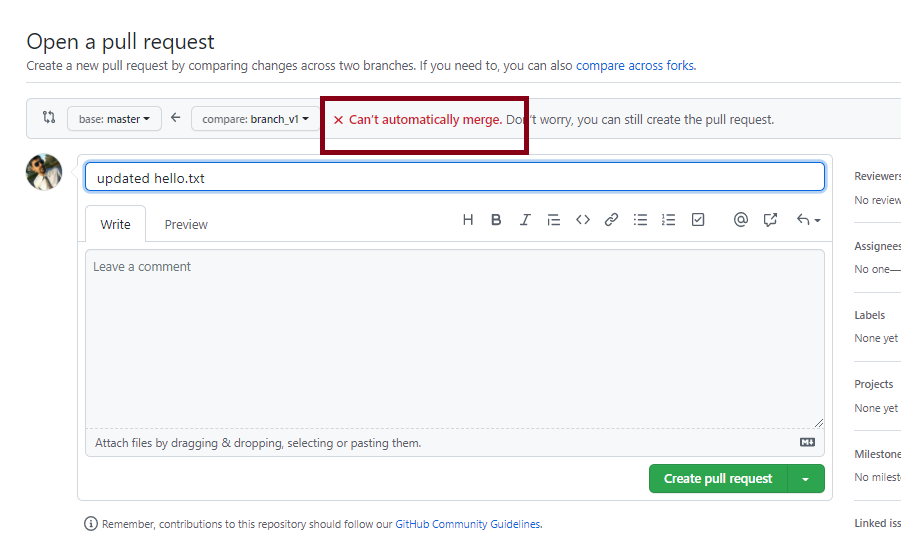
1. If you get a conflict then the local repository must pull the changes of master to its master branch, suppose for the below changes we get conflict when we try to update



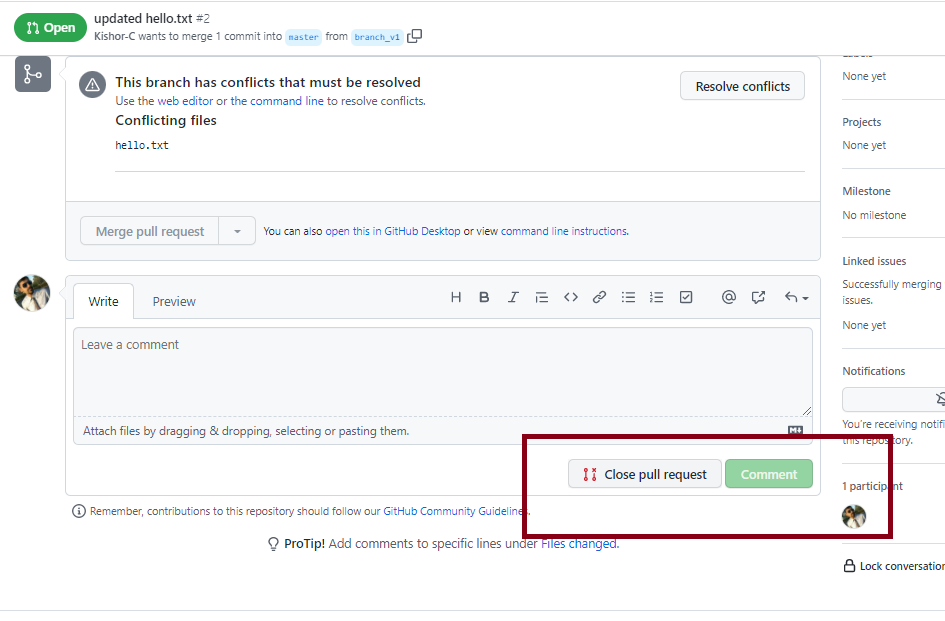
1. Add some content to the hello.txt & try to push the branch



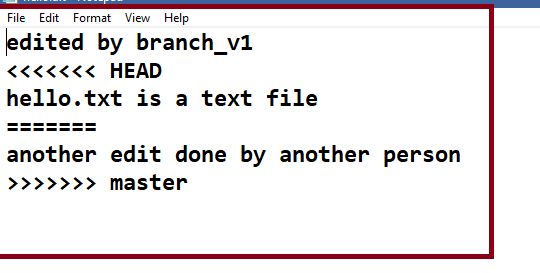
1. In the remote we see a pull request, but it can’t merge as remote master & the custom branch has conflict



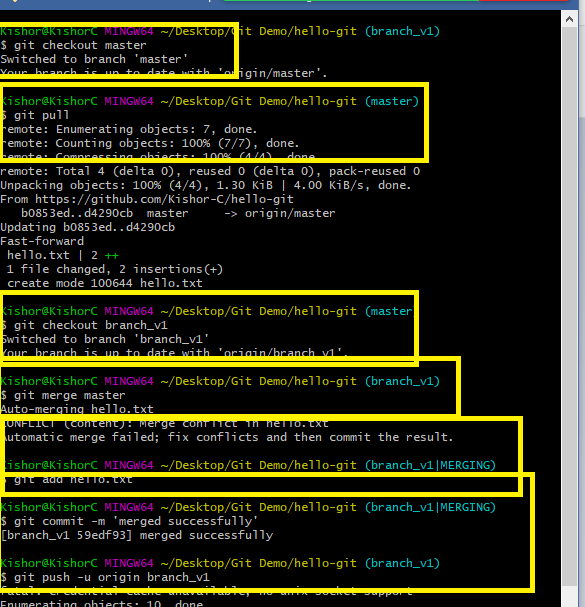
1. Close the pull request & delete the branch in the remote



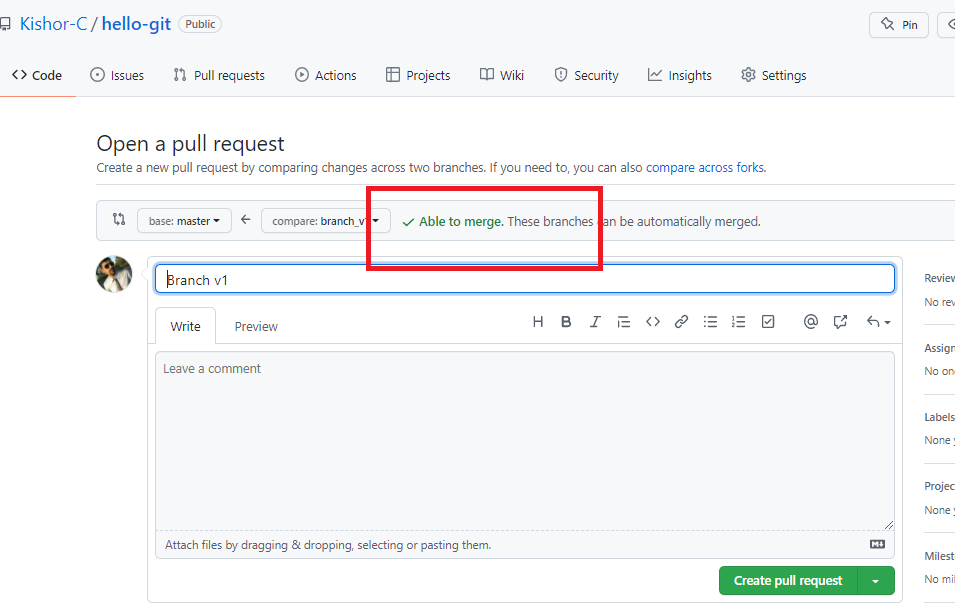
1. Local master must pull the remote master, then the custom branch should merge the local master and resolve the conflict
   1. Checkout to master: git checkout master
   2. Pull the remote master: git pull
   3. Checkout to custom branch: git checkout branch\_v1
   4. Merge the master with custom branch: git merge
   5. You will get conflict, edit that file, add & commit & push, you will the conflict file as below:



* 1. All the steps performed.



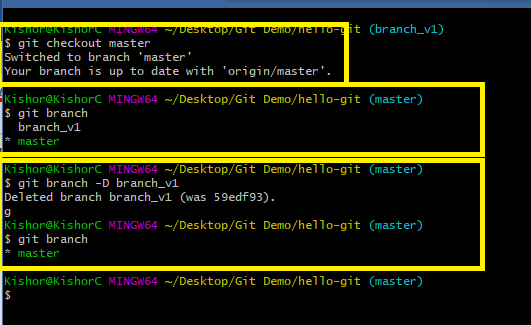
1. Now you can see the pull request in the remote

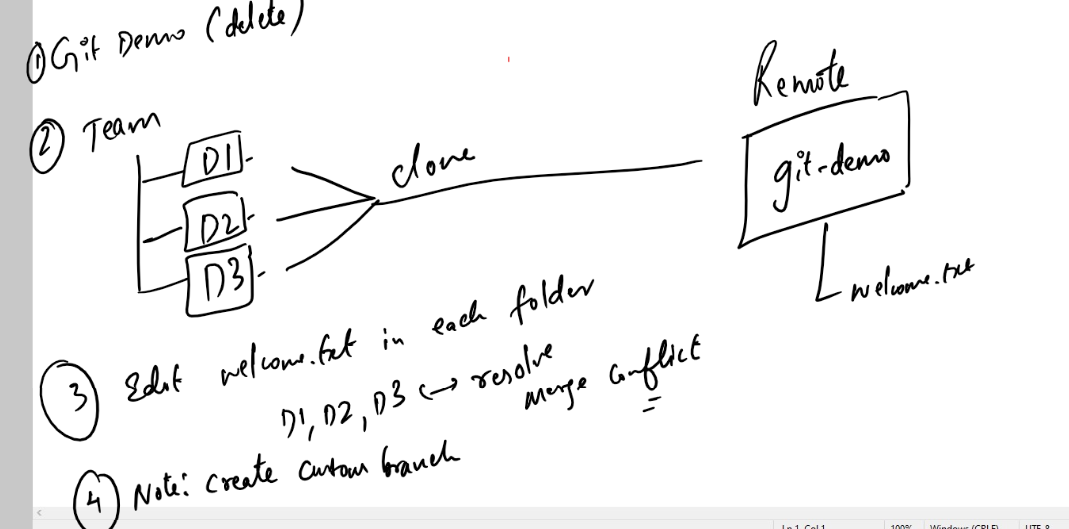


1. Create pull request & merge it and delete the branch in the remote, if required delete it in the Local also

Command: git branch -D branch\_name

Note: Checkout to different branch to delete



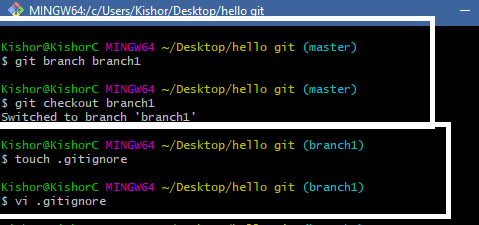


Summary:

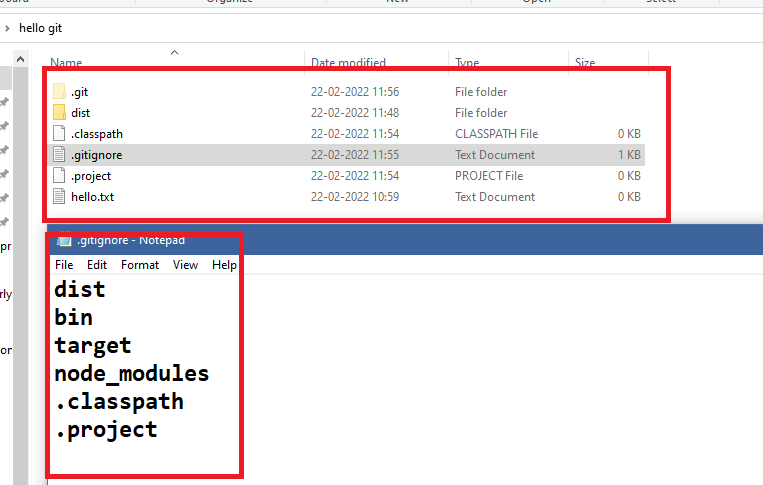
1. There are 2 ways you can create the local repository
2. using git clone remote-url
3. using git init & add remote url in the local repository
4. Important commands of Git
   1. git clone: creates a copy of remote repository in the local machine
   2. git init: creates a local repository
   3. git log: shows all the commits done with date, time, message & author information
   4. git status: shows all the tracked & untracked updates
   5. git add: adds the update to the staging area
   6. git commit: commits the updates in the staging area
   7. git push: pushes the branch to the remote from the local repo
   8. git pull: pulls the branch from the remote to the local repo
   9. git merge: merges the branch with another checked out branch

.gitignore:

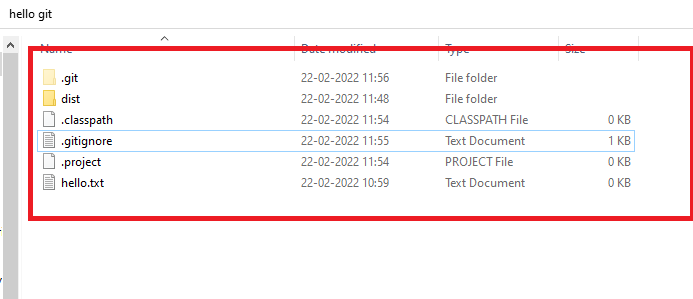
It is a file used by git to ignore the files/folders to be tracked.



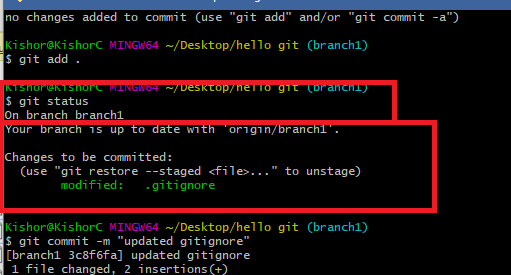
Add the content to .gitignore as mentioned below



Create some folders and files like dist, .classpath, .project and so on in the local repository

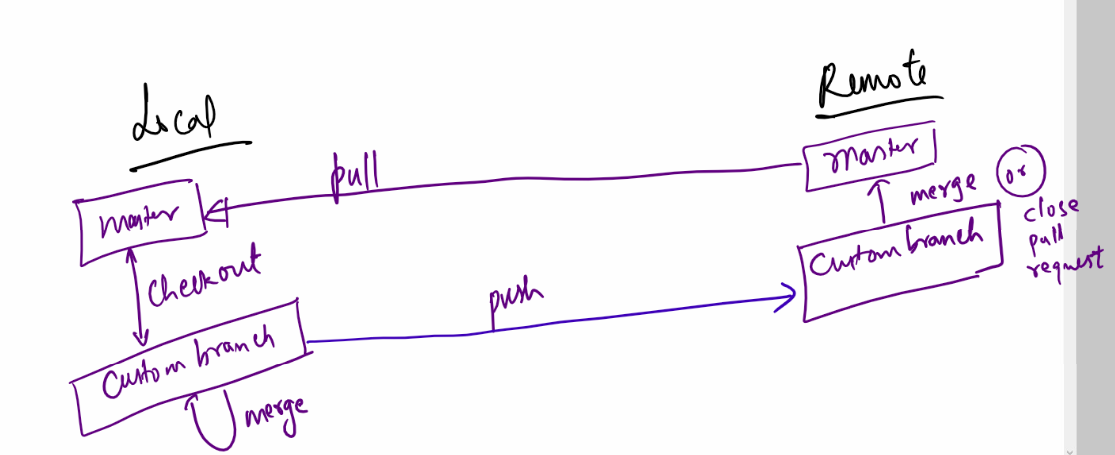


If you use git status you can see dist, .project & .classpath wouldn’t be tracked because its mentioned in gitignore



This .gitignore will be pushed to the remote, so that other developers would also get this file, it means everyone will have .gitignore file in their local machine, but the list of files/folders mentioned in the .gitignore wouldn’t be tracked at all from any machine.

Git Cycle



Git Organization:

It allows group of people to collaborate their work, only the members within group can update the repository inside the organization.

Steps:

1. Create the organization
2. Name the organization & fill all the details
3. Add the members by entering their username of git
4. Select the required options what you are going to with the organization, like manage code, collaborate work, team-size and so on
5. Change the role of members to owner.
6. Create one repository which all these members can access.

Activity

1. Create a team of 5 members (Trainer will do it)
2. Choose a team leader (Trainer will do it)
3. Team leader must create organization
4. Team leader will add members & members will accept the invitation
5. Team leader will change each members role
6. Team leader will create a repository
7. Team leader will create a file called hello.txt & pushes the master to the remote
8. Everyone in the team including the team leader edits hello.txt with their names and pushes their custom branch
9. Team leader will merge their custom branch or close the pull request depending on the scenario
10. Someone in the team will create another file welcome.txt and each members should pull that file and write their names again in welcome.txt and push that update to the remote, team leader will again merge or close the pull request

Summary:

* Git is a centralized version controlling system
* Git will have two types of repositories - local & remote
* You can initialize the git repository either using git clone or git init
* Fork is used to clone the repository at the remote/server side
* You must always work in custom branch
* Important GIT commands:
  + clone
  + init
  + push
  + pull
  + add
  + status
  + log
  + merge
  + branch
  + checkout
  + commit
* touch, vi, ls, mkdir, cd, these are unix related commands, you use this without git command

ex: ls

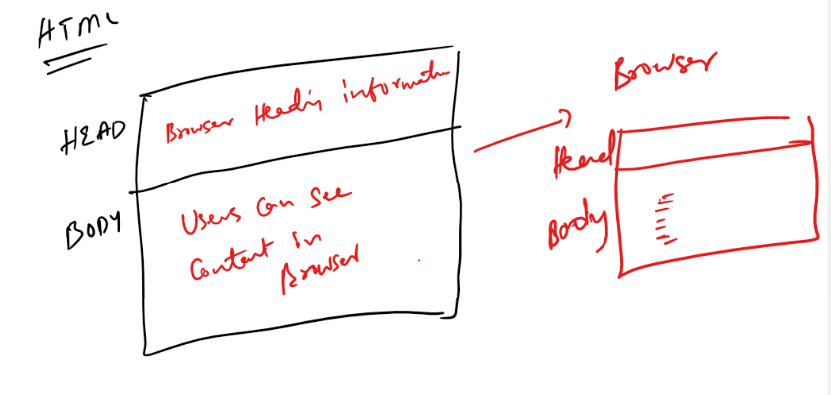
HTML:

It stands for Hyper Text Markup Language, it is mainly used to create websites and show the contents in the websites.

HTML uses tags which are called as markup to display the content, these content can be a text, a table, an image, a form, heading and so on

HTML as lot of predefined tags, these tags can be understood by browser

Note: HTML is the default language the browser can understand, along with HTML browser can understand CSS & Javascript



Softwares required

1. Editor: Notepad or VS Code
2. Browser: To render/display HTML content.

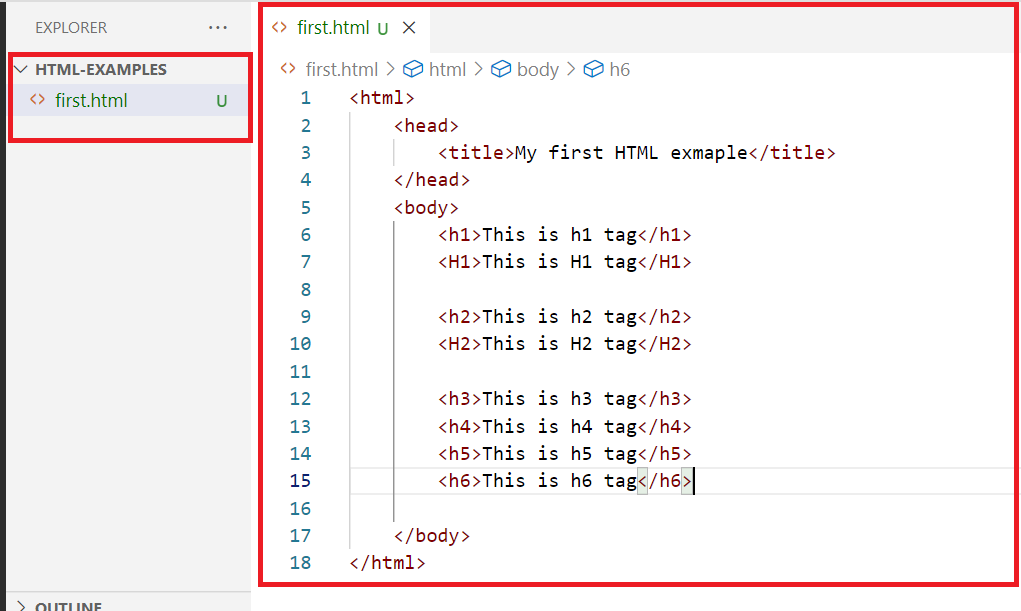
HTML has lot of inbuilt tags, all these tags is understood by browser, some of the tags are:

html, head, title, body, h1, h2, h3, h4, h5, h6, p, table, form, div, span, input, b, i, a, img, br, hr, select, option, video, audio, canvas, article, header, footer, section, nav, aside, mark, pre, marquee, sub, sup, script, style, meta, link, ol, ul, li,

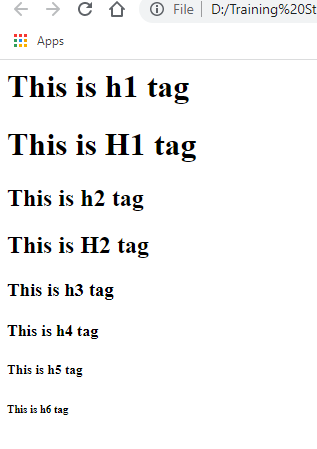
Note:

1. HTML doesn’t show any error if there’s a mistake
2. HTML is case insensitive i.e., if you have a <body> then you can close using </Body>
3. But most preferred case is lower case

first.html

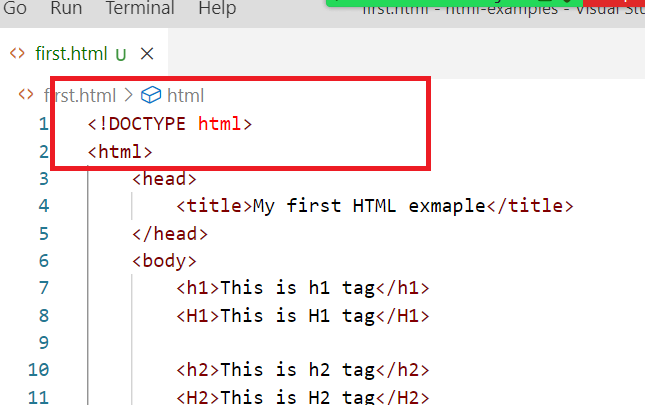


Output:

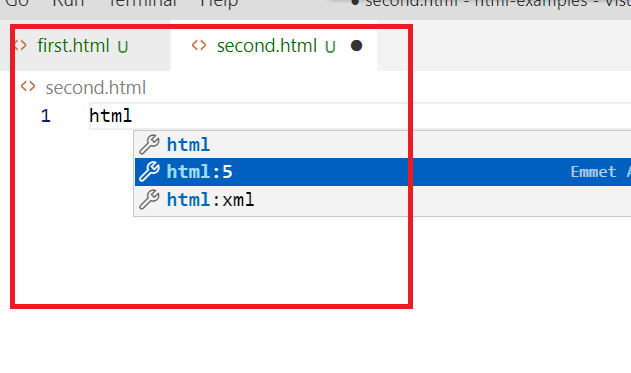


<!Doctype html>

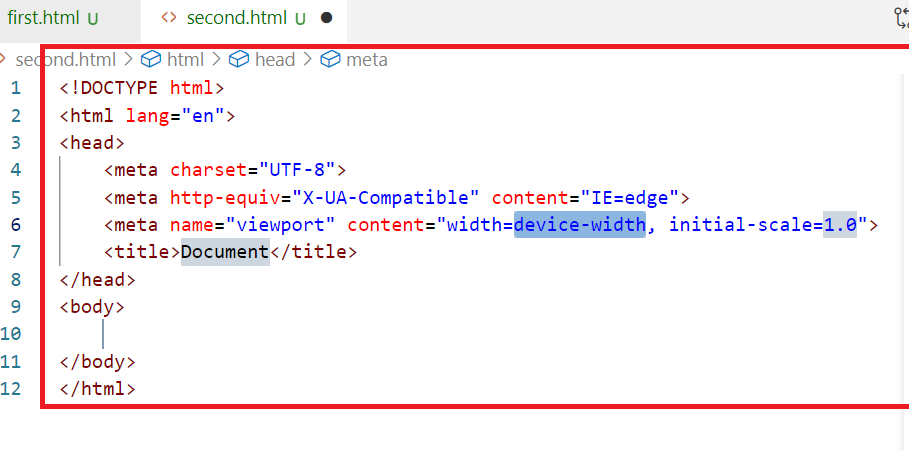
This declaration is recommended so that browser will understand immediately what document its loading.



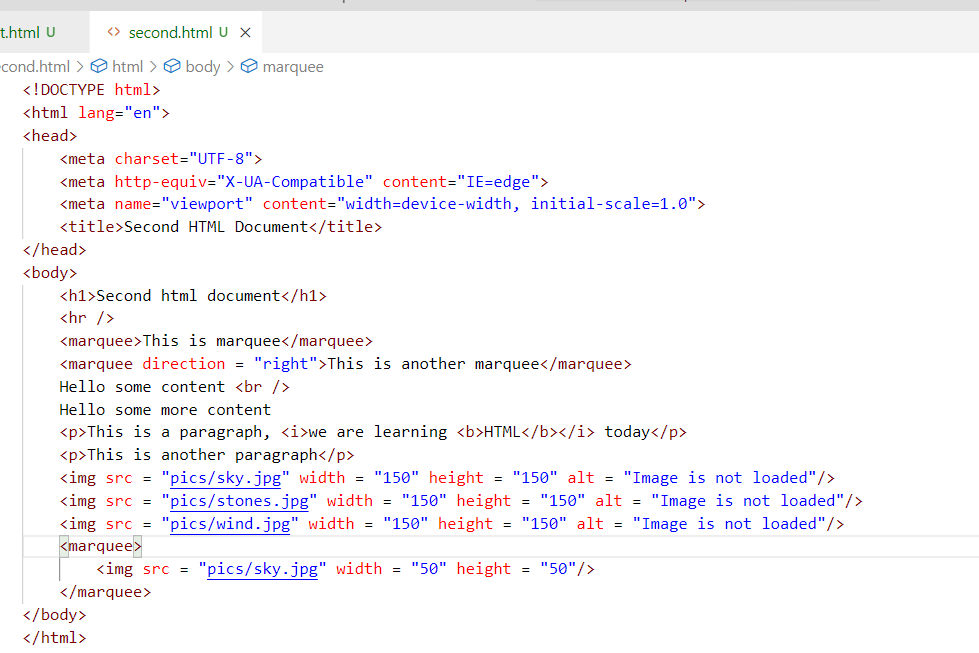
Note: In VSCode you can type html, to automatically have html template



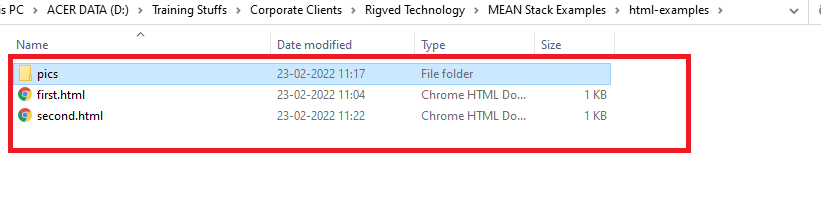
Once you select html: 5 you will get the below template



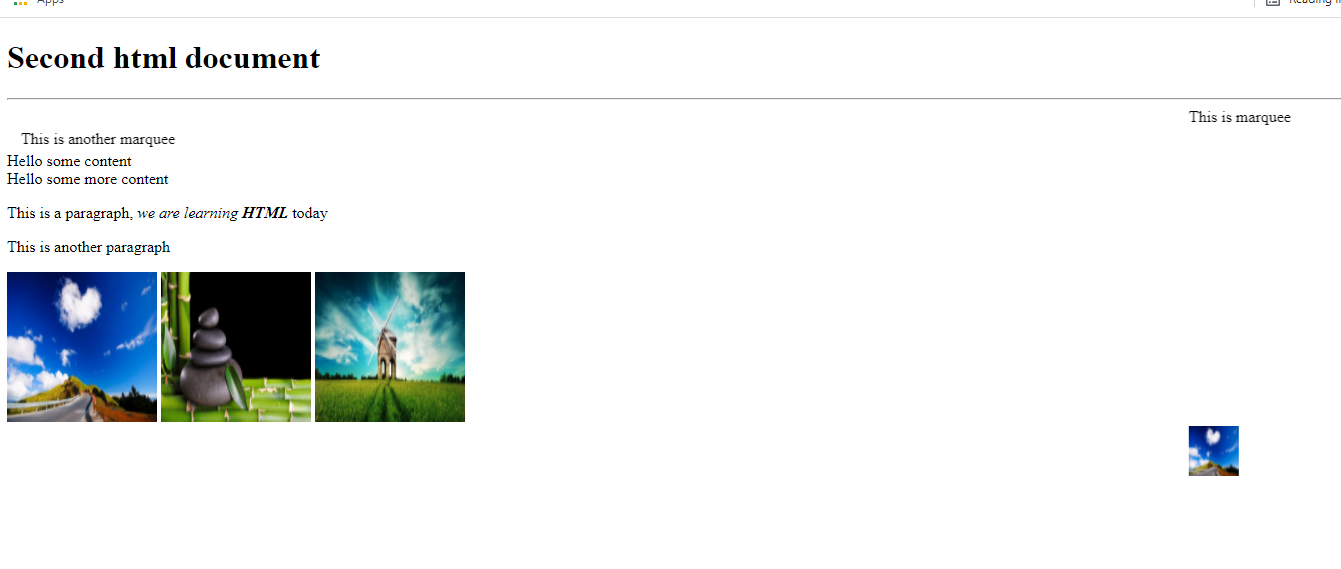
Understanding <p>, <b>, <i>,<marquee> <img>, <br>, <hr>



Note: There is a pics folder created in the location where we have html files

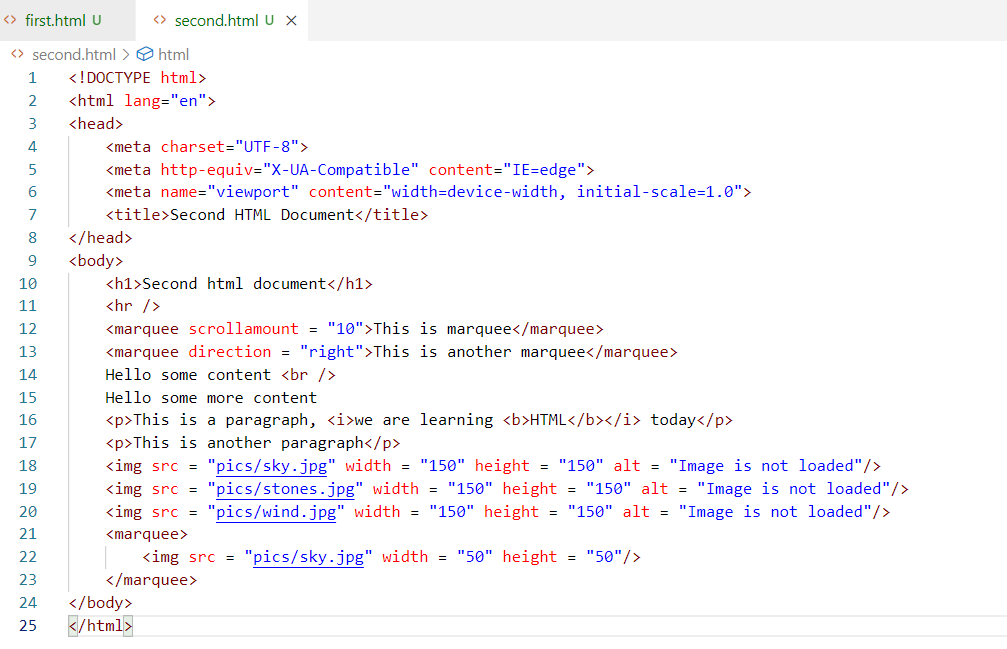


Output:

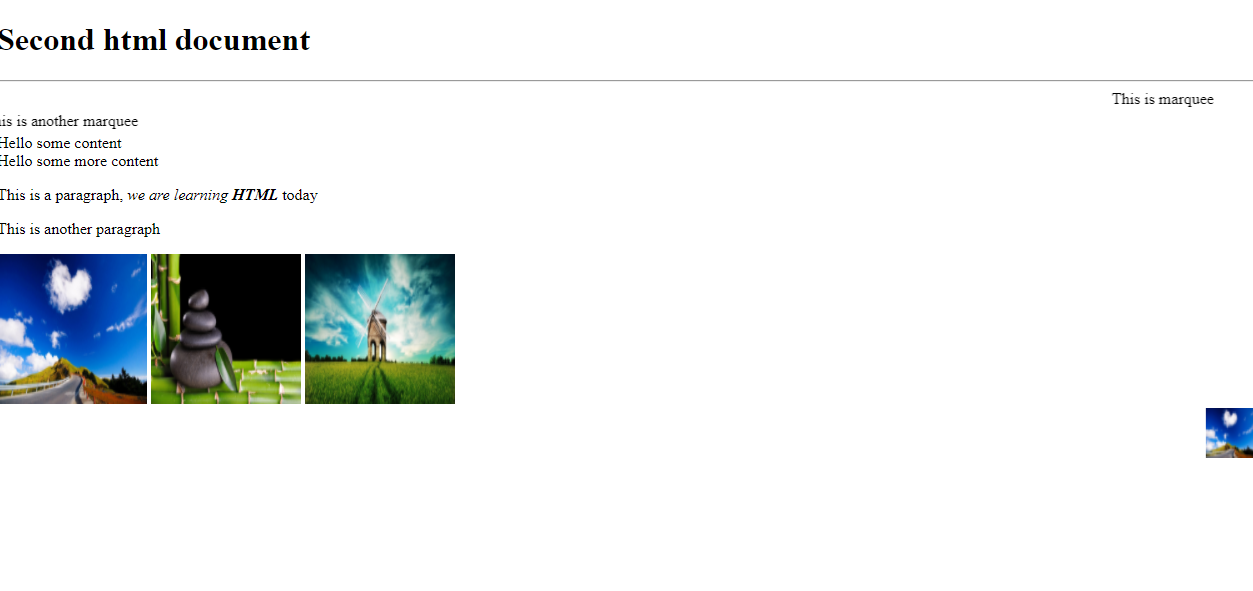


<marquee scrollamount = “10”> increases the speed, 10 is a number if you give greater numbers, the speed will be more

second.html



Output:



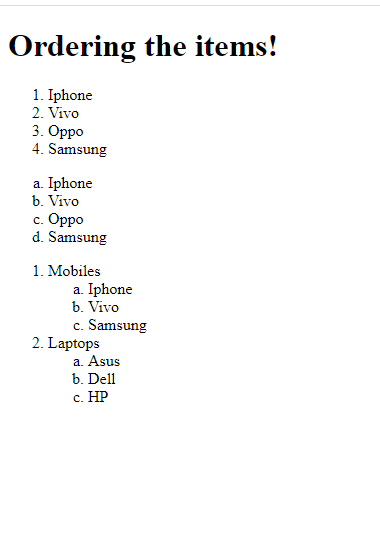
Ordering the items

We can create list of items using <ol> and <ul> both will use <li> to list the items.

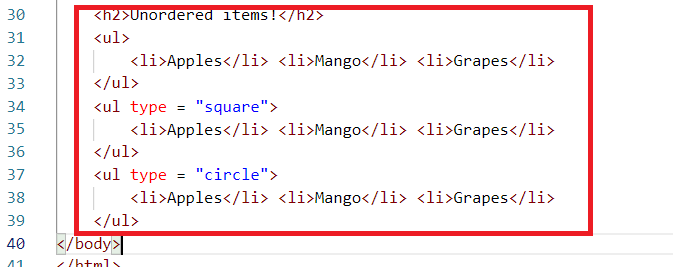
third.html



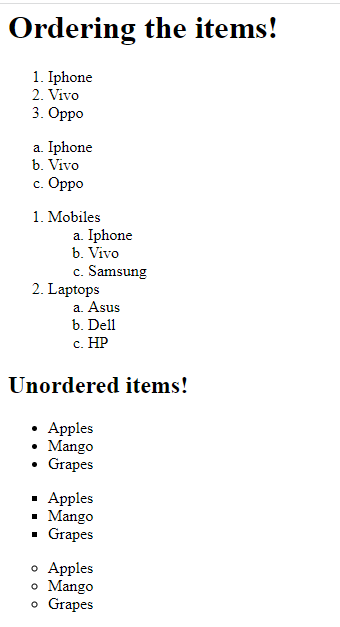
Output:



Unordered items: They will use bullets instead of numbers or alphabets



Output:



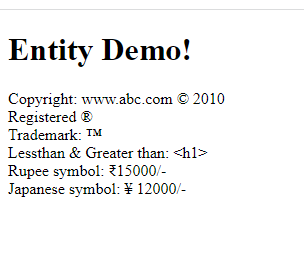
Entities in HTML

These are some special characters that you want to show to the user, like registered symbol ®, copyright symbol ©, ™ trade mark, Rupee symbol, greater than, less than and so on.

To create entities you must start with &entityName; that displays the entity on the browser



Output:



Some more entities:

&quot; For quotation

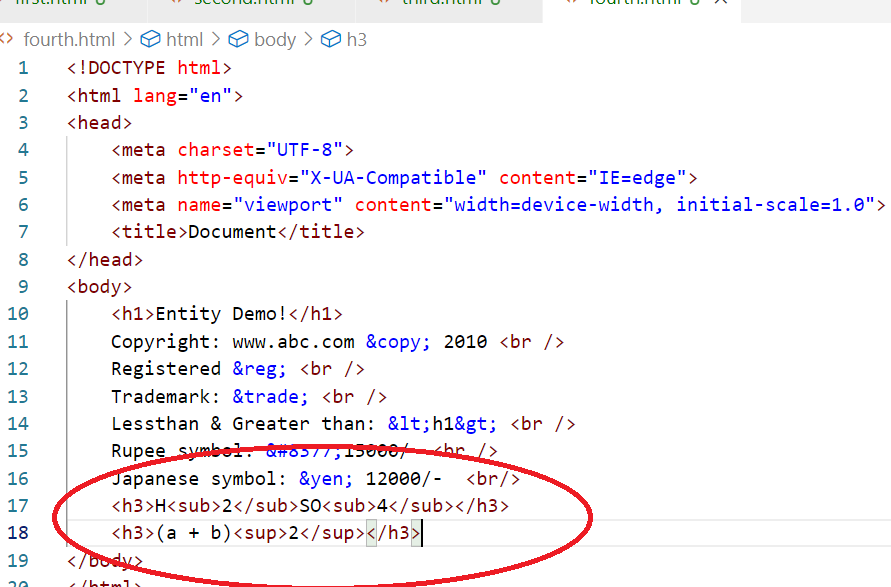
&amp; For &

&nbsp; For a space

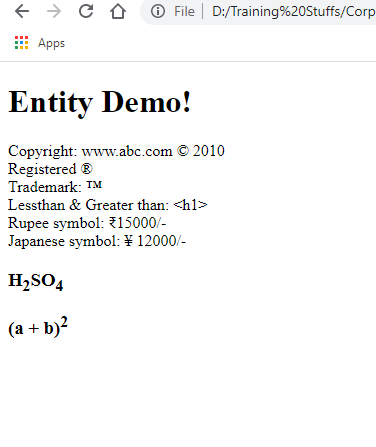
Super script & Sub script

These are used to show the content in a different position beside the normal position.

Suppose you want to show a chemical formula or mathematical formulas, these super script & subscript would be helpful.

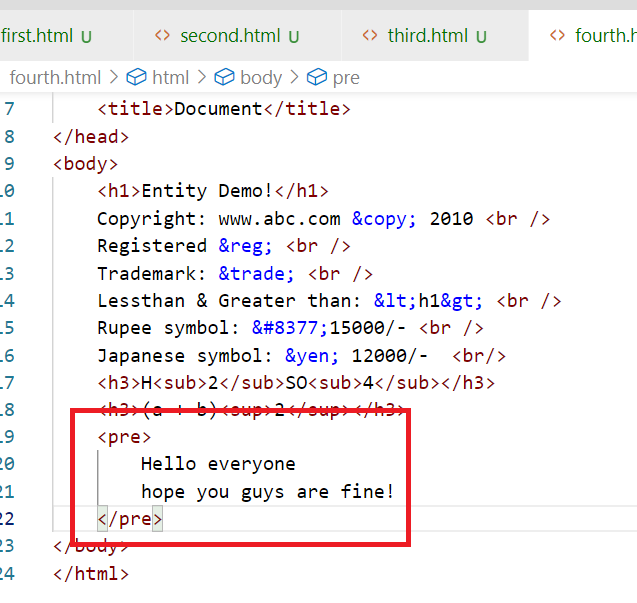


Output:



<pre>

It is used to show the contents in a way you specify in the HTML document



Output:

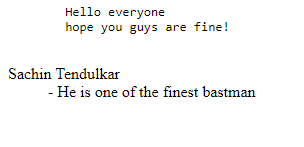


Data list

It is used to create some content with some brief description

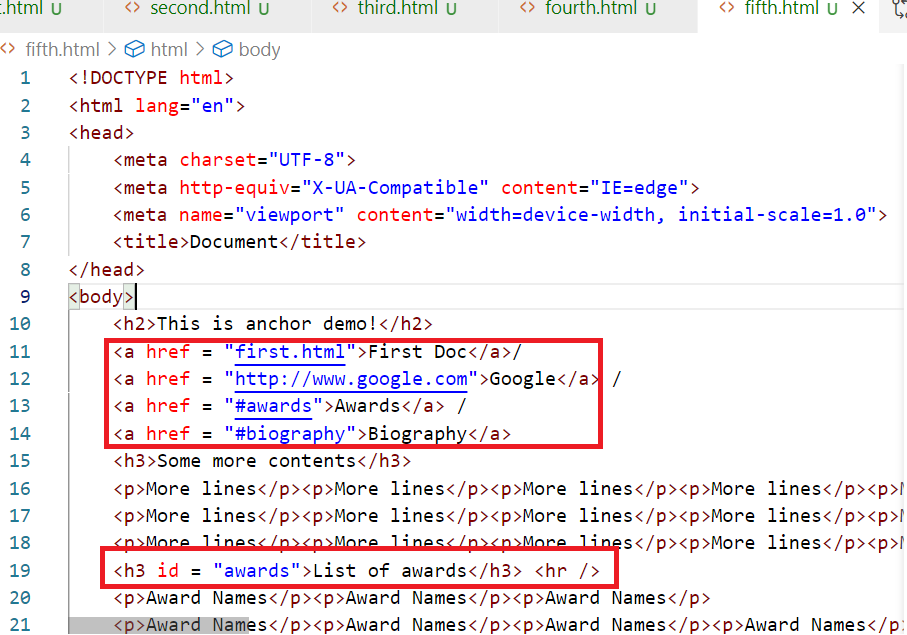


Output:



<a>anchor tag:

It is used to hyperlinks to different websites or server side resources



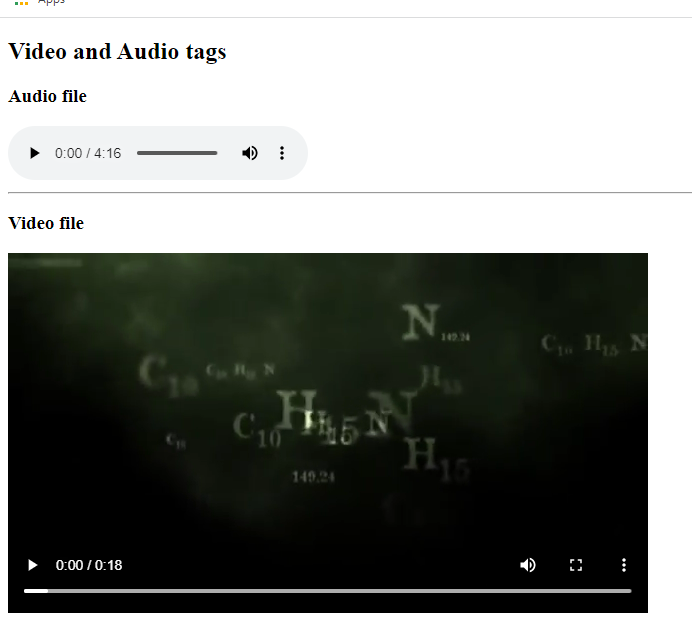
Here <a href = “#awards”> refers to the resource in the same page, the resource must have the id attribute i.e., <h3 id = “awards”> similarly you can have another <h3 id = “biography”> and refer it by <a href = “#biography”>

<video> & <audio>

You can have media tags in the HTML to render video’s are audio’s



Output:



Note: You can also add links of different videos like youtube videos

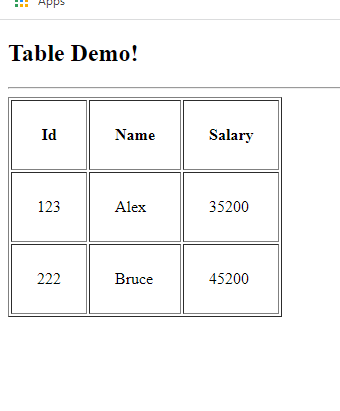
table:

It allows you to create tables with rows & columns, a table must have heading which specifies how many columns it can have and body that will have content of the table.

seventh.html



Output:



Activity:

1. Solve all the above examples discussed (this you don’t have to push to the git)
2. Fork the repository : <https://github.com/Kishor-C/rigved-students-activity.git>
3. Clone the rigved-students-activity repository created in your account
4. Create a folder like html-handson or html-examples and keep all the html related assignments in this folder,
5. Similarly for other technologies you will create separate folders, ex: for javascript, you will create javascript folder, for angular you will create angular folder and so on

Activity 1:

Create a table that will have product and its information, it must look as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Product Id | Name | Price | |
| Rs | Ps |
| 1122 | Laptops | 23500 | 35 |
| 3311 | Mobile | 10200 | 50 |
| 4422 | Watch | 5200 | 20 |
|  |  |  |  |

container tags

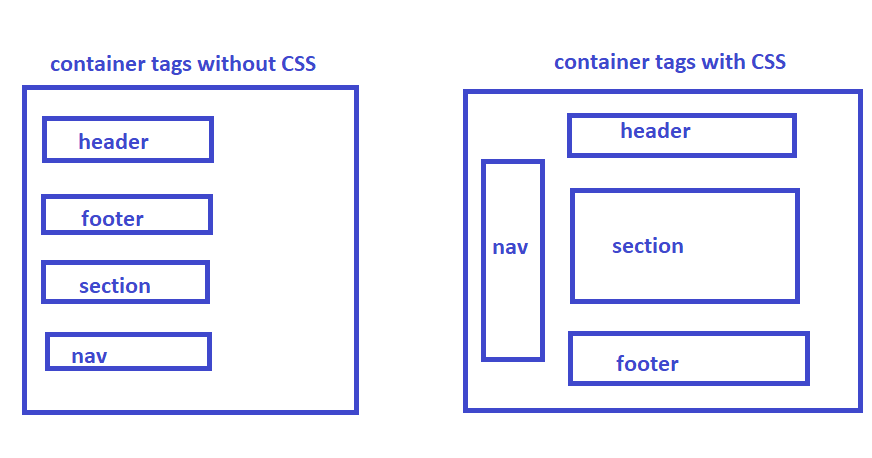
These tags can include other html elements, by default html gives you body tags which is a container tag, along with that HTML gives you <div> tag that can include other html elements.

Till HTML4 we had only <div> but HTML5 onwards more container tags were released which can be used to provide in a specific layout, like <header>, <footer>, <article>, <nav>, <section>, <aside>, all these have the same behaviour of <div> but they can categorise the HTML document to create a layout.

Note: These tags would create a layout only when you apply CSS, else they all would be in a default position where you keep in the HTML document.



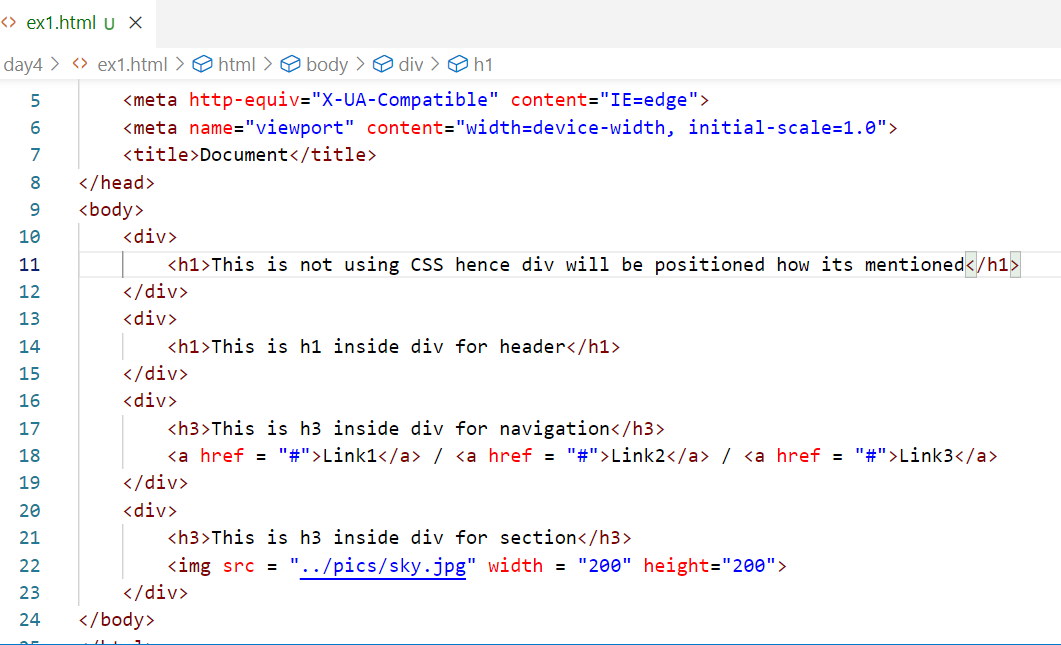
Since <div> doesn’t categorise the layout in the HTML document, HTML5 has released container tags that can be used to layout the document, but they will not have any default position until you apply css

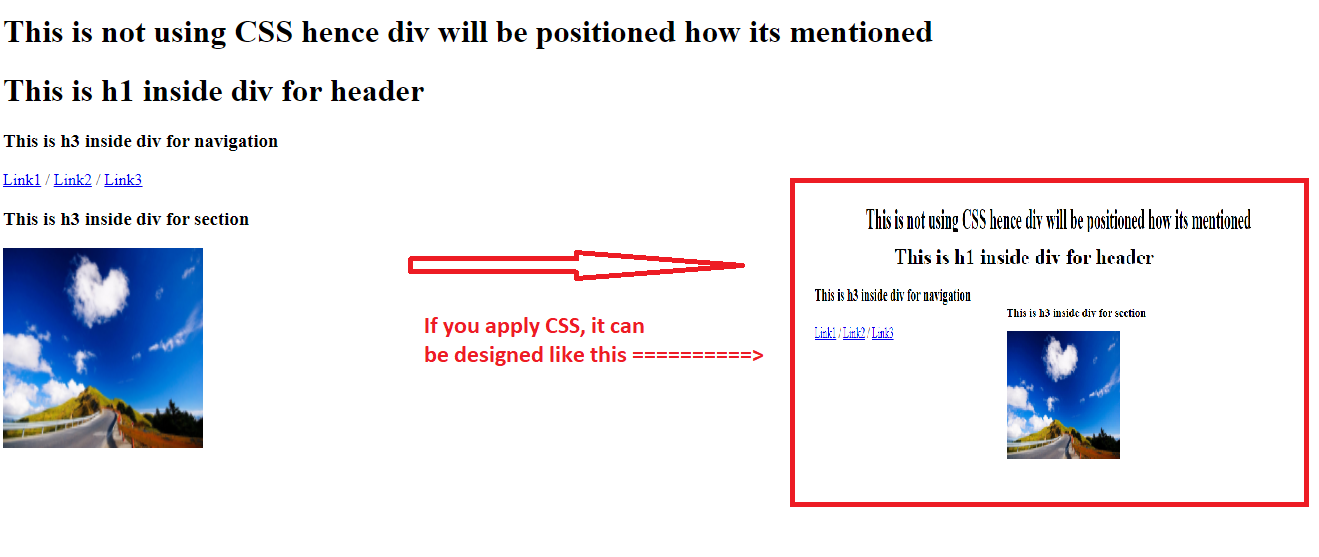


If you apply css to the container tags you can position it wherever you want, HTML doesn’t give any error for it, but you must follow the standard rule that header means it will be always at the top, footer means it will be always at the bottom,

Since <div> was the only tag earlier we had it doesn’t give any meaning to the users/developers, but header, footer, nav, aside, article, section, gives the meaning to the developers where they should appear.

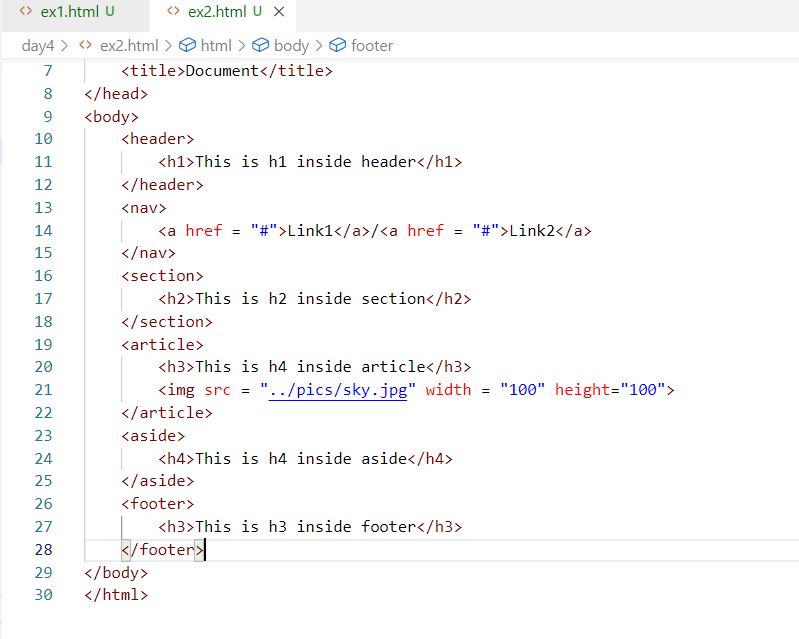
day4/ex1.html





Same code can also use <header>, <footer>, <nav>, <section>, <article>, <aside> but these tags must also use CSS to form a layout so that they can be positioned at the right place.

day4/ex2.html

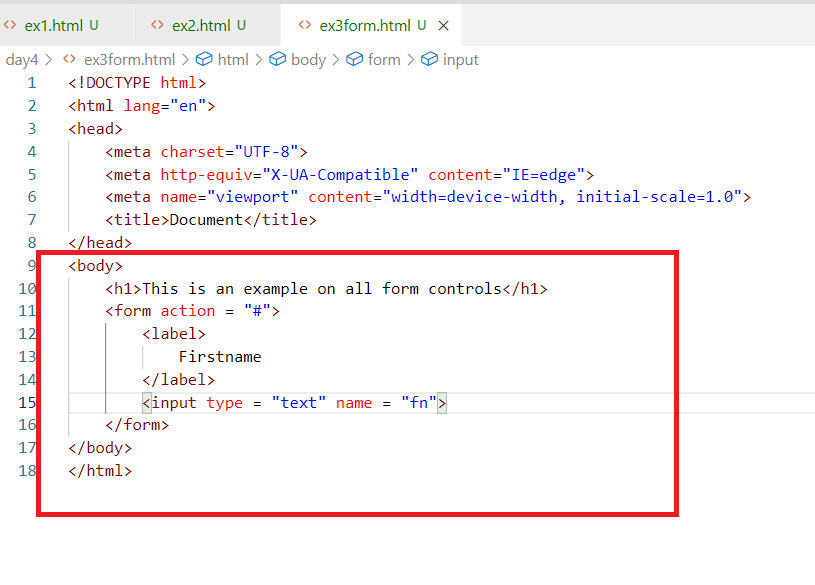


Output:



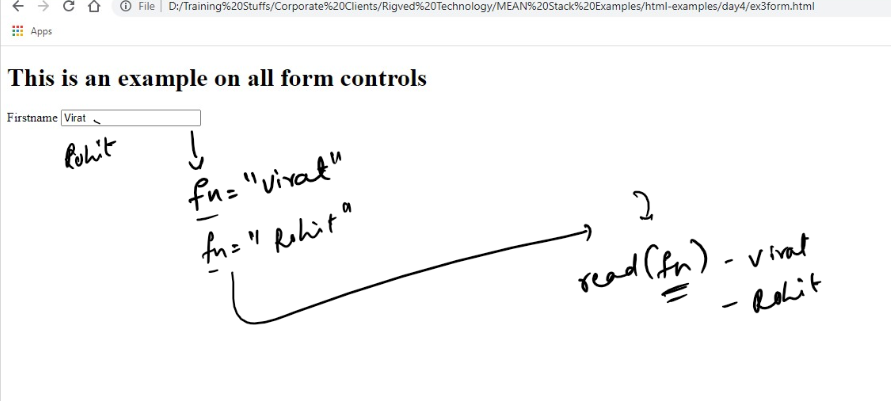
form tags

These are used to create input controls, where user can enter data, you can create input box, password box, radio button, check box, drop down, text area, file upload, buttons,

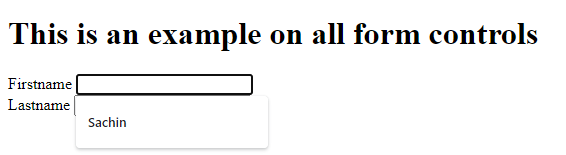


Here fn is a temporary variable that stores the value you enter in the input box

Output:



The below form shows the history of input you have entered, which might hide the elements below the input like below

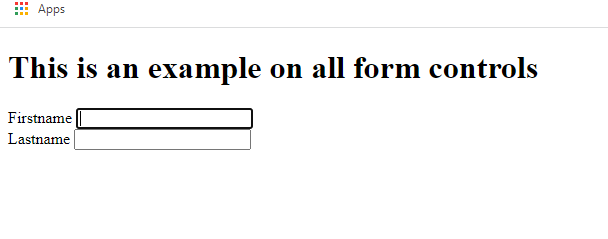


The above input shows some history of input you have entered, which would hide the below elements, to avoid that we can use autocomplete=”off”

ex3form.html

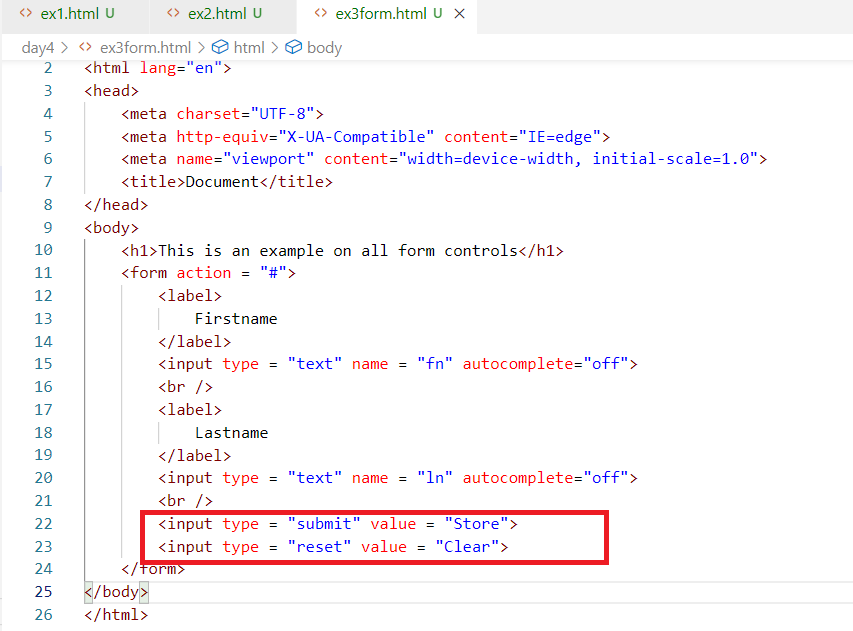


Output:

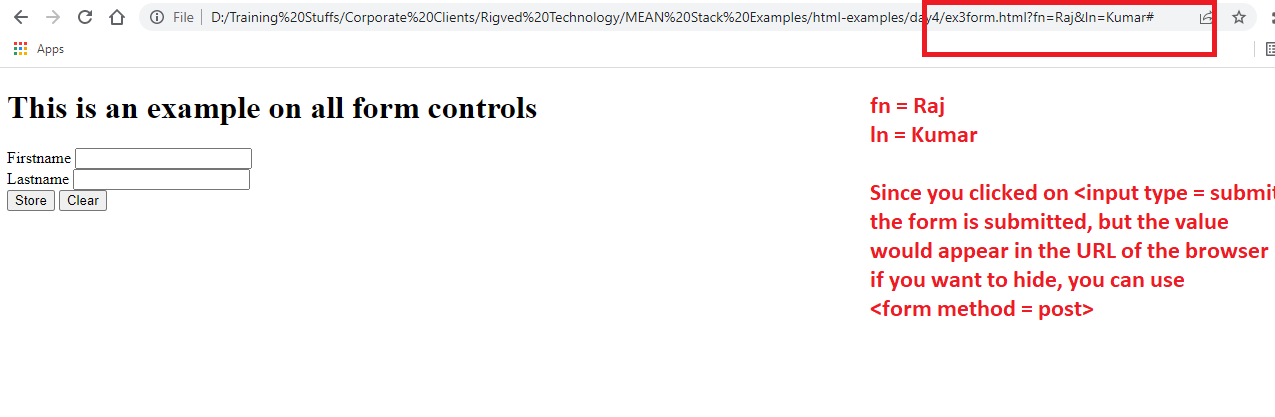


Since these input can’t be submitted, we can create buttons like submit, reset that can control the form input

ex3form.html



Output:

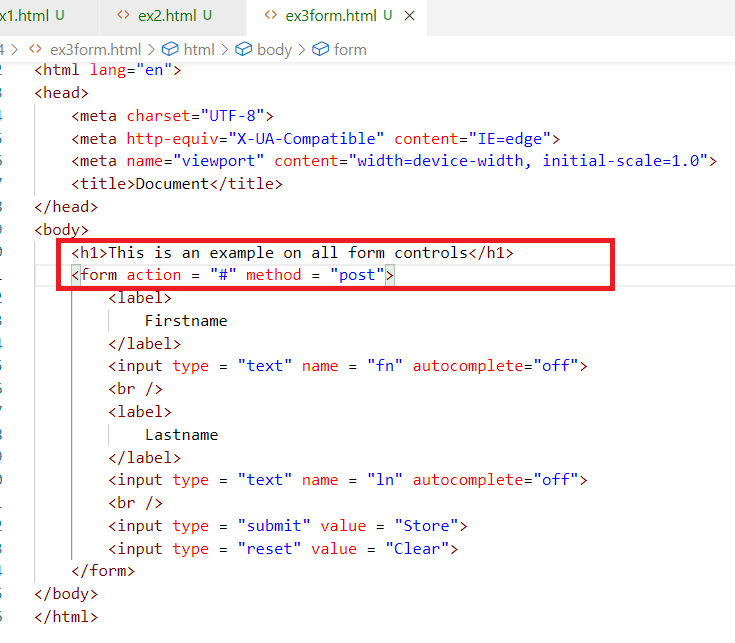


Most of the times form data must not be visible, hence you must use one attribute called method which can take values like GET & POST.

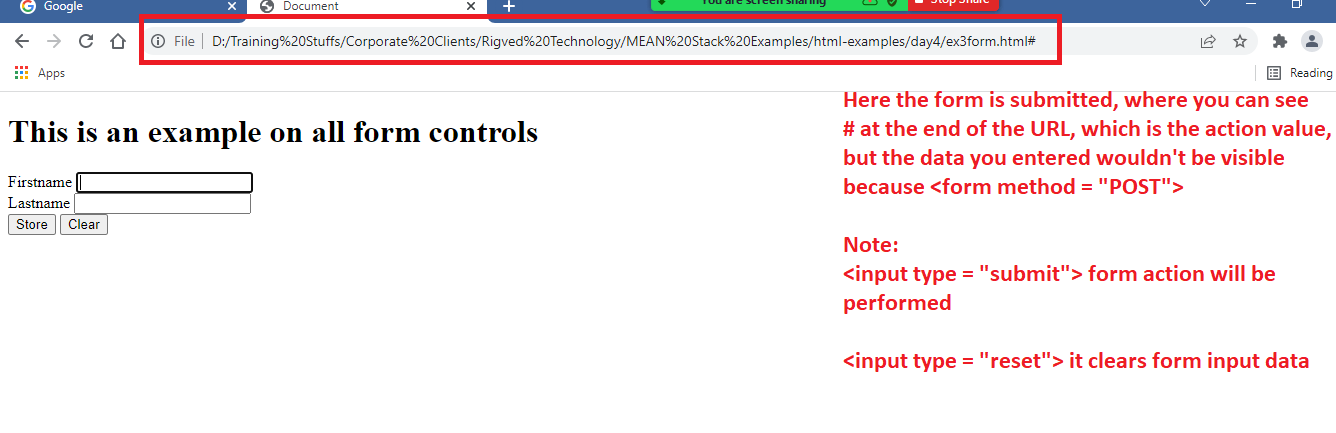
GET: This is the default behaviour of the form submission, which sends the data in the url

POST: This if used in the method, then the form data submitted, would be sent in the body of the document, not in the url, which wouldn’t be visible.

ex3form.html



Output:



The input attribute type can accept lot of keywords like text, submit, reset, it can use below list of keywords:

1. password
2. button
3. radio
4. checkbox
5. file
6. number
7. date
8. email
9. range

Note: Usually when the form submitted, a request is send to the server resource mentioned in the <form action = “url-pattern”>.

A request will be of two parts

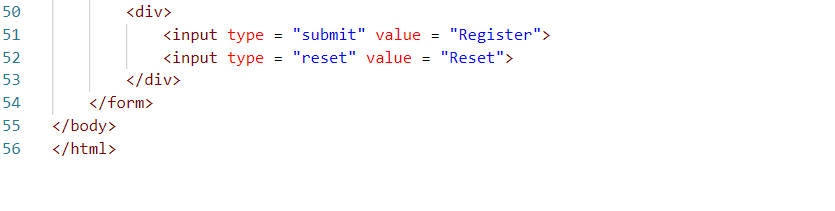
1. header: will have information’s like url & other properties of request like size of data, length of data, type of data
2. body: will have the data

Difference between GET and POST

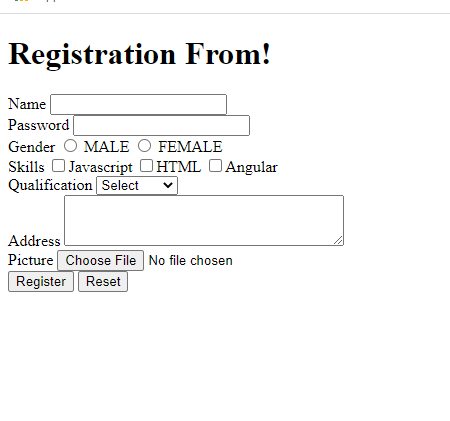
|  |  |
| --- | --- |
| GET | POST |
| This is the default method of <form> | This must be mentioned explicitly in the method of <form> |
| The data <form> submits appear in the URL, because the data is sent in the header part of the request | The data <form> submits doesn’t appear in the URL, but data will be sent in the body of the request |
| Maximum characters you can enter in the URL is 256 | No limits in the number of characters |
| Not Secure | Secured |
| Faster | Slower compare to GET, certain milliseconds or nanoseconds delay |

Creating other form controls like password, radio, checkbox, drop down, file upload and etc.

ex4regform.html

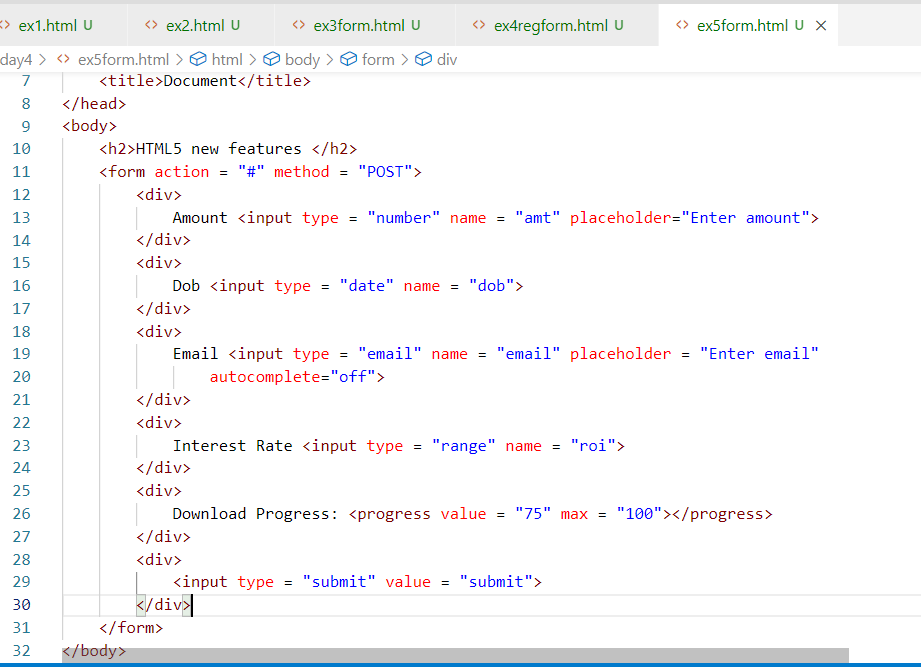


Output:



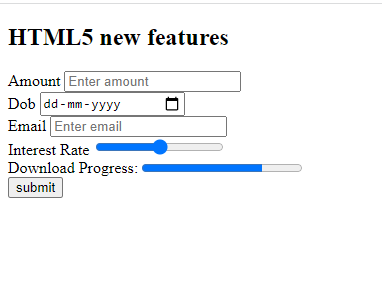
Other types of input controls: email, date, number, range, progress, placeholder - these were introduced in HTML5

day4/ex5form.html



The progress value / range value can be dynamically updated through Javascript

Output:



Actvity2:

1. Try all the above exercises discussed today
2. Create the same registration form with proper alignment using <table> tag

CSS: CSS stands for Cascading Style Sheet, it is mainly used to style the HTML document.

There are 3 types of CSS

* Inline CSS
* Internal CSS
* External CSS

CSS Syntax  
selector { property : value; property: value }

selector means it could be an element or group of element or a particular element id

Inline CSS: Styling the particular element using style attribute

ex: <p style = “color:red; font-family:arial”>SOME TEXT</p>

Internal CSS: You will have styles common to the entire HTML document, it is written using <style> tag inside <head>

ex:   
<style>   
 h1 { color : green; background-color : yellow }  
 .c1 { color : blue } /\* this is for multiple elements using class name c1\*/  
 #a { color : red } /\* this for a particular element having unique id “a” \*/  
</style>

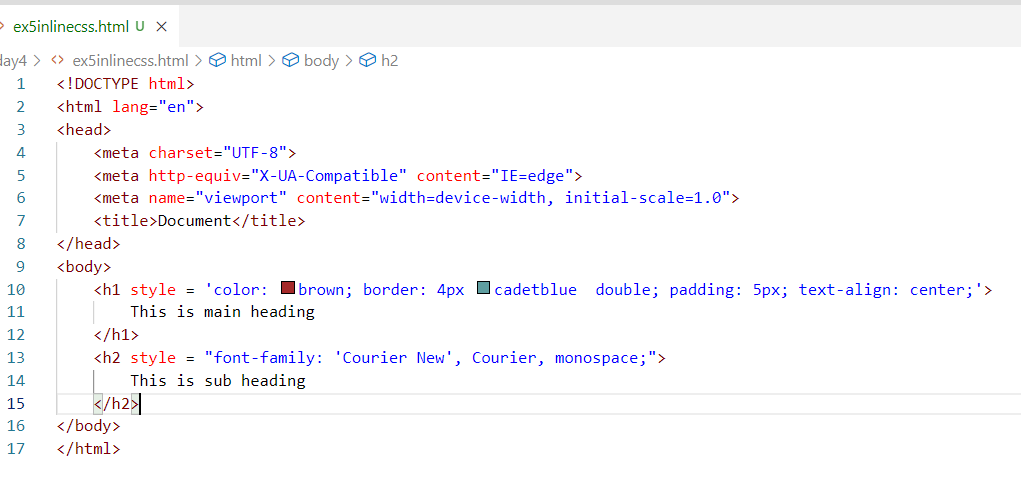
<h2 class = ‘c1’>SOME TEXT</h2>  
<h3 class = ‘c1’>SOME TEXT</h3>  
<b class = ‘c1’>SOME TEXT</b>  
<h1>SOME TEXT</h1>  
<h1>SOME TEXT</h1>

<h3 id = “a”>SOME TEXT</h3>

External CSS: It is an external css file that is linked to one or more HTML document, so that multiple HTML documents can have the styles written in that external CSS file

You need to use <link> tag to link the external css with html file.

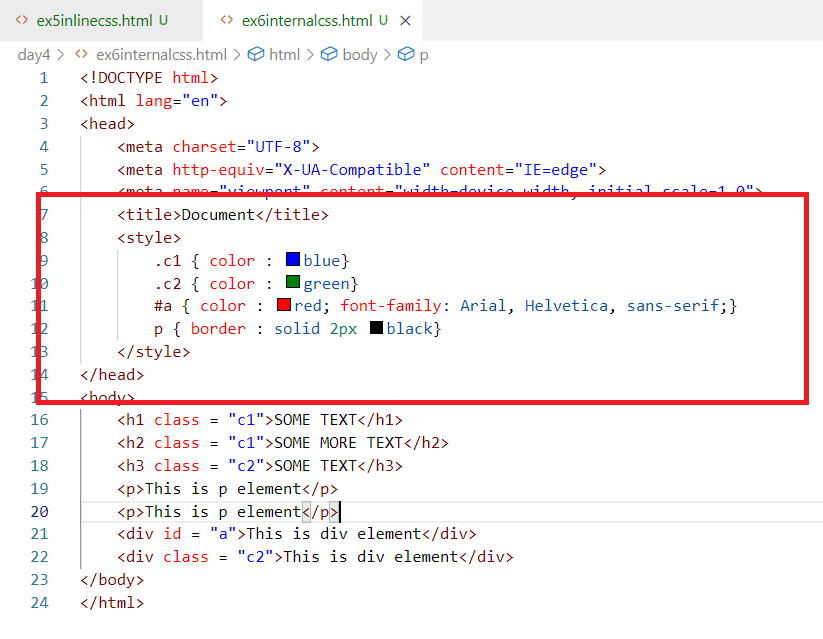
day4/ex5inlinecss.html



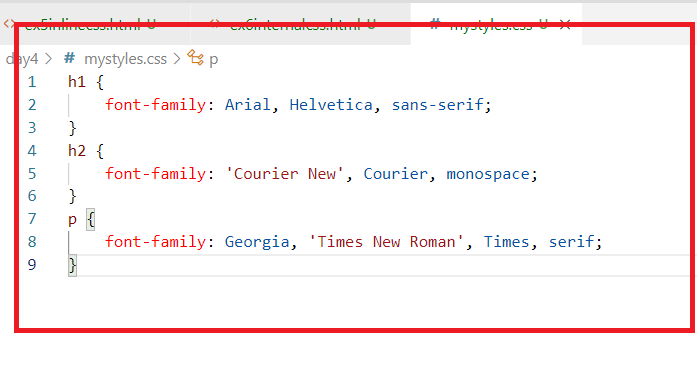
Output:



internalcss.html



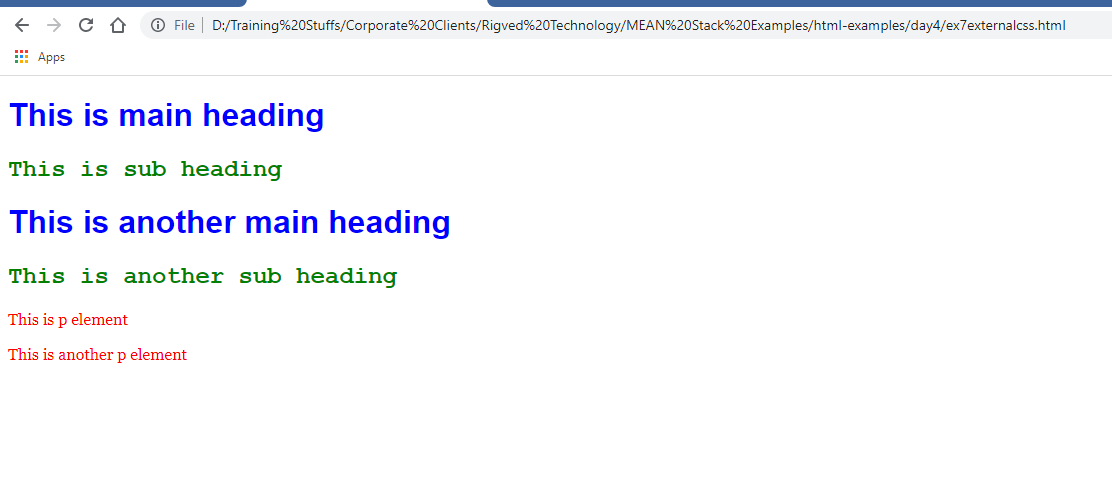
day4/mystyles.css



ex7externalcss.html



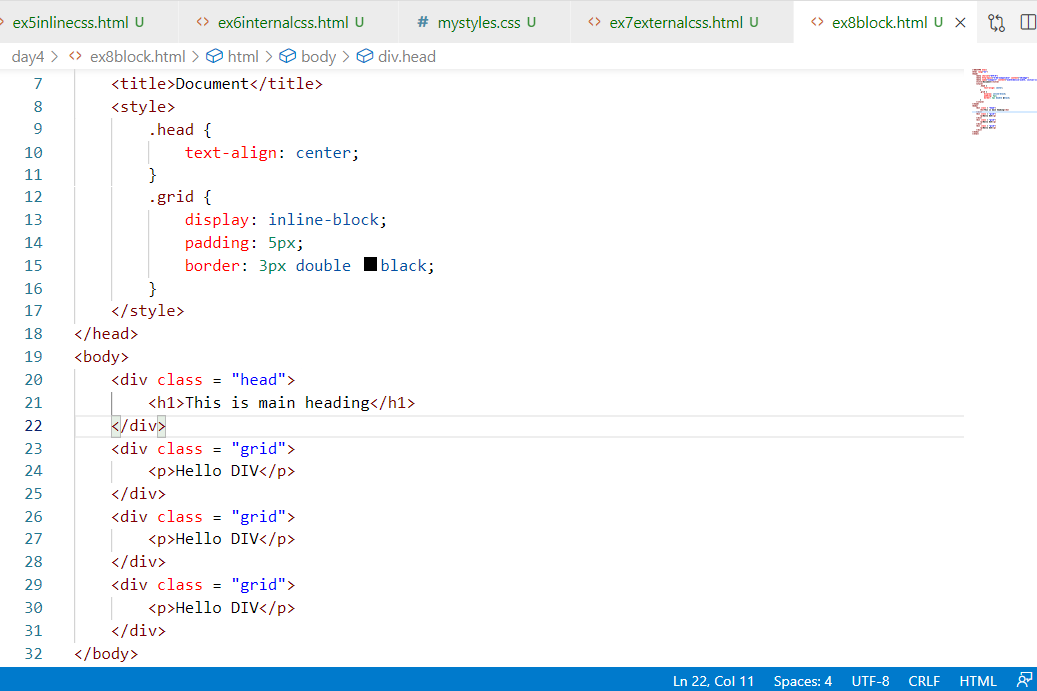
Output:



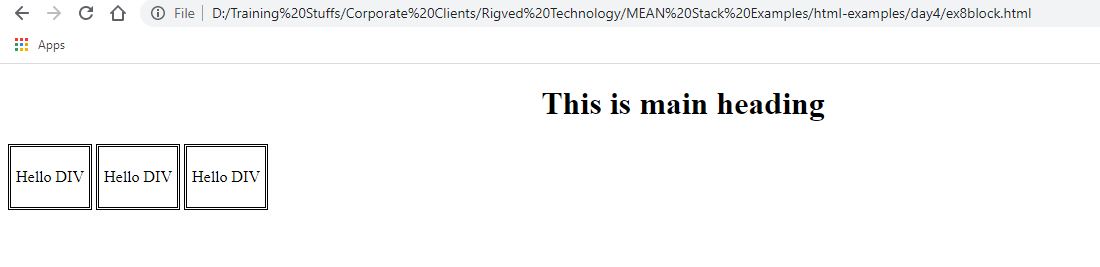
Keeping all the container tags in the same line

You can use a css property called display:inline, that doesn’t give line break to the container tags.

day4/ex8block.html



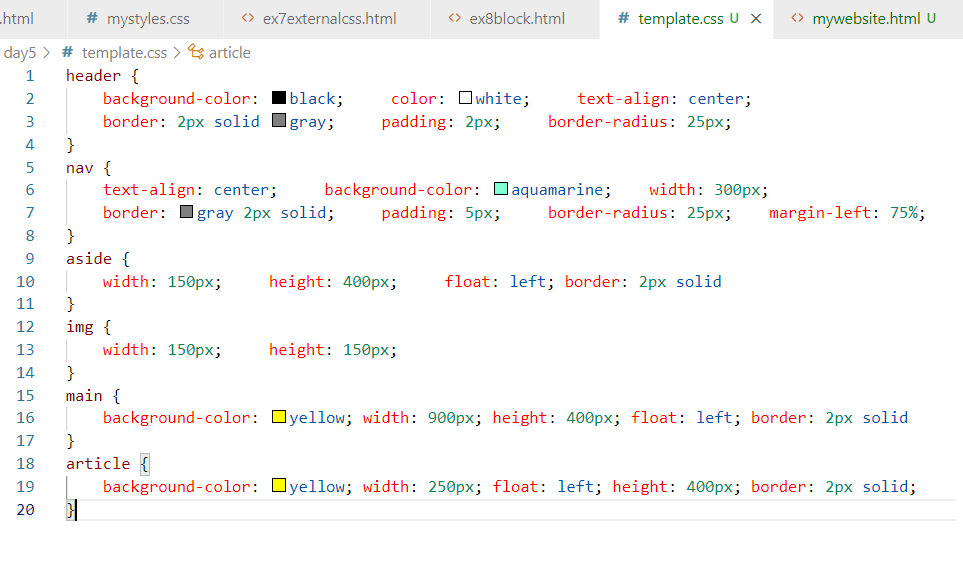
Output:



Activity:



day5/template.css



day5/mywebsite.html



Output:

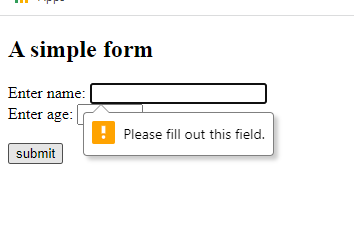


HTML 5 has provided some validation features

1. required
2. max, min
3. pattern
4. search type, tel type, number type and so on



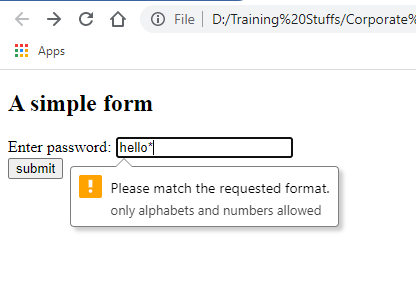
Output:



Pattern: This is an attribute which tests whether the input matches to the given regular expression



Output:



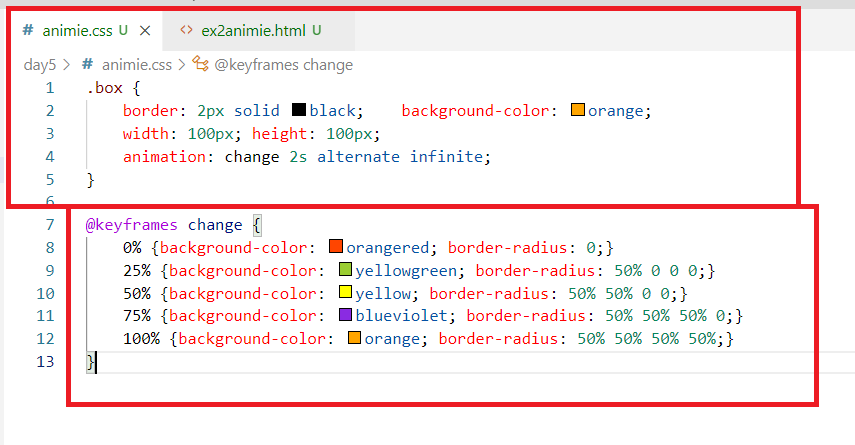
CSS3 Features

* Rounded corners
* Media Queries
* Transitions
* Transformations
* Animations
* Inheritance: Inheriting the other css properties in a css file

Animations:

It gives extra effect to the HTML elements, it is done using @keyframes to apply the animation rules that can specify the effects the HTML element must take within some duration.

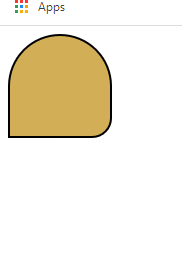
animie.css



day5/animie.html



Output:

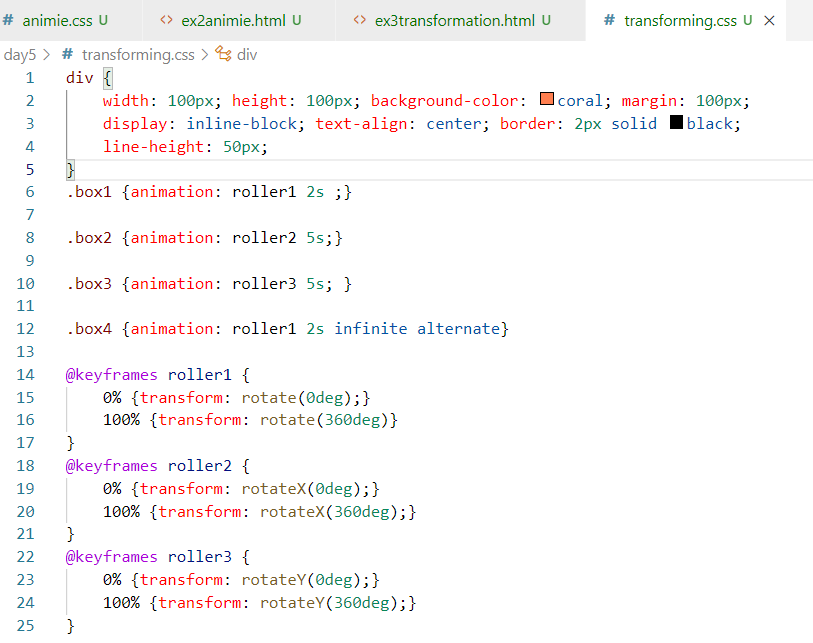


Transformation:

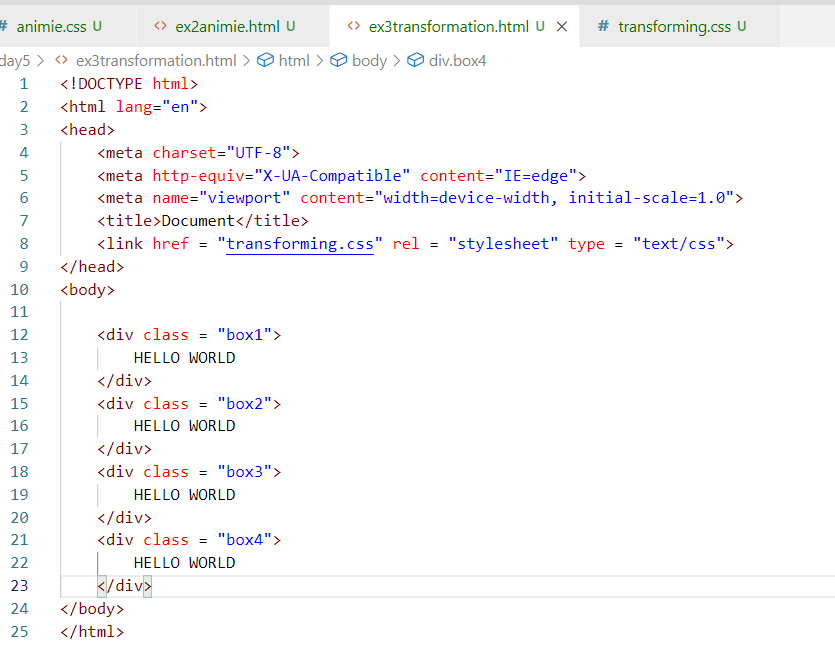
This is used to add extra properties to the animated element, where you can transform the HTML element direction like rotating in different axis

Ex: rotate(360deg), rotateX(360deg), rotateY(180deg)

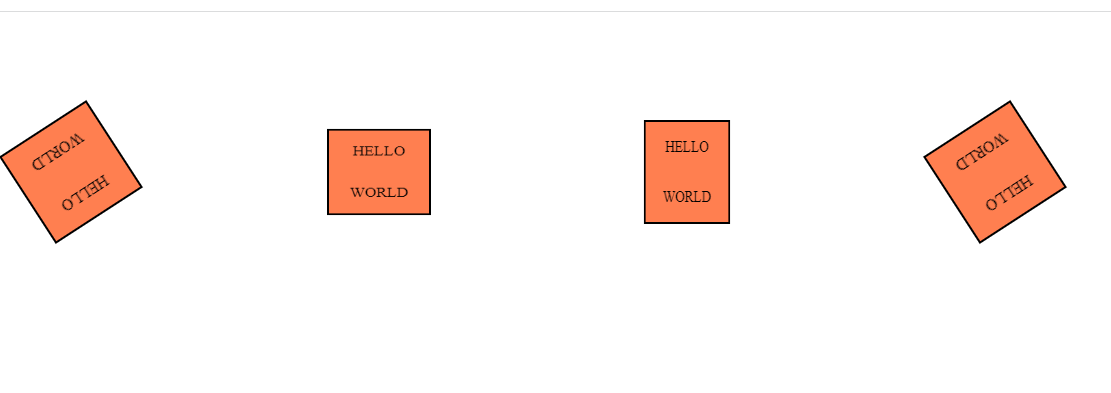
transforming.css



ex3transformation.html

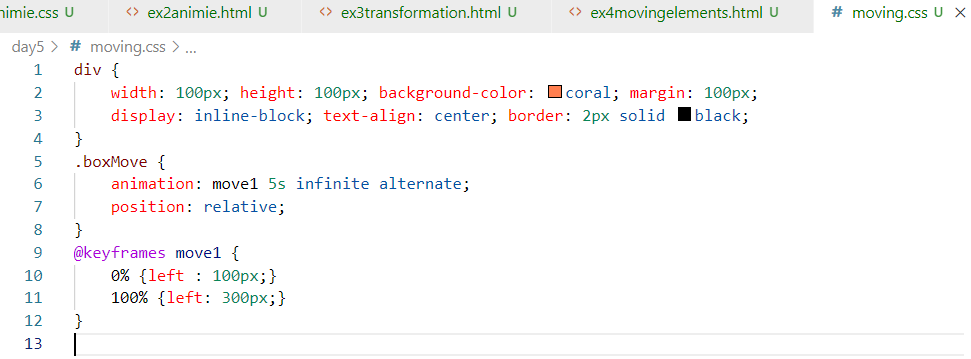


Output:



Moving the elements

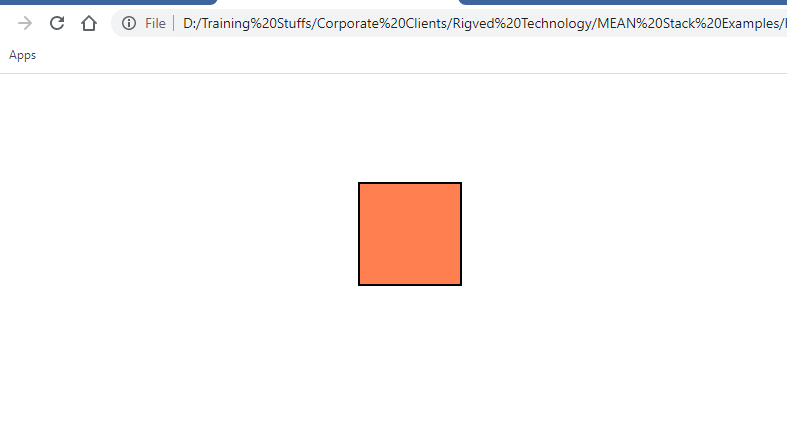
moving.css



ex4movingelements.html



Output:

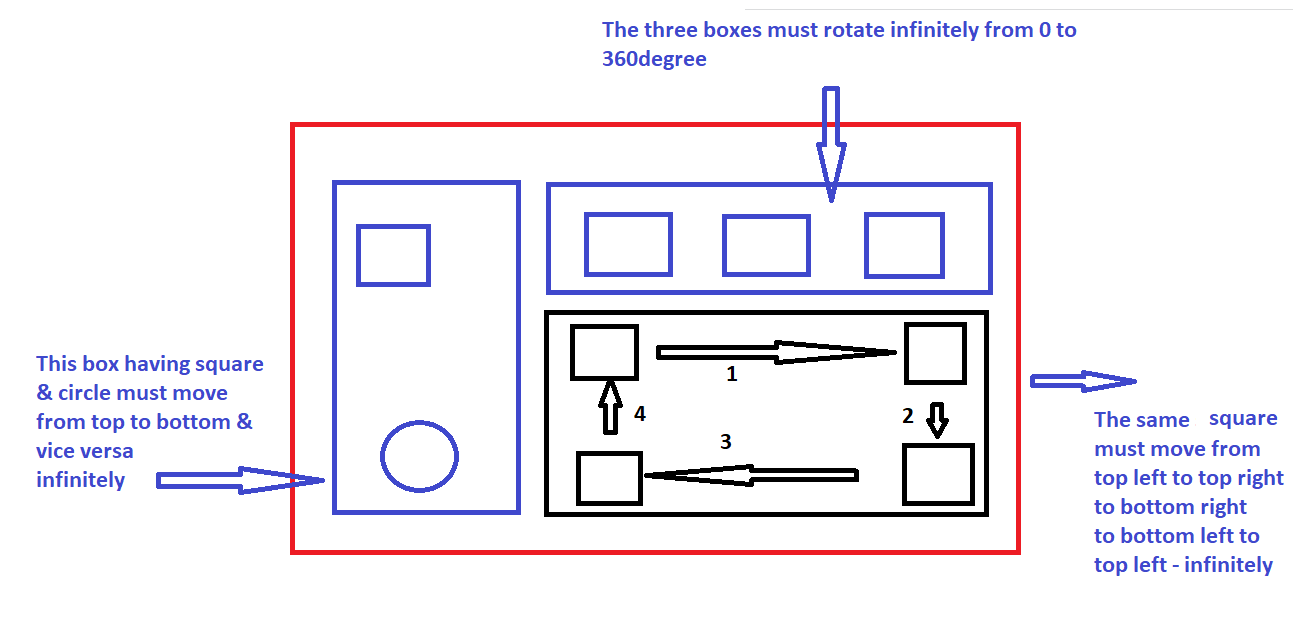


HTML 5 Features

1. Canvas element: This is to create 2d drawings, but we need to have Javascript knowledge
2. Drag & Drop feature
3. Web Storage: Where you can store the user data in the browser storage area, there are two types of web storage 1. Session Storage 2. Local Storage, here we need to use Javascript
4. Video & Audio tags
5. Different input types number, range, progress, placeholder, pattern, required, etc.
6. Details, Summary, datalist, output, header, footer, aside, main, nav, article, section, mark

Activity

1. Try the above exercises on html & css - animations, transformations, transitions
2. Work in a team to create a website that will have CSS features like animations, transformations, transitions, below is the activity result you must get



Here team must divide this task and create an external CSS and apply style to the single

HTML document, it means all the members work in a single HTML document, use GIT to take care of merging other person work

RWD: Responsive Web Design

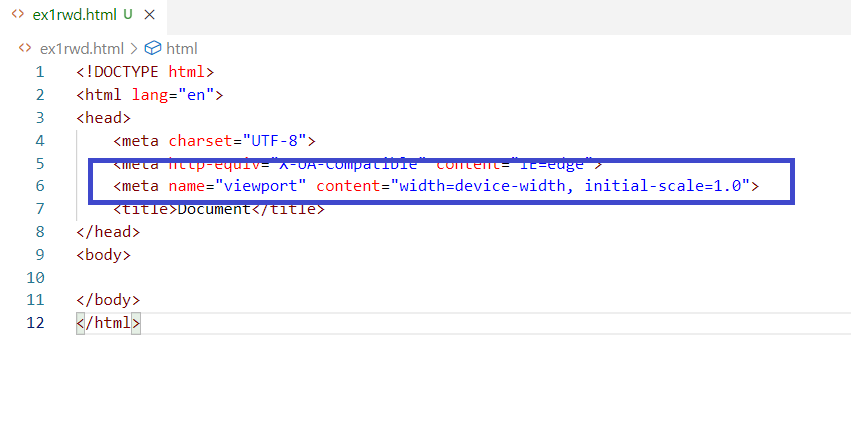
RWD makes your web page fit to all the types of devices could be desktop, mobiles, printing devices, tablets etc.

Since earlier web designer designed the web pages for desktops, the screen was fitting to the desktop screen, but when people started using mobiles, tablets the browser had to scale the web page to fit the device size, this was just a quick fix but not a permanent fix,

To fix this issue, HTML5 introduced a method to control the users visible area which is called as view port

The html:5 template in vs code will have <meta> tag with view port access.

ex1rwd.html



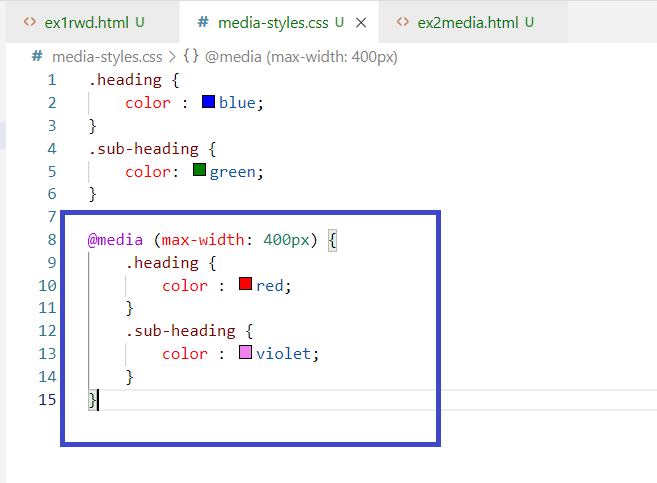
Here the <meta> sets the width of the page to the device width, the content would fit to the device width and the initial-scale is the zooming size is 1.0, now any content you have in the HTML would be rendered to any kind of devices without any manual scaling of the web pages in the browser.

Media Query:

It is a CSS technique to apply the block of CSS properties when some condition is true.

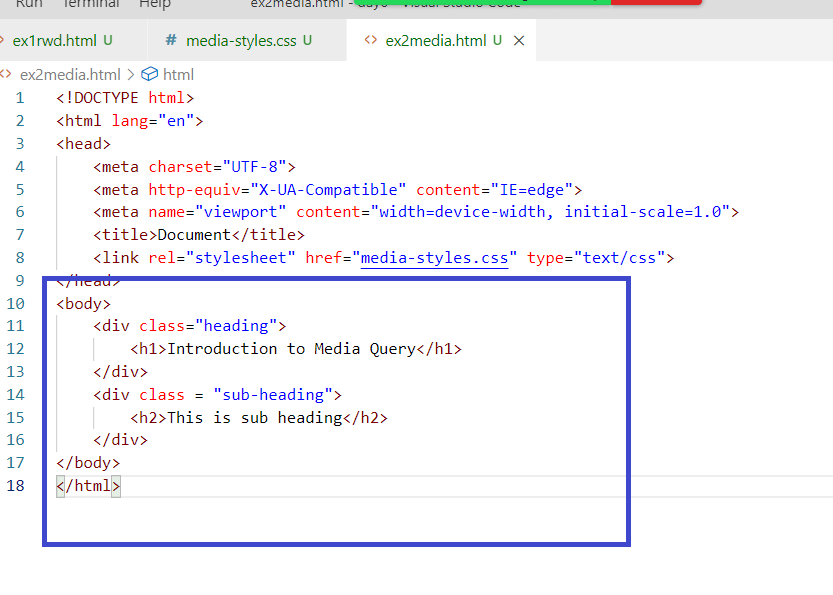
It is used by @media with some rules this will have CSS properties.

media-styles.css

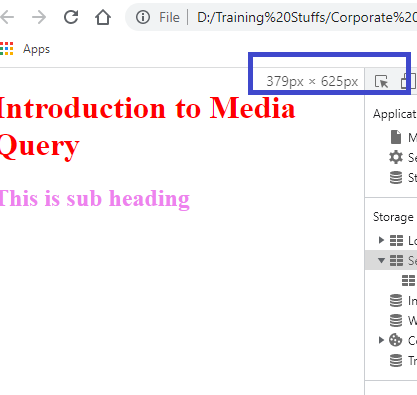


Once the width of any device is <= 400px then the .heading & .sub-heading color would change.

ex2media.html



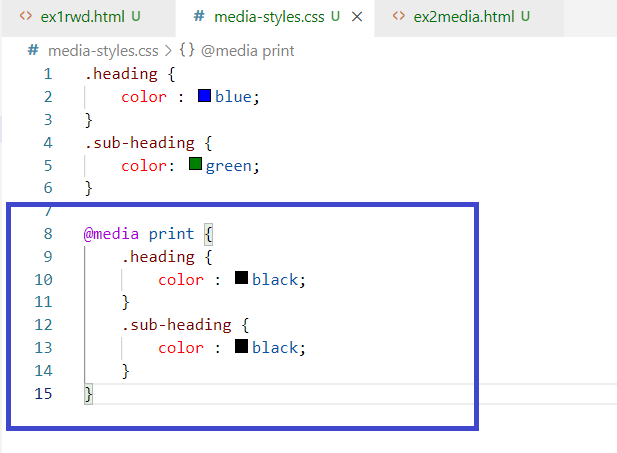
Output:



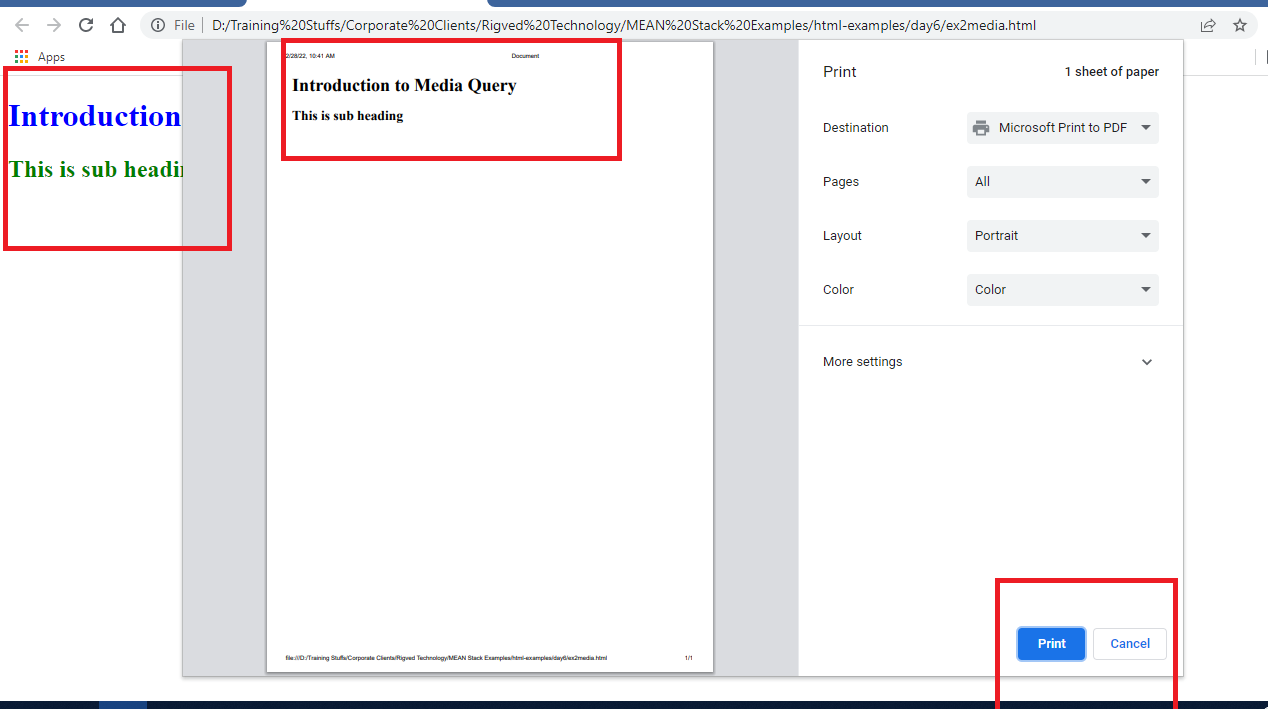
If the width is above 400px



Suppose you want to change the style while taking the print you can use the @media for print devices

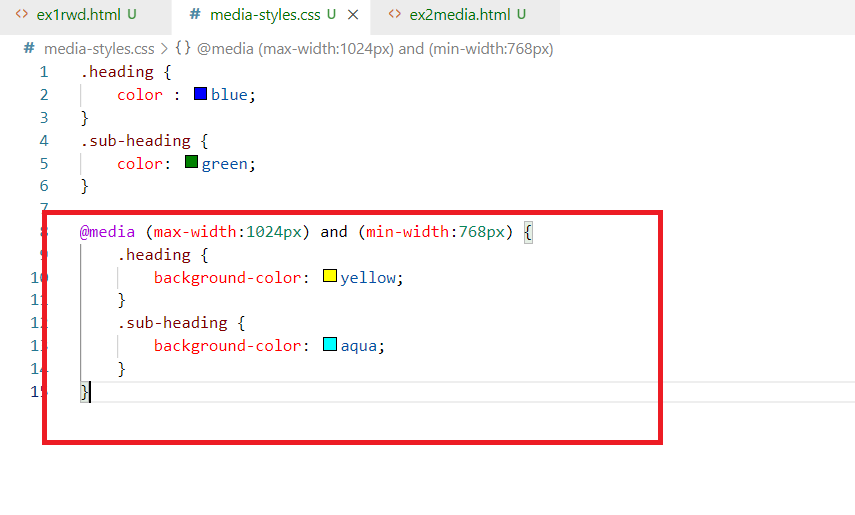


Output:

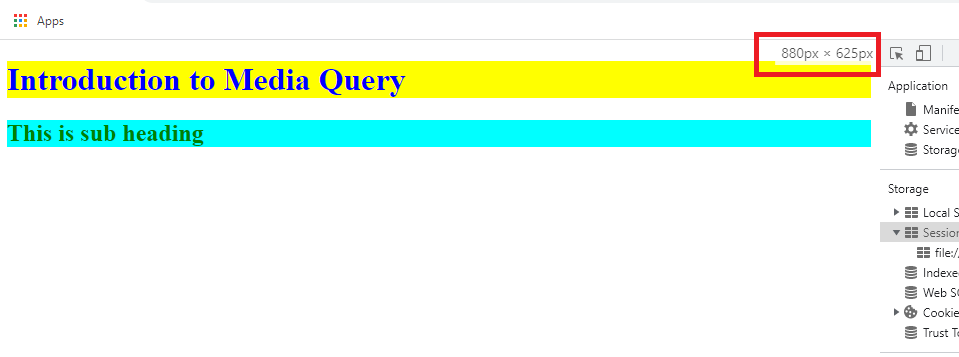


Applying the media rules for screen between some size.

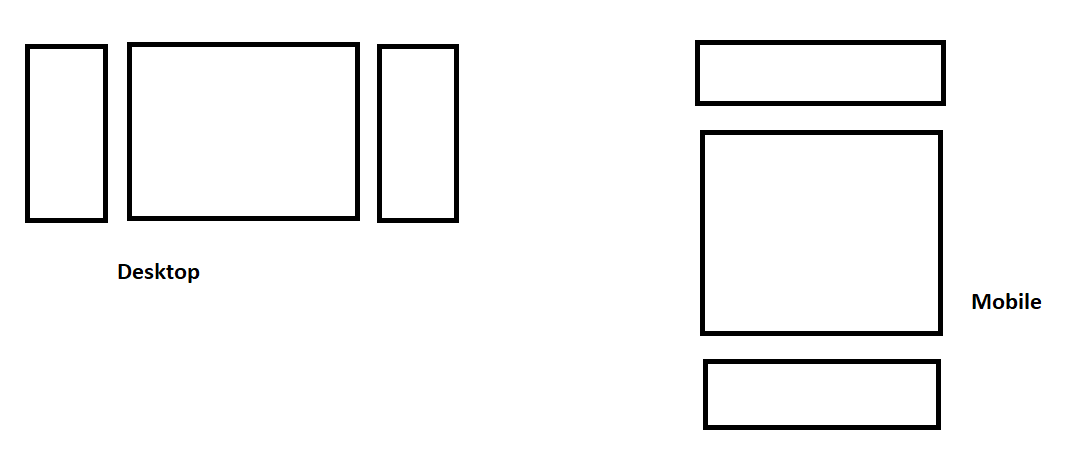
media-styles.css



Output: Screen having 768px to 1024px will have the background color



Flex box layout



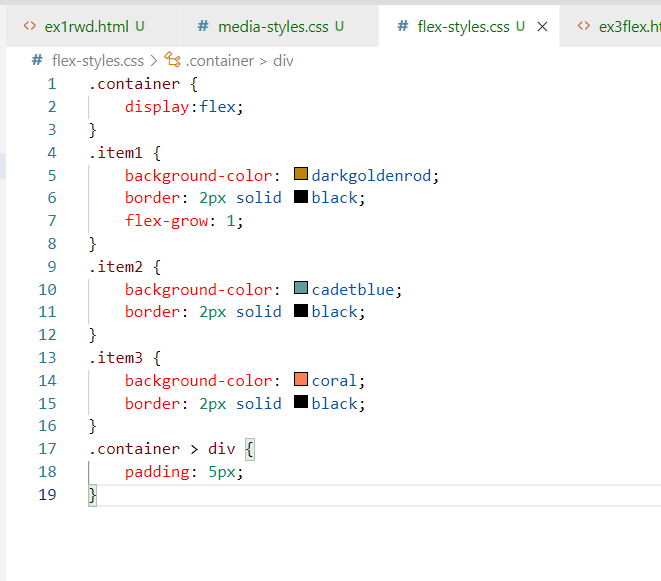
It provides an efficient way to layout, align, distribute space among items in the container, even when their size is unknown.

The main idea behind the flex box is to best fill the available space to accommodate all kind of devices & screen size.

A flex container can expand or shrink whenever need.

It is achieved using a property value called display:flex

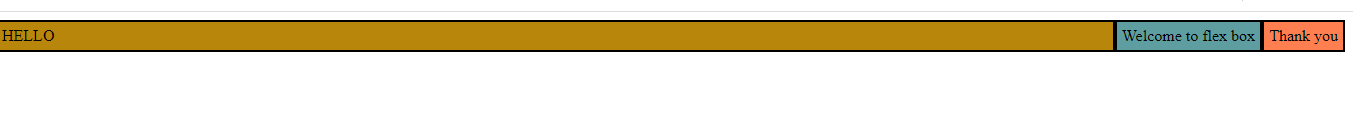
flex-styles.css



ex3flex.html

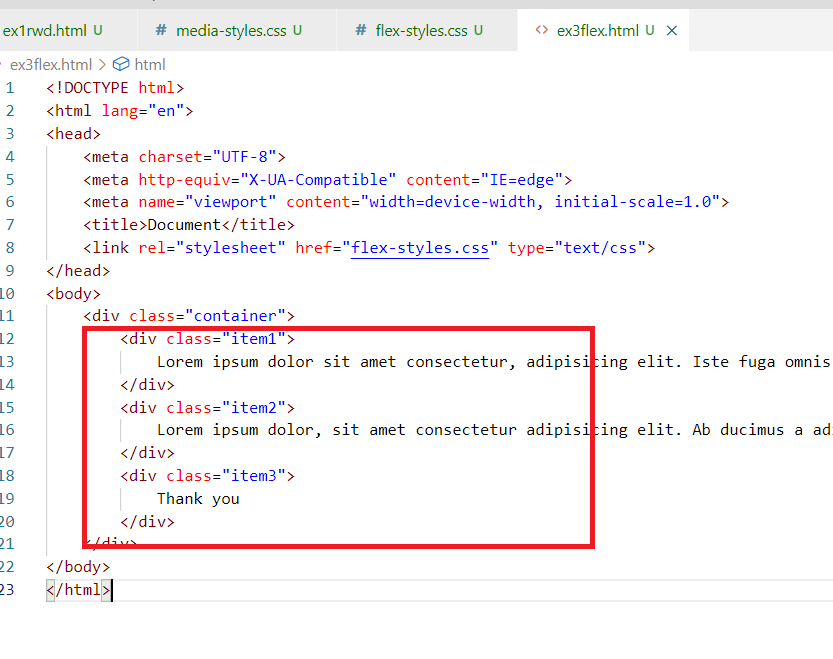


Output:

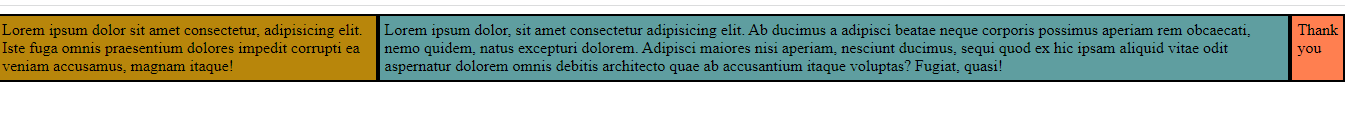


The contents would fit and the items would grow when the contents are more to accommodate the space.

ex4flex.html



Output:

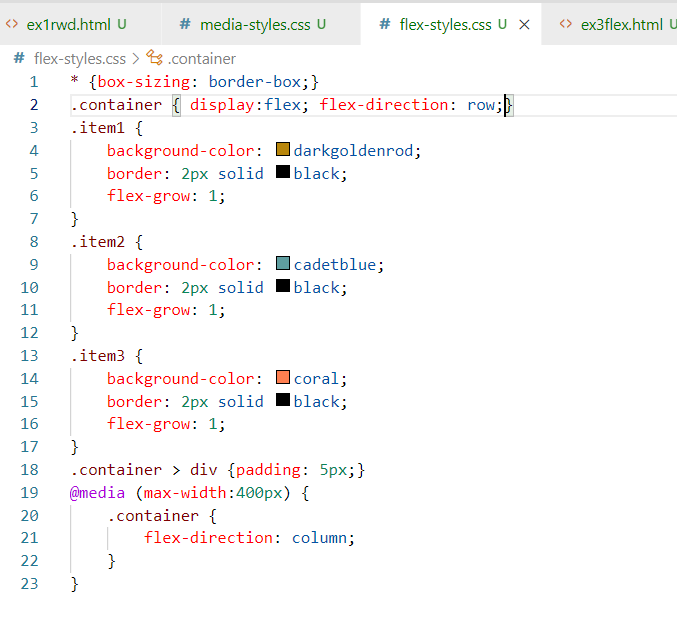


Output when you shrink the size

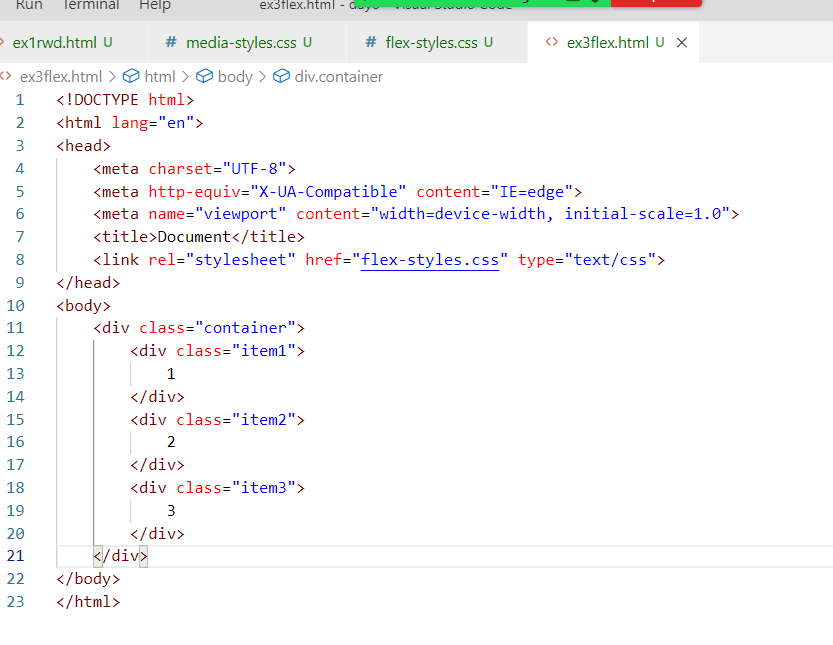


But you can use flex-direction to row or column to arrange the items to row or columns, usually when you want to arrange the items based on the device size if its less than 400px you can have items arranged vertically, if more than 400px you want items to appear horizontally, then you can use flex-direction.

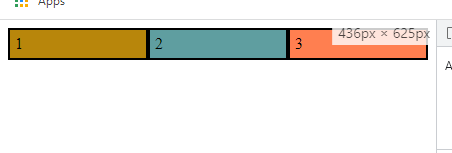
flex-styles.css



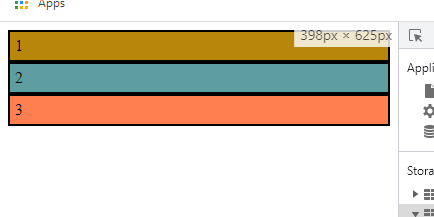
ex3flex.html



Output when screen size is > 400px:



Output when screen size is <= 400px



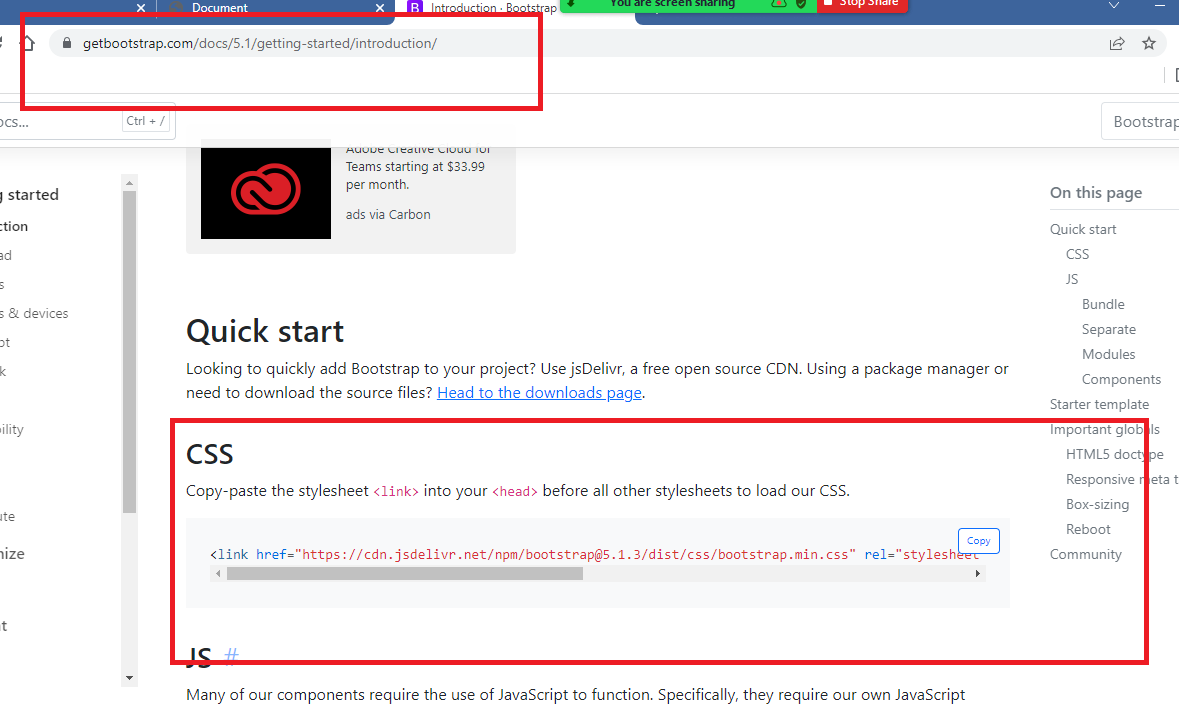
Bootstrap.js

It is a third party library which is used to style the websites, it has many inbuilt classes which are created keeping all kinds of devices & size in mind, you can use these classes in your project and they behave as per the device.

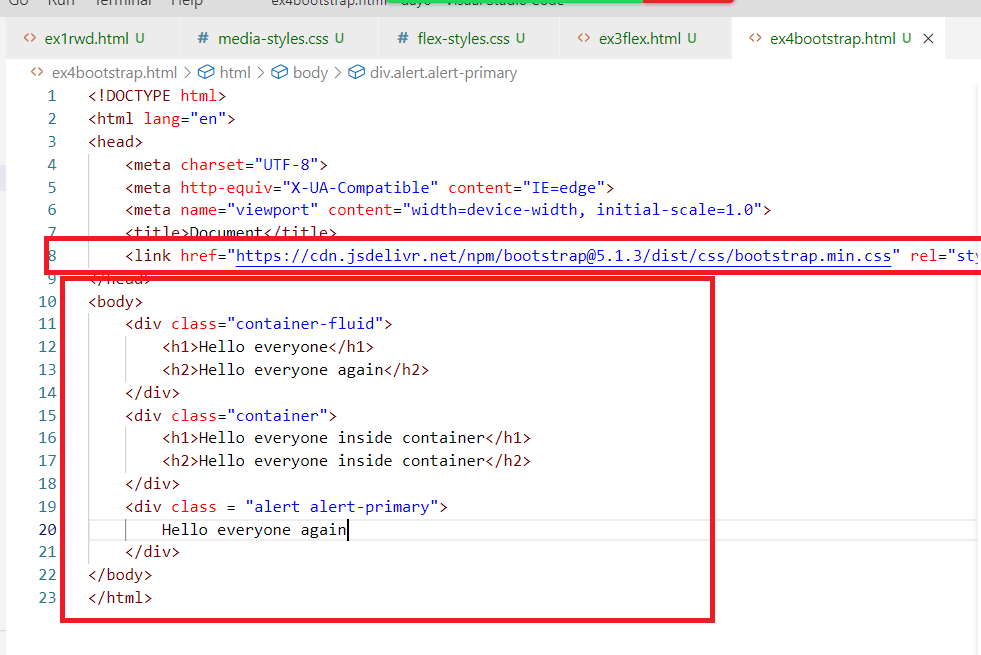
<https://getbootstrap.com/>

Above is the official website of bootstrap.js, this website provides the examples so that you can see the result of using their classes.

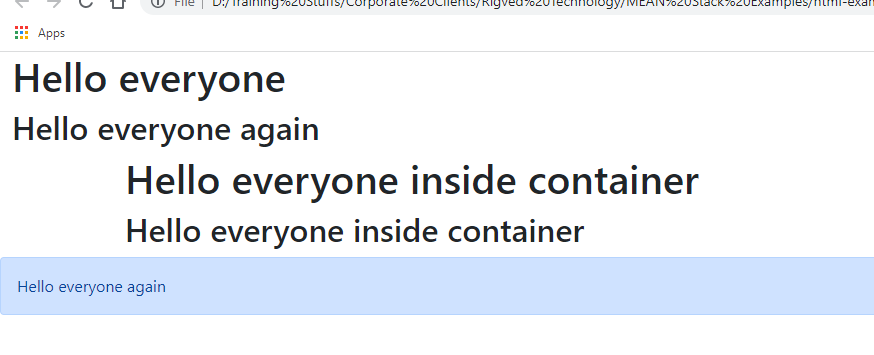
Adding bootstrap.js to the website



ex4bootstrap.html



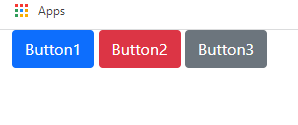
Output:



You can use different button classes



Output:

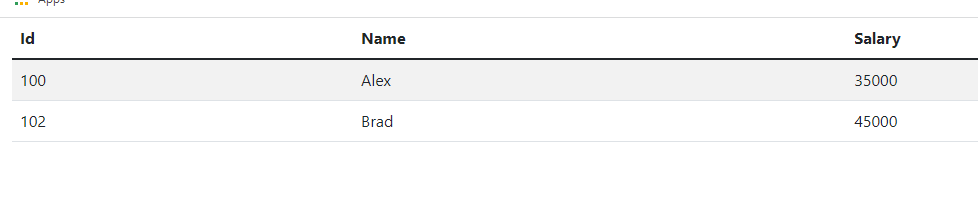


There are many types of classes like table, card, row & col and so on

table, table-striped



Output:



row & col classes

It is like a flex box, where you can have items in a row, but maximum 12 items you can have in a row, these items are created with class col, these col would occupy maximum 12 items space.

i.e.,

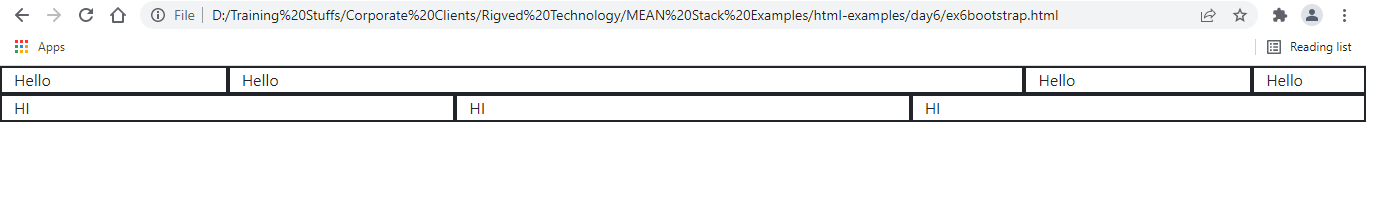
col-1 -> occupy 1 space

col-3 -> occupy 3 column space

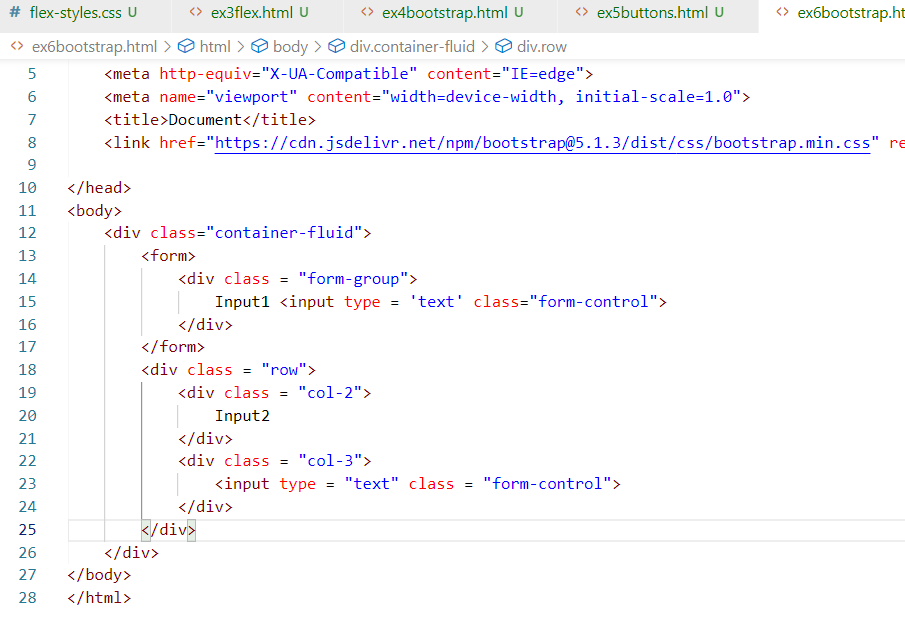


Note: You can see at 10th line div.row > div, which means selects the <div> inside <div class = “row”>

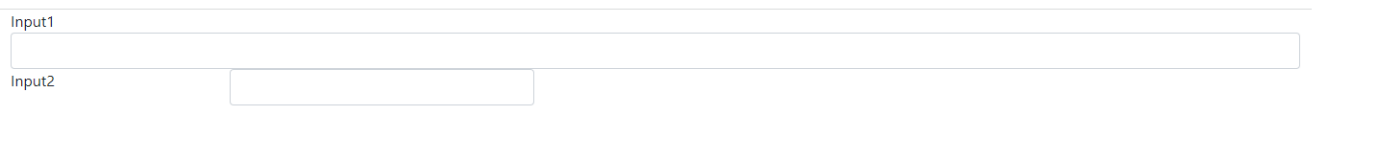
Output:



You can also apply lot of other classes like card, form-control and so on.



Output:



Activity:

1. Try today’s exercises like media query, flexbox, some bootstrap examples
2. Goto the bootstrap website <https://getbootstrap.com/docs/5.1/getting-started/introduction/>

Try out all the classes and observe the result, some important classes you had to try.

1. card
2. row & col
3. container, container-fluid
4. alert, alert-danger
5. navs & tab
6. navbar
7. spinners
8. toasts
9. pagination
10. progress

Along with the above classes, try other classes as well.

Day7:

Javascript

* It is a scripting language used to access HTML elements and
* can add dynamic behaviour to the web page,
* it can perform DOM (Document Object Model) manipulation - adding element, updating element, removing element and so on,
* it can add CSS at runtime,
* it can handle the events generated by HTML elements,
* it can perform client side validations,
* it can execute logics based on some user actions
* it can be used as a scripting language to write client & server side programs

Client: The browser can execute the javascript, client side scripts can work at the front-end to perform actions on the UI-side i.e.., browser or mobile

Server: Javascript is executed by backend environment like node.js, server side scripts can perform various business logics like interacting with the database, accessing filesystems, securing the applications and so on

We are going to learn Client side javascript, means it is executed by browser, but we are also going to learn Server side javascript which is executed by node.js later.

Either you write javascript for front-end or back-end the syntax or lexical structure is same.

Libraries, Frameworks or Tools in Javascript

* Ext.js
* Node.js
* Angular Framework
* React.js
* Vue.js
* Backbone.js
* jQuery

All the above lists are Javascript libraries & frameworks which simplifies writing javascript code, to understand them we must know Javascript first.

You can use <script> tag to add javascript to the HTML document, you can add Javascript directly to the HTML or create a script file and include in the HTML, means there are two types of Javascript

1. Internal Javascript
2. External Javascript

Where to write <script> tag in the HTML

You can write <script> either in <head> or <body> or both

<script> inside <head> loads when the head of the document is loaded, but before body, if there are more scripts then the document takes time to load the content

<script> inside <body> loads when the body of the document is loaded, if you write <script> after all the contents then the script is loaded after all the contents are loaded.

Whether you write script in body or head they are loaded in two ways

1. Scripts that are executed as soon as they load
2. Scripts that are executed only when some actions are performed

How many <script> tags can be written in the HTML

You can write any number of <script> but they are executed in the order they appear

Fundamentals / Lexical Structure of Javascript

* Datatypes
* Comments
* Keywords
* Variables
* Functions/Methods
* Loops
* Conditions
* Operators
* Classes & Objects

Inbuilt objects in Javascript

Javascript provides lot of inbuilt objects which every browser understands like

* window
* document
* console
* Math
* String
* Date

Note: Javascript is case sensitive

window: It is a global object, all the objects are part of window whether its document or console or Math or String etc, window is the main object of the browser, it appears at the root level

document: It is used to access browser DOM, using which you can manipulate DOM elements like accessing HTML element, removing HTML element, updating the content and etc

Object: It is a real world entity which will have 2 things

1. property / variable
2. behaviour/function

Property: Object properties like what object has

Behaviour: Object actions like what object does

Comments: These are ignored by browser or execution engine, in Javascript you can write two types of comments

1. single line comment: //
2. multi-line comments: /\* \*/

day7/ex1helloworld.html



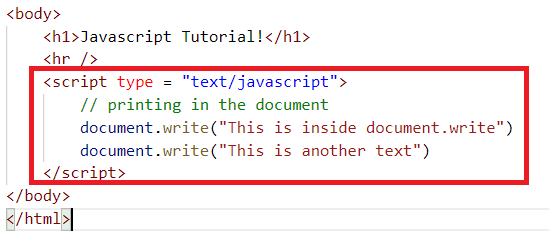
Output:



Scripts must be inside the <script> tag, since <script> tag supports other types of scripts you need to mention type = “text/javascript”, so that browser understands what type of script you are writing.

document.write(): It is used to write content to the document, write() is a function inside document object, it is case sensitive

Note: semicolon at the end of the statement is optional in Javascript as long as each statement is in separate lines, if you have multiple statements in the same line then each statement must be separated by semicolon.



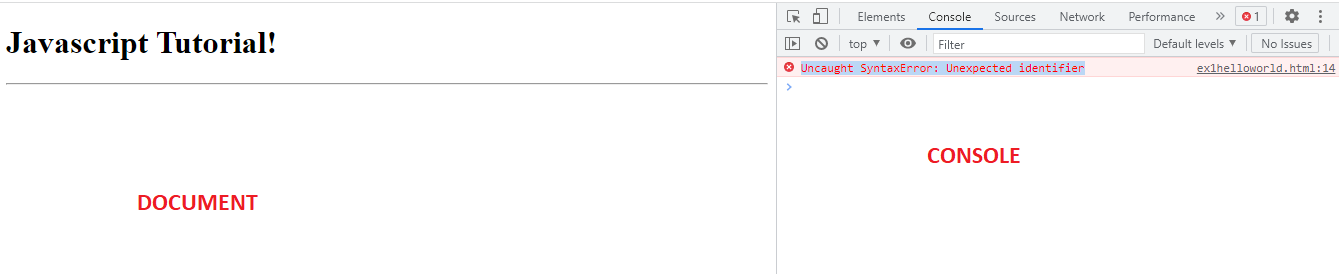
Output:



Since HTML doesn’t consider any extra space or line unless you use <br> tag, the contents appear in the same line



Here two statements are written in the same line without semicolon, but browser doesn’t show any error in the document, it shows error in the console (developer tools)

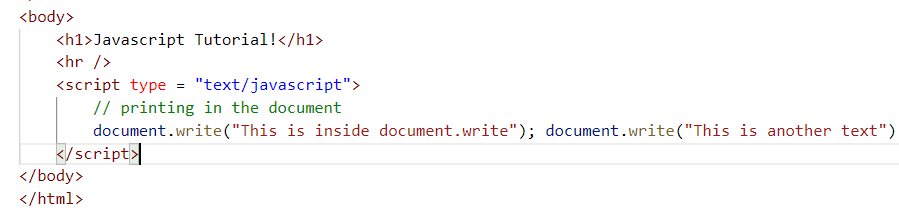


Here the document is not showing any error, if you are not getting the output as you expect check the console if there are any errors, try to understand the error and resolve it.

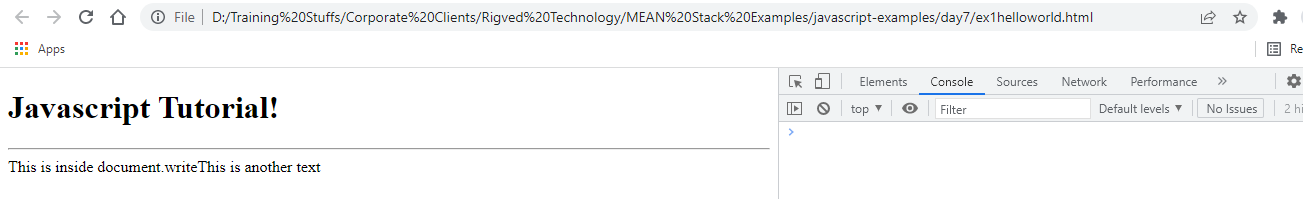
How to resolve this error

There are two ways

1. Write both the statements in separate line
2. If both are in the same line use ; (semicolon)

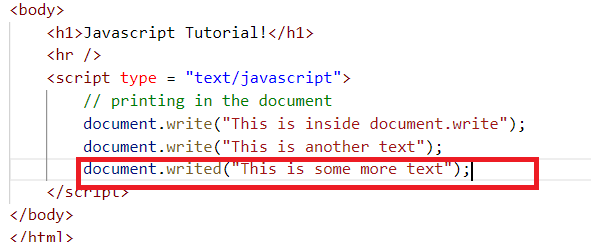


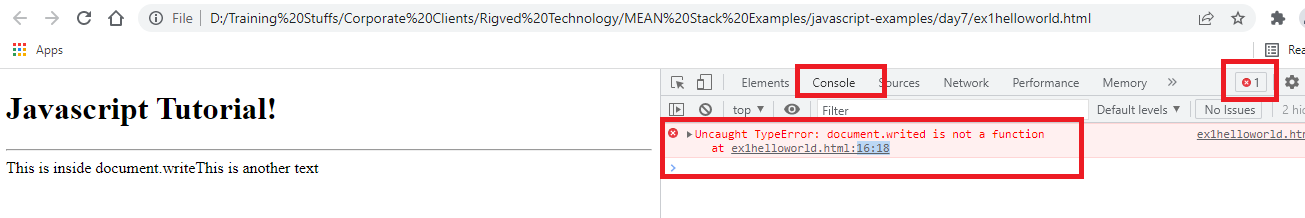
Output:



Note: Any new programmer can improve the logical skills when the errors in the program is resolved by himself.

Another mistake in document.write





Declaring Variables:

Variables are memories to store the data, you can store any kind of data in a Javascript variable, to declare variable you can use a keyword ‘var’

var name = “Alex”;

var age = 35;

var salary = 35200.35;

var isMarried = true;

Variable names can have alphanumeric characters, means alphabets and numbers, but they must begin with letters followed by alphabets or numbers.

Valid Variable names

var num1; // valid

var num123; // valid

var num12abc; //valid

var first\_name;// valid, you can use underscores

var num$xyz; // valid, but most of them don’t use $

var \_; // valid, just an underscore is also valid

Invalid variable names

var 123abc; // invalid, can’t start with numbers

var 123; // invalid, can’t have just numbers

var abc-xyz; // invalid, can’t have hyphen, because treated as 2 variables need to subtracted

Literals: These are values the variables take, you can have strings, numbers, boolean, objects as literals

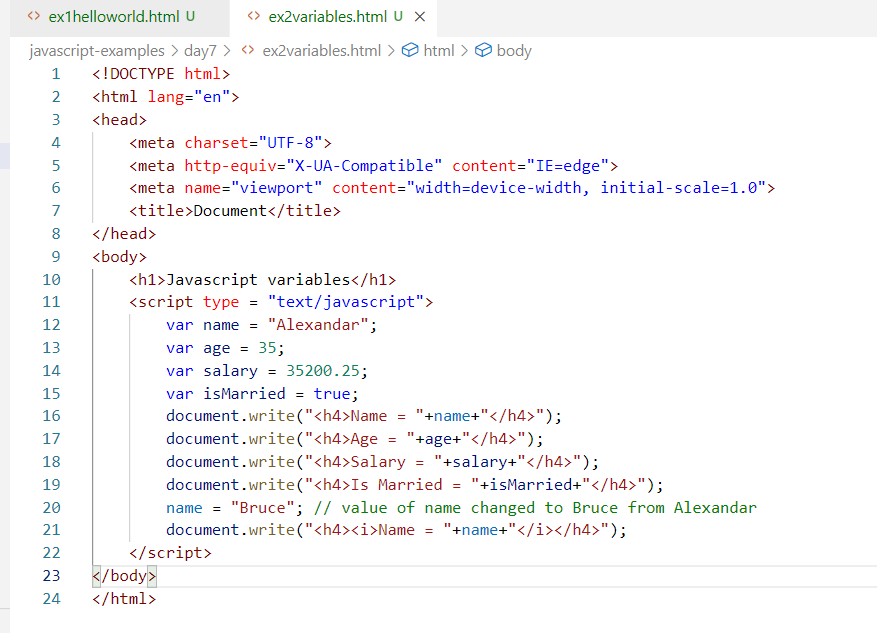
Note: A string can be in single or double quotes

var name = ‘Alex’; // valid

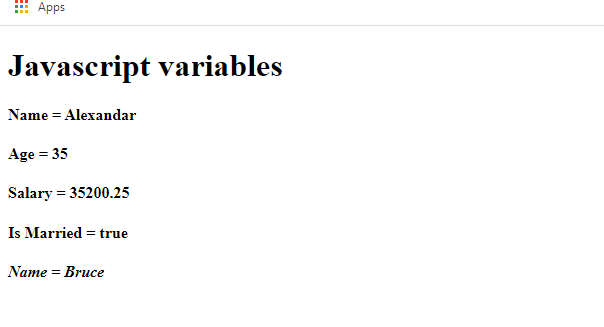
var name = “Alex”; // valid

var name = “Alex’; // invalid, because you are closing with single quote

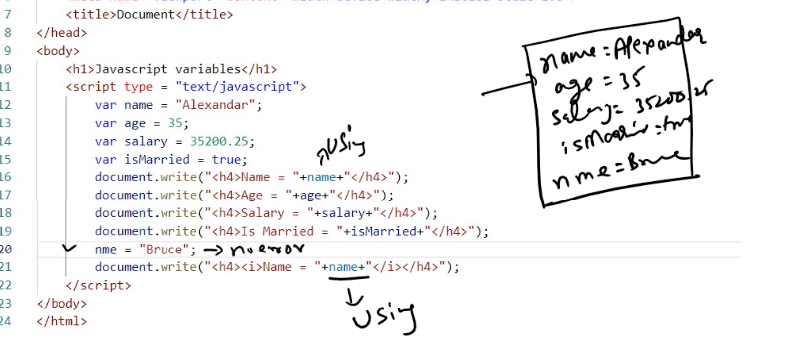
ex2variables.html



Output:

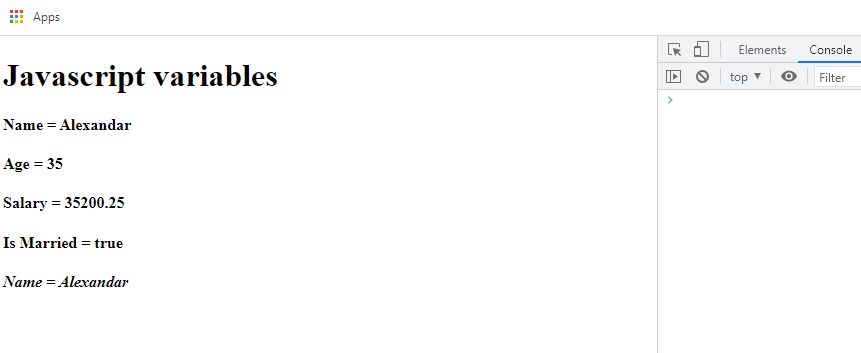


In Javascript you can create variables without using var keyword also



The above code at 20th line will create a new variable ‘nme’, hence you don’t get error.

Output:



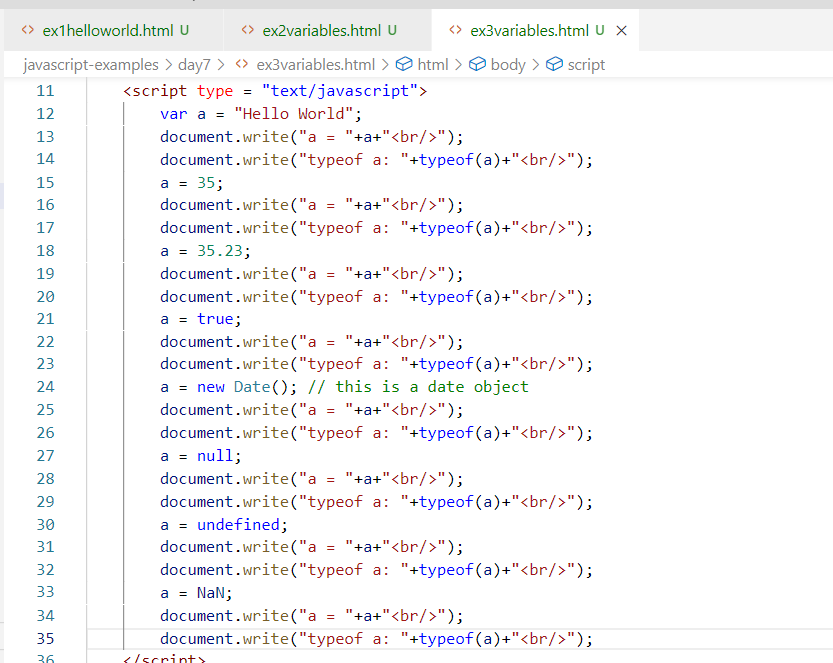
Note: When you are accessing the variable without declaring then only you will get undefined error.

var name = “Alex”;

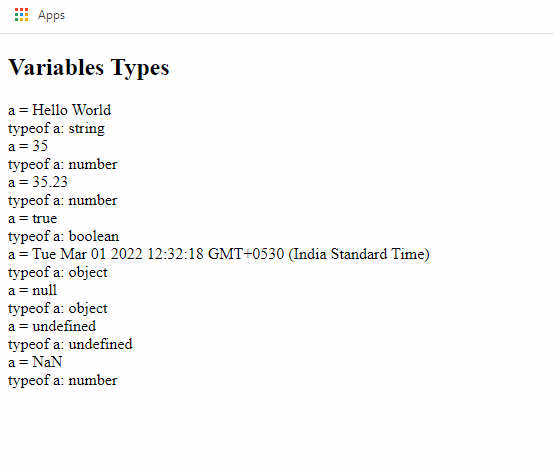
document.write(name); // accessing name variable

fullname = “David” + name; // accessing name variable, we are concatenating name with David

fullname is the variable that is getting initialized, thought it is not using var fullname, it is still okay in javascript.



Output:



Note: NaN is also a number type which means Not A Number, undefined is also one type, null is an object type.

Pop up boxes

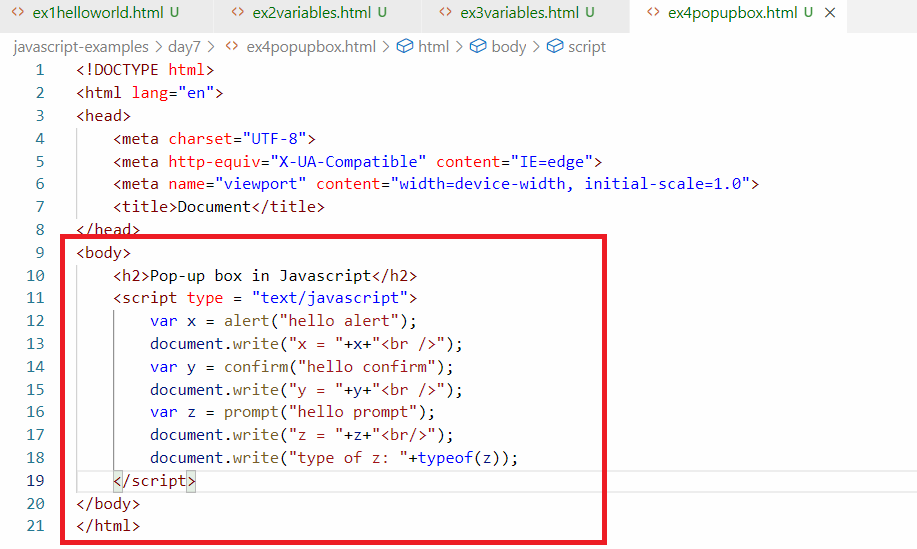
We can three types of popup box in javascript

alert(): creates an alert box

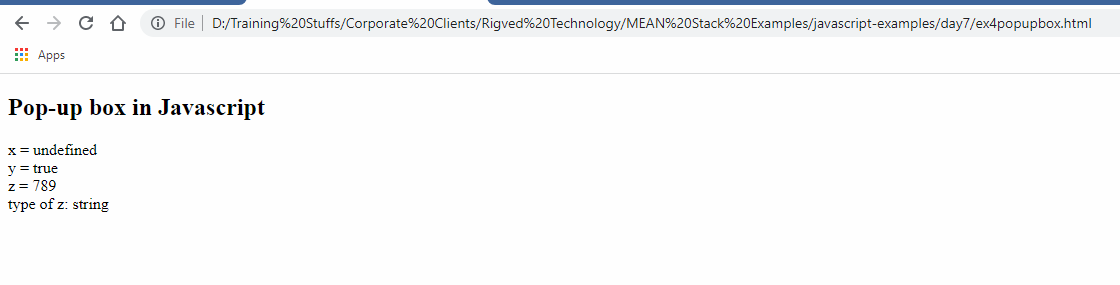
confirm(): creates a confirm with ok or cancel, returns true/false

prompt(): accepts an input from the user & returns to the program, the value returned by the prompt will always be string, which can be converted to other types using some inbuilt functions like parseInt(), parseFloat()

ex4popupbox.html



Output:



Operators in Javascript

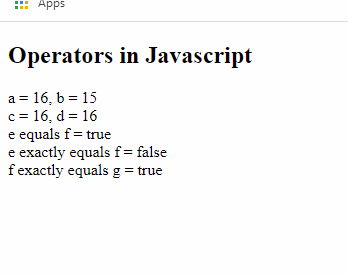
We have arithmetic operators, logical operators, bitwise operators, comparison operators, conditional operators and so on

+, -, \*, =, /, ++, --, <=, >=, <, >, !=, ==, ===, &&, ||, +=, -=, \*=, /=, %, ^

ex4operators.html



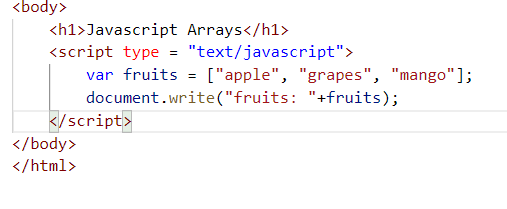
Output:



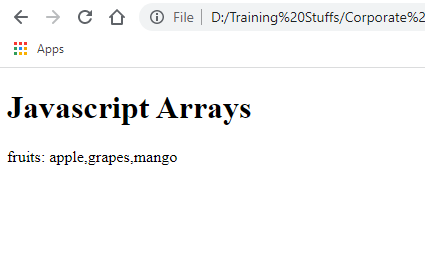
Datatypes: They are type of data a variable can store, there are two types of datatypes

1. primitive types: like string, number, boolean
2. non-primitive types: like arrays, objects

Arrays: It is a variable which can store more then one value



Output:



Loops

Executing same statements again and again repeatedly until some condition is met.

In Javascript there 3 types of loops

1. for
2. while
3. do while

Conditional branches

Executing some statements when some condition is met

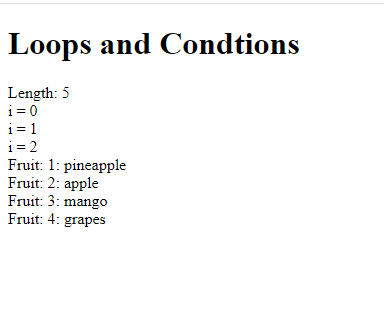
In Javascript there are following conditional branches

1. if
2. if else
3. if else if else if … else
4. switch

ex7loopsconditions.html



Output:

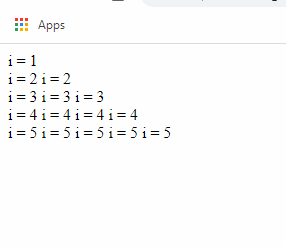


Nested loops:

A loop inside a loop is nested loop



Output:



Activity

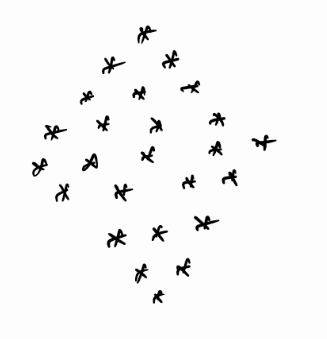
1. Try out all the above examples to understand variables, operators, loops & conditional branches
2. Using loops & conditional branches print only the odd numbers from 1 to 10
3. Using loops & conditional branches print only the even numbers from 1 to 10
4. In an array of elements [10, -3, 50, -2, 70, 40, 60], print the largest number & smallest number i.e., largest number must be 70, and smallest must be -3
5. Using loops & conditional branches create some patterns
6. Pyramid pattern

Output:



1. Diamond pattern

Output:



1. Reverse pyramid

Output:

