

request response cycle

frontend --> html css and js

**html** → hyper text markup language  
 ↓  
 structure and formatting

note -> when we talk about our html we don't talk about style, colour , or any functionality, we only talk about what items is present over page.

```
// boiler plate code
```

this is standard formate or skeleton of writing html code

paragraph element:

for paragraph we have <P> tag.

```
<p>Lorem ipsum dolor sit amet consectetur  
adipisicing elit. Hic voluptates quia  
delectus quas obcaecati fugiat doloremque  
dolore minus. Rerum nobis voluptates aliquid  
odit quasi pariatur animi labore, aperiam non  
esse?</p>
```

list element

<ul> -- unordered list

<ol> -- ordered list

<li> -- list item

```
2 <html>
8 <body>
.0   <p>ordered list</p>
.1   <ol type="i">
.2     <li>bread</li>
.3     <li>butter</li>
.4     <li>jam</li>
.5   </ol>
.6
.7   <ul>
.8     <li>bread</li>
.9     <li>butter</li>
.10    <li>jam</li>
.11  </ul>
.12
.13 </body>
```

ordered list

i. bread  
ii. butter  
iii. jam

• bread  
• butter  
• jam

```
<body>
  <h1>heading 1</h1>
  <h2>heading 2</h2>
  <h3>heading 3</h3>
  <h4>heading 4</h4>
  <h5>heading 5</h5>
  <h6>heading 6</h6>
</body>
</html>
```

**heading 1**

**heading 2**

**heading 3**

**heading 4**

**heading 5**

**heading 6**

attributes : these are used to add more information to the tag

<html lang ="en"> here lang="en" is attributes

**anchor tag :**

if we used add any link to page for that we have anchor tags .

```
<a href="https://www.netflix.com/in/">netflix</a>
```

netflix

**<br>** : used to add next line or line break in our page

```
<body>
  <p>Lorem,
  |   <br> ipsum <br>dolor.</p>
</body>
</html>
```

***bold italic and underline***

they are used to highlight the text on our page

<b>, <i>, <u>

```
<body>
  <b>bold</b><br>
  <i>italic</i><br>
  <u>underline</u><br>
</body>
</html>
```

**bold**  
*italic*  
underline

comments: this is part of code that should be not parsed

```
<!-- this is comment -->
```

html is not case sensitive language

<P> is same is <p>

<H1> is same <h1>

```
<body>
  <H1>this is capital heading</H1>
  <h1>this is small heading</h1>
</body>
</html>
```

**this is capital heading**

**this is small heading**

Img : to add image on page.

```

```

Task :- create a simple portfolio.

**KETAN GUPTA**

## Education

- 10th Class : 10 CGPA
- 12th Class : 92.4 %
- B.tech (CSE) : 8.2 CGPA

## Skills

- HTML, CSS and JAVASCRIPT
- NodeJS, SQL
- Python Java

## Hobbies

- Reading
- Coding

## Contact Me

- Email : user@gmail.com
- LinkedIn : linkedin.com/user131

\* created by frontend developer ★

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>PROTFOLIO</title>
</head>
<body>

  <h1>KETAN GUPTA</h1>
  

  <h2>Education</h2>
  <ul>
    <li><b>10th Class : </b> 10 CGPA</li>
    <li><b>12th Class : </b> 92.4 %</li>
    <li><b>B.tech (CSE) : </b> 8.2 CGPA</li>
  </ul>

  <h2>Skills</h2>
  <ul>
    <li>HTML, CSS and JAVASCRIPT</li>
    <li>NodeJS, SQL</li>
    <li>Python Java</li>
  </ul>

  <h2>Hobbies</h2>
  <ul>
    <li>Reading</li>
    <li>Coding</li>
  </ul>

  <h2>Contact Me</h2>
  <ul>
    <li><b>Email : </b> user@gmail.com</li>
    <li><b>Linkedin : </b> linkedin.com/user131</li>
  </ul>

  <p>CreatedBy frontend developer &#9733;</p>

</body>
</html>

```

## inline and block

block element : take up the full width available over the page and always start with new line.

```

ex.html > ↗ html > ↗ body
<html lang="en">
<body>
  <p style="background-color: aquamarine;">para 1</p>
  <p style="background-color: red;">para 2</p>
  <p style="background-color: green;">para 3</p>
</body>

```

inline element : this inline element will take only necessary width and they don't start from new line.

```

<body>

  <a href="" style="background-color: aquamarine;">google</a>
  <a href="" style="background-color: red;">facebook</a>
  <a href="" style="background-color: rebeccapurple;">instagram</a>

</body>

```

## div element

it is also called content division element.

div is container that is used to hold the other html element or group element together

div is block element

The screenshot shows a code editor with the following HTML code:

```
<body>
  <div href="" style="background-color: #lightblue;">
    <a href="" style="background-color: #aquamarine;">google</a>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum distinctio eligendi dolorum culpa veniam eum repellat omnis blanditiis officiis, ea ullam repudiandae quaerat consequatur! Veritatis suscipit facere sequi animi! Temporibus.</p>
  </div>

  <div href="" style="background-color: #orange;">
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Iure doloremque sint enim neque vitae eum perspiciatis excepturi fugiat aliquid a libero, consequuntur provident quas! Alias asperiores quaerat velit animi beatae!</p>
    <a href="" style="background-color: #red;">facebook</a>
  </div>
```

To the right of the code editor, there is a preview window showing the rendered HTML. It consists of two stacked 

elements. The top 

has a light blue background and contains a [google](#) link and a paragraph of lorem ipsum text. The bottom 

has an orange background and contains a 

element with more lorem ipsum text and a [facebook](#) link.

## span element

span is also used to create generic container that is used to hold the other html elements

it is an line element

The screenshot shows a code editor with the following HTML code:

```
<head>
<body>

  <span href="" style="background-color: #lightblue;">
    <a href="" style="background-color: #aquamarine;">google</a>
  </span>

  <span href="" style="background-color: #lightblue;">
    <a href="" style="background-color: #aquamarine;">google</a>
  </span>
  <span href="" style="background-color: #lightblue;">
    <a href="" style="background-color: #aquamarine;">google</a>
  </span>
```

To the right of the code editor, there is a preview window showing the rendered HTML. It consists of three inline  elements, each with a light blue background and containing a [google](#) link.

hr : horizontal line or rule on page

```
<body>
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Est, beatae.</p>
  <hr>
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Est, beatae.</p>
  <hr>
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Est, beatae.</p>
  <hr>
```

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Est, b  
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Est, b  
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Est, b

```
<body>
  <p>this is super script</p>
  a2 = b2 + c2
  <p>this is sub script</p>
  H2O
</body>
```

this is super script  
a<sup>2</sup> = b<sup>2</sup> + c<sup>2</sup>  
this is sub script  
H<sub>2</sub>O

semantic markup : it is the markup that relates to the meaning of the content.

semantic -> h1, p, img.

non semantic -> div, span

semantic markup

header tag

main tag

footer tag

nav tag

html entities.

it is a piece of text that begin with and '&' and ends with ''

it is used to display reserved characters

```
<body>
    <p> a &lt; b &lt; d &gt; ; &#8364; &
    #38;&diam;&ampforall&part;&#128507;&#128514;|
</body>
```

a < b < d >  
¢ € &♦ ∀∂ ☁ 😊

Task :- **this is rainy**  **and this is sunny**  (write this string in html para using html entities.)

```
<body>
    this is rainy &#127783; and this is sunny &
    #9728;
</body>
</html>
```

this is rainy  and this is sunny 

html table  
table are used to represent real life data.

to have table in html we have `<table>` tag

`<th>` --> table head  
`<tr>` --> to define row in table  
`<td>` --> to have data in table

`<caption>` --> to give caption

```

<body>
    <table border="4">
        <caption>table data </caption>
        <tr>
            <th>name</th>
            <th>age</th>
            <th>marks</th>
            <th>gender</th>
        </tr>
        <tr>
            <td>rahul</td>
            <td>21</td>
            <td>34</td>
            <td>male</td>
        </tr>
        <tr>
            <td>jai</td>
            <td>23</td>
            <td>45</td>
            <td>male</td>
        </tr>
        <tr>
            <td>Karan</td>
            <td>34</td>
            <td>55</td>
            <td>male</td>
        </tr>
    </table>
</body>

```

table data

name	age	marks	gender
rahul	21	34	male
jai	23	45	male
Karan	34	55	male

## semantics in table

```
<thead>  
<tbody>  
<tfoot>
```

## colsapn and rowspan

user to create cells which span over multiple rows or columns

```
<body>  
  <table border="3">  
    <caption>proportion of fruits</caption>  
    <tr>  
      <td rowspan="2"></td>  
      <th colspan="2" rowspan="2">property</th>  
      <th rowspan="2">profit</th>  
    </tr>  
    <tr>  
      <th>color</th>  
      <th>price</th>  
    </tr>  
    <tr>  
      <td>mango </td>  
      <td>yellow</td>  
      <td>100rs</td>  
      <td>king of fruits</td>  
    </tr>  
    <tr>  
      <td>apple</td>  
      <td>red</td>  
      <td>100rs</td>  
      <td>healthy fruit</td>  
    </tr>  
    <tr>  
      <th colspan="4" rowspan="1">this is fruits table </th>  
    </tr>  
  </table>  
</body>
```

proportion of fruits			
	property		profit
	color	price	
mango	yellow	100rs	king of fruits
apple	red	100rs	healthy fruit
this is fruits table			

to have form in html we have <form tag>

form are used to collect the data from the user in efficient manner

action attribute :- used to define what action need to perform when our form is going to be submitted or where the data of the form should be sent.

input

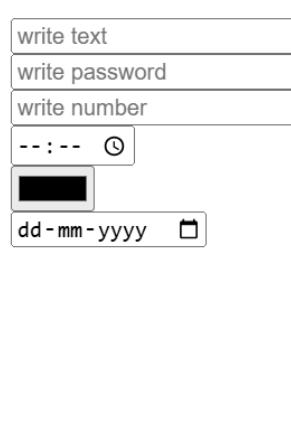
used to create multiple form control

there are multiple type of input which is created using the type attribute.

```
<body>
  <form action="/action">

    <input type="text" placeholder="write text "><br>
    <input type="password" placeholder="write
password"><br>
    <input type="number" placeholder="write number"><br>
    <input type="time"><br>
    <input type="color"><br>
    <input type="date" name="" id=""><br>

  </form>
```



The screenshot shows a browser window with a dark theme. On the left is the HTML code for a form with six input fields. On the right, the corresponding UI elements are displayed: a text input field with placeholder "write text", a password input field with placeholder "write password", a number input field with placeholder "write number", a time input field (a small clock icon), a color input field (a small color swatch), and a date input field with placeholder "dd-mm-yyyy".

```
<body>
  <form action="/action">
    <label for="username">username : </label>
    <input type="text" placeholder="write
here" id="username">
    <br>
    <label for="pass">password : </label>
    <input type="password" placeholder="write
here" id="pass">
  </form>
```



The screenshot shows a browser window with a light theme. On the left is the HTML code for a form with two labeled inputs. On the right, the UI is shown: a label "username :" followed by a text input field with placeholder "write here", and a label "password :" followed by a password input field with placeholder "write here". Below the form is a large black button with the word "Chat" in white.

```
<button type="button">click me </button>
<br>
<button type="reset">reset</button>
<br>
<button type="submit">submmmit</button>
</form>
```

click me  
reset  
submmmit

name attribute :- control name of the form.  
submit form the value which is sended after submitting form it go in the form name/value pair

username : ashish

The screenshot shows a web page with a form and a sidebar. The form contains fields for username and password, and a submit button. The sidebar on the right shows a 'Chat' section with a message from Shash.

```
<html lang="en">
<body>
<form action="/action">
<label for="username">username : </label>
<input type="text" placeholder="write here" id="username" name="username">
<br>
<label for="pass">password : </label>
<input type="password" placeholder="write here" id="pass" name="userpassword">
<br>
<button type="submit">submmmit</button>
</form>
```

username : write here  
password : write here  
submmmit

Chat  
from Shash  
yes

username : rahul  
password : .....  
submmmit

① 127.0.0.1:5500/action?username=rahul&userpassword=12345

```

<body>
    <!-- create a search box for google search -->
    <form action="https://www.google.com/search?q=a1">
        <label for="google">search in google : </label>
        <input type="text" name="q" id="google">
        <button type="submit">search</button>
    </form>

```

## Check box

```

<body>
    <!--check box -->
    <form action="/action">
        <h3>select fruits</h3>
        <input type="checkbox" name="fruitsName" id="mango" value="mango">
        <label for="mango">mango</label>
        <br>
        <input type="checkbox" name="fruitsName" id="apple" value="apple">
        <label for="apple">apple</label>
        <br><input type="checkbox" name="fruitsName" id="banana" value="banana">
        <label for="banana">banana</label>
        <br>
        <br>
        <button type="submit">submit</button>
    </form>

```

**select fruits**

- mango
- apple
- banana

**submit**

## Radio button

```

<body>
    <!--radio button -->
    <form action="/action">
        <h5>are you 18+</h5>
        <input type="radio" name="usage" id="above" value="moreThan18">
        <label for="above">above 18</label>
        <br>
        <input type="radio" name="usage" id="belwo" value="lessThan18">
        <label for="belwo">belwo 18</label>
        <br>
        <br>
        <button type="submit">submit</button>
    </form>

```

**are you 18+**

- above 18
- belwo 18

**submit**

## Dropdown

```
<body>
    <!--draw down -->
    <form action="/action">
        <h5>Choose one fruit</h5>
        <label for="fruit">choose fruit : </label>
        <select name="fruitsName" id="fruit">
            <option value="mango">mango</option>
            <option value="guava">guava</option>
            <option value="banana">banana</option>
            <option value="apple">apple</option>
        </select>
        <br>
        <br>
        <button type="submit">submit</button>
    </form>
```

Choose one fruit

choose fruit : mango

mango  
guava  
banana  
apple

submit

```
<body>
    <!--range down -->
    <form action="/action">
        <label for="sound">sound : </label>
        <input type="range" name="soundRange" id="sound"
        min="0" max="100">

        <br><br>
        <button type="submit">submit</button>
    </form>
```

sound : 

submit

## Text area

```
<body>
    <!-- text area -->
    <form action="/action">
        <label for="feedback">provide your feedback : </label>
        <textarea name="feedback" id="feedback"></textarea>
        <br><br>
        <button type="submit">submit</button>
    </form>
</body>
```

provide your feedback :

submit

## Time table

```
<body>
  <h1>Class Time Table</h1>
  <table border="1" cellpadding="10" cellspacing="0">

    <tr>
      <th>Time</th>
      <th>Monday</th>
      <th>Tuesday</th>
      <th>Wednesday</th>
      <th>Thursday</th>
      <th>Friday</th>
    </tr>
    <tr>
      <td>9:00 - 10:00</td>
      <td>Math</td>
      <td>English</td>
      <td>Science</td>
      <td>Computer</td>
      <td>Math</td>
    </tr>
    <tr>
      <td>10:00 - 11:00</td>
      <td>English</td>
      <td>Science</td>
      <td>Math</td>
      <td>Computer</td>
      <td>English</td>
    </tr>
    <tr>
      <td>11:00 - 12:00</td>
      <td colspan="5">Lunch Break</td>
    </tr>
    <tr>
      <td>12:00 - 1:00</td>
      <td>Science Lab</td>
      <td>Math</td>
      <td>English</td>
    </tr>
  </table>
</body>
```

```

        <td>Computer Lab</td>
        <td>Project Work</td>
    </tr>
    <tr>
        <td>1:00 - 2:00 </td>
        <td>Sports</td>
        <td>Library</td>
        <td>Science</td>
        <td>English</td>
        <td>Math</td>
    </tr>

</table>

</body>

```

## Class Time Table

Time	Monday	Tuesday	Wednesday	Thursday	Friday
9:00 - 10:00	Math	English	Science	Computer	Math
10:00 - 11:00	English	Science	Math	Computer	English
11:00 - 12:00	Lunch Break				
12:00 - 1:00	Science Lab	Math	English	Computer Lab	Project Work
1:00 - 2:00	Sports	Library	Science	English	Math

## Task 2 :- product catalog.

```

<body>

    <h1>Product Catalog</h1>
    <p>Explore our latest collection of amazing products. Click "Buy Now"
to visit the product page.</p>

```

```

<table border="1" cellpadding="10">
  <tr>
    <th>Product Image</th>
    <th>Product Name</th>
    <th>Price</th>
    <th>Buy Now</th>
  </tr>
  <tr>
    <td></td>
    <td>Wireless Headphones</td>
    <td>49.99K</td>
    <td><a href="https://www.amazon.in/Apple-Headphones-Cancellation-Transparency-Personalised/dp/B0DGJ62NRY/ref=sr_1_1_sspa?crid=1FSJOHVPMODC&dib=eyJ2IjoiMSJ9.gO3N4uoVwqD1q04Yx2xIgoBWR71qqqjQmZsu22VP2YgTspCuOTPSeq31Iez2kLjw8QXGFB-EPFzzT6mYSpqVtPFTiFldptPIfVRoIr98E1xJ30_1hATwr96xbnNz_x6o-4AU6o0cHOaH4ZOh-qP3eVvgRpp5E14RN8Fgoeb7NF1vfwyMk1Q0s1QQAUegjeUUYMmd87sS5wG4ItlkfJCyC1mumOFdFZiMNRSvSZSlpUg.9UIh6azCfEpsoqOPx5g9ZrtESRUcqWUKJiNx6MzYEw&dib_tag=se&keywords=apple%2Bheadphone&nsdOptOutParam=true&qid=1763040272&sprefix=apple%2Bhead%2Caps%2C217&sr=8-1-spons&aref=bj0dKuLsaW&sp_csd=d21kZ2V0TmFtZT1zcF9hdGY&th=1" target="_blank">Buy Now</a></td>
  </tr>
  <tr>
    <td></td>
    <td>Smart Watch</td>
    <td>79.99K</td>
    <td><a href="https://www.amazon.in/Apple-Watch-Starlight-Aluminium-Sport/dp/B0FQFYMKMF/ref=sr_1_4?crid=3L8XUNR1F7CHX&dib=eyJ2IjoiMSJ9.n_y1kbhbfn8CW5xaGQXazjsrmh-rN8z3tn93fEthvU45bdQUN2WGEV3fpgrtIQcmN-4Cnr8UhbJ9xz-moSNJ7n1cN9nE90Br1BN6LyLs2FZLJoPaNZ2rOGBaFsUtbOpgZduV4IyLbIMYOEp46GWrlEoxLz0vb6KCSMxfUgqCyHtrn3_bC7FkAABrmk8LZmeLlFzWhphFhABNu67GeNkECP8G2aoaT7ipNDYFK0GMz8.wEftvwVurQV23G40i5G7gxSukmZ6QBMg50XLmziI7yA&dib_tag=se&keywords=apple%2Bwatch&nsdOptOutParam=true&qid=1763040424&sprefix=applewatch%2Caps%2C225&sr=8-4&th=1" target="_blank">Buy Now</a></td>
  </tr>
  <tr>
    <td></td>
    <td>Smart phone</td>
  
```

```
<td>99.99K</td>
<td><a href="https://www.amazon.in/iPhone-Pro-512-Promotion-Breakthrough/dp/B0FQF5DG3P/ref=sr_1_2?crid=1ZWQQ7M0ZSGPA&dib=eyJ2IjoiMSJ9.nmwGHrewwiw-c3C0yW3Sq4tib_EVxgeGBatA6N00_OuAR0x161_OxqEhxWWGOyF0t5HHEU4k9bovxvcx6EnFENZMRnTa8LDq8n0hbey0dDGNQtXeL7ArGCbga2Ugdmyr56njYTQE70yozE0WkKjayXX1NfaJ0YkbAByTsMvhBn7IUdn1dg3L7F6b-ImSh3Szmk5AmfPr5_k4wpn68DSMaXdbKxQPozmccVyVWNcI2Q.FOAEmMSfLq5J2yJcIhFMOjlr3sWgXph9hnG7LL_vHZc&dib_tag=se&keywords=apple+phone+17&qid=1763040639&sprefix=apple+phone+%2Caps%2C282&sr=8-2" target="_blank">Buy Now</a></td>
</tr>
</table>
</body>
```

# Product Catalog

Explore our latest collection of amazing products. Click "Buy Now" to visit the product page.

Product Image	Product Name	Price	Buy Now
	Wireless Headphones	49.99K	<a href="#">Buy Now</a>
	Smart Watch	79.99K	<a href="#">Buy Now</a>
	Smart phone	99.99K	<a href="#">Buy Now</a>

## Customer feedback form

```
<body>

    <h1>Customer Feedback Form</h1>
    <p>Please fill out this form to share your feedback with us.</p>

    <form action="#">

        <label for="name">Name : </label><br>
        <input type="text" id="name" name="userName"><br>
        <br>
        <label for="email">Email : </label><br>
        <input type="email" id="email" name="email"><br>
        <br>
        <label for="rating">Rate our service (1 to 10) :</label><br>
        <input type="range" min="1" max="10" id="rating"
name="userRating"><br>
        <br>
        <label for="comment">comment : </label><br>
        <textarea name="comment" id="comment"></textarea><br>
        <br>
        <p>Would you recommend us?</p>
        <input type="radio" name="recommend" value="yes" id="yes">
        <label for="yes">Yes</label>
        <input type="radio" name="recommend" value="no" id="no">
        <label for="no">No</label>

        <br><br>
        <button type="submit">Submit Feedback</button>

    </form>
</body>
```

# Customer Feedback Form

Please fill out this form to share your feedback with us.

Name :

Email :

Rate our service (1 to 10):



comment :

Would you recommend us?

Yes  No

## Travel Blog Post page.

```
<body>

    <h1>My Travel Blog</h1>
    <p>Welcome to my travel blog! I share my experiences, destinations, and travel tips here.</p>

    <h2>My Latest Trip</h2>
    <p>I recently visited the mountains and it was an amazing experience!
    <a href="#readmore">Read More</a></p>

    <br><br>

    <video controls width="500">
        <source src="vid.mp4" type="video/mp4">
    </video>

    <h2>My Favorite Destinations</h2>
    <ul>
        <li>Manali, India</li>
        <li>Goa, India</li>
        <li>Bali, Indonesia</li>
        <li>Paris, France</li>
    </ul>

    <h3 id="readmore">Read More</h3>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptates quis quas eos modi, eius quidem possimus minus provident facilis, iste laborum voluptate quos dolore officiis veniam, perferendis ipsa. Adipisci recusandae quisquam necessitatibus possimus optio obcaecati fugit laudantium pariatur laboriosam, tempore tempora quibusdam impedit ipsum aliquid fuga ex! Eos officia in ab nostrum modi culpa ut earum similiqe
```

sed asperiores? Facere delectus ab natus ut enim ad maiores optio sit alias ullam? Enim adipisci repellat nobis doloremque incident dignissimos ex asperiores, ad at vero, id nesciunt saepe dolorum eveniet sit eos reiciendis et! Facere beatae natus vel, illum, sed culpa quos soluta adipisci, nobis recusandae dolorem sint! Consequuntur maxime perferendis nulla eveniet explicabo delectus sunt, cum corporis ea commodi quis repellendus facere neque autem nostrum adipisci aliquam magnam iure. Molestias in porro libero quia, totam dicta, aut perferendis quisquam hic provident, obcaecati sapiente. Nisi illum vero, hic harum sint dignissimos alias iusto, sequi facere quibusdam possimus quae saepe qui laborum velit id suscipit molestiae voluptatem corrupti voluptates vel officiis. Dolore ipsum quam porro voluptates, excepturi eos cumque? Nihil odit odio consequuntur quasi quisquam dicta ut expedita aliquid sapiente, numquam laudantium explicabo quae. Optio, nemo harum? Impedit quos possimus blanditiis provident. Aliquam modi voluptatibus sunt, enim repellendus eaque soluta recusandae fugiat provident voluptatem id hic nemo qui fugit libero! Voluptatibus eligendi incident eaque sunt veritatis rerum maxime?</p>

</body>

# My Travel Blog

Welcome to my travel blog! I share my experiences, destinations, and travel tips here.



## My Latest Trip

I recently visited the mountains and it was an amazing experience! [Read More](#)



## My Favorite Destinations

- Manali, India
- Goa, India
- Bali, Indonesia
- Paris, France

### [Read More](#)

Traveling helps me discover new cultures, meet wonderful people, and create memories for life. Stay tuned for my next adventure! Lorem ipsum dolor sit amet consectetur adipisici perferendis aperiam quos voluptatem eaque, cupiditate eveniet voluptates similique. Debitis mollitia incidunt asperiores aliquid ducimus, libero enim similique! Eligendi minima, de voluptatibus, praesentium earum ad consequatur obcaecati velit? Ullam, velit impedit veniam natus expedita quidem dolores ut minus facere deserunt. Laudantium exercitationem al maxime, enim aspernatur animi qui repellat sunt perferendis autem pariatur excepturi a aperiam! Itaque, hic! Quo doloribus dicta provident esse sapiente iure maxime, placeat id rep obcaecati, doloribus distinctio rem dolore, ullam cumque. Placeat quisquam quasi sequi ab a error rem sint eum, quo iure quam ipsum suscipit molestiae quas beatae earum quae eaq omnis, quo commodi animi obcaecati quisquam in reprehenderit, id temporibus fugiat perspiciatis libero distinctio ab quia provident autem inventore? Voluptate velit ea, beatae sed dolorem. Provident iusto pariatur ex expedita quisquam voluptas vero earum porro nobis.

## Student registration form.

```
<body>
  <header>
    <h1>Student Information Portal</h1>

    <nav>
      <a href="">Home</a> |
      <a href="">Add Student</a> |
      <a href="">Student List</a> |
    
```

```

        <a href="">Contact</a>
    </nav>
</header>
<hr>
<h2>Add Student</h2>

<form action="#">
    <label for="studentName">Student Name: </label><br>
    <input type="text" name="StudentName" id="studentName"> <br><br>
    <label for="roll">Roll Number: </label><br>
    <input type="text" name="StudentRoll" id="roll"> <br><br>
    <label for="email">Email: </label><br>
    <input type="email" name="StudentEmail" id="email"> <br><br>
    <label for="Course">Course : </label>
    <select name="course" id="Course">
        <option value="BCA">BCA</option>
        <option value="BTech">B.Tech</option>
        <option value="MCA">MCA</option>
        <option value="Bsc">B.sc</option>
    </select>

    <br><br>
    <button type="submit">Add Student</button>
</form>

<h2>Student List</h2>

<table border="1" cellspacing ="0" cellpadding="10">
    <thead>
        <tr>
            <th>Roll No</th>
            <th>Name</th>
            <th>Email</th>
            <th>Course</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>101</td>
            <td>Vinay jai</td>

```

```
<td>vinay@gmail.com</td>
<td>BCA</td>
</tr>
<tr>
<td>102</td>
<td>Vinay jai</td>
<td>vinay@gmail.com</td>
<td>BCA</td>
</tr>
<tr>
<td>103</td>
<td>Vinay jai</td>
<td>vinay@gmail.com</td>
<td>BCA</td>
</tr>

</tbody>
</table>

<footer>
    <p>Contact us: intellipaat@mail.com | +91 99999-99999</p>
    <p>© 2025 Student Information Portal</p>
</footer>
</body>
```

# Student Information Portal

[Home](#) | [Add Student](#) | [Student List](#) | [Contact](#)

---

## Add Student

Student Name:

Roll Number:

Email:

Course:

BCA 

[Add Student](#)

## Student List

Roll No	Name	Email	Course
101	Vinay jai	vinay@gmail.com	BCA
102	Divya Verma	divya@gmail.com	B.Tech
103	raj singh	raj@gmail.com	MCA

Contact us: [intellipaat@mail.com](mailto:intellipaat@mail.com) | +91 99999-99999

© 2025 Student Information Portal

# CSS

Cascading style sheet.

It is a language that is used to describe the style of a document.

It controls the flow and the color.

## Basic format of css

```
h1 {  
    color : "red";  
}
```

h1 is selector

color is property name

red is property value.

## How to include css style?

### Inline css

Write style directly in the html tag using style attribute

```
<body>  
    <h1 style="background-color: orange;">inline css</h1>  
</body>
```

### Internal css

In this we use style tag.

We write our css in the style tag which is present in our head of html document

```
<head>
    <!-- inter css -->
    <style>
        h2{
            background-color: aqua;
        }
    </style>

    <title>Document</title>
</head>
<body>
    <h1 style="background-color: orangered;">inline css</h1>
    <h2>internal css</h2>
</body>
```

## External css

Here we write all our css in a separate css file and link it with our html document with the help of link tag

```
<head>
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
```

```
index.html      # style.css  X
css > # style.css > h3
1  h3{
2      background-color: palevioletred;
3  }
```

## Color property

Used to set the color of foreground

```
h1{  
    color: maroon;  
}
```

color property

## Background color property.

Used to set the background color.

```
h1{  
    color: maroon;  
    background-color: pink;  
}
```

color property

## Text properties.

Text-align

Font-weight

text-decoration

Line height - control the height of the line of text

Letter-spacing - control the horizontal spacing behaviour between text character

Font-size - control the size of the text

```
p{  
    text-align:justify;  
    text-decoration: underline red wavy;  
    font-weight: 1000;  
}
```

text proerites

```
p{  
    text-align:justify;  
    Line-height:2;  
}
```

Loreum ipsum dolor, sit amet consectetur adipisicing elit. Expedita, dicta! Loreum ipsum dolor sit amet consectetur adipisicing elit. Exercitationem, voluptatibus.

A screenshot of a browser window. On the left, there is a dark sidebar containing some CSS code. On the right, the main content area shows the rendered text "text proerites".

```
p{  
letter-spacing: 3px;  
}  
  
h1{  
font-size: 40px;  
}  
  
p{
```

text proerites

## Css units

We have two types of unit

**Absolute - px , pt, pc, cm, mm, in**

**relative - % , em , rem, ch , vh, vw, etc**

## Font family

Specifies a prioritized list of one or more font family names

A screenshot of a browser window. On the left, there is a dark sidebar containing some CSS code. On the right, the main content area shows the rendered text "text proerites". A dropdown menu is open, listing various font families.

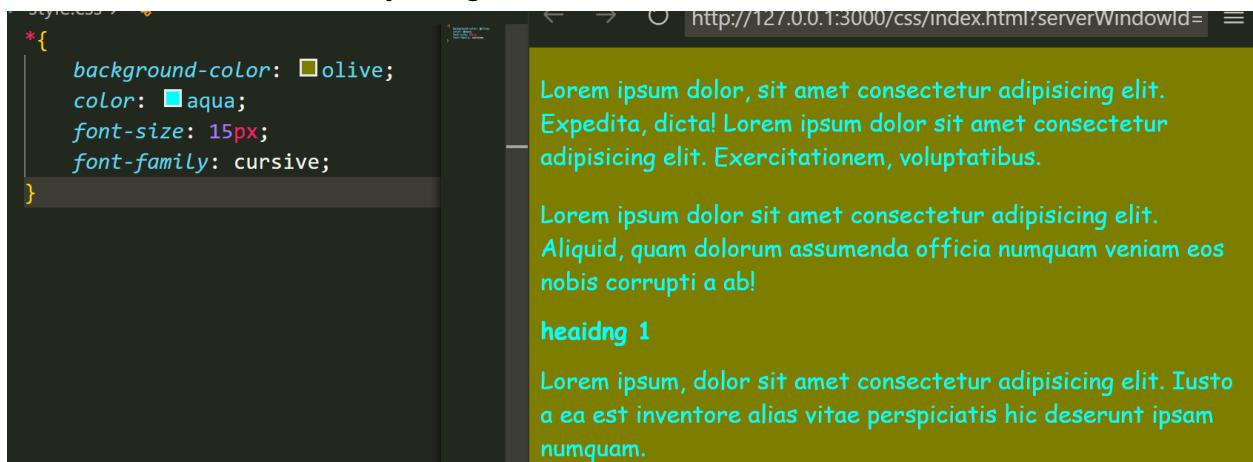
```
p{  
font-family:cursive;  
}
```

Text content: Lorem ipsum dolor, sit amet consectetur adipisicing elit. Expedita, dicta! Lorem ipsum dolor sit amet consectetur adipisicing elit. Exercitationem, voluptatibus.

- ☛ 'Courier New', Courier, monospace
- ☛ 'Franklin Gothic Medium', 'Arial Narrow', Arial, ...
- ☛ 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet ...
- ☛ 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Gra...
- ☛ 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif
- ☛ 'Times New Roman', Times, serif
- ☛ 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Gr...
- ☛ Arial, Helvetica, sans-serif
- ☛ Cambria, Cochin, Georgia, Times, 'Times New Roman...
- ☛ Georgia, 'Times New Roman', Times, serif
- ☛ Impact, Haettenschweiler, 'Arial Narrow Bold', sa...
- ☛ Verdana, Geneva, Tahoma, sans-serif
- ☛ cursive

## Universal selector

It is used to select everything in a document.



A screenshot of a web browser window displaying a CSS file and its corresponding HTML output. On the left, the CSS code for the universal selector (\*) is shown:

```
*{  
background-color: olive;  
color: aqua;  
font-size: 15px;  
font-family: cursive;  
}
```

The right side shows the rendered HTML with the following content:

http://127.0.0.1:3000/css/index.html?serverWindowId=

Heading 1

Heading 2

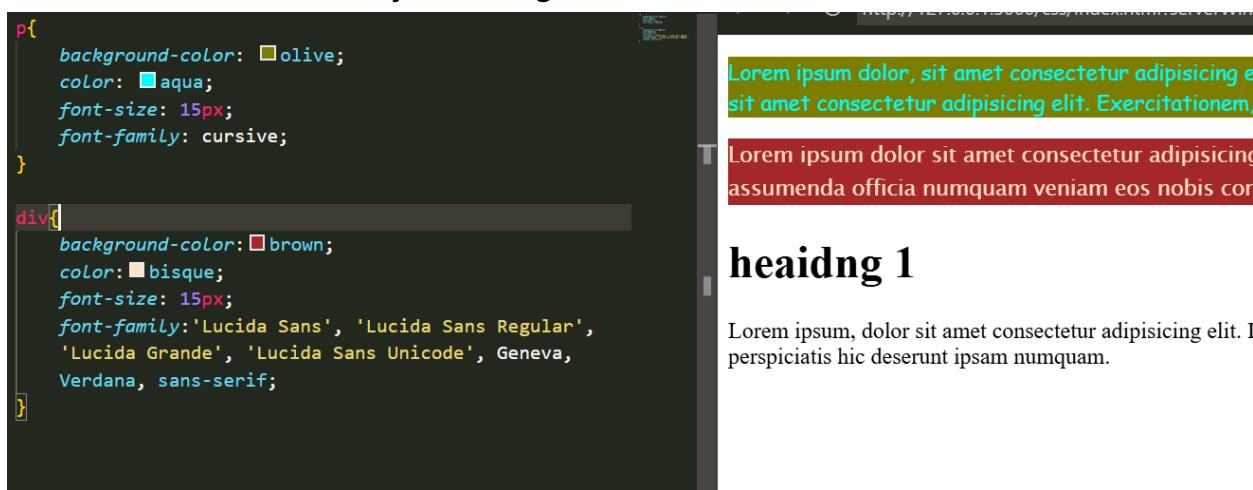
Heading 3

Text

The entire page content is styled with olive background and aqua text.

## Named selector

Select all the element by their tag name



A screenshot of a web browser window displaying a CSS file and its corresponding HTML output. On the left, the CSS code for the p and div tags is shown:

```
p{  
background-color: olive;  
color: aqua;  
font-size: 15px;  
font-family: cursive;  
}  
  
div{  
background-color: brown;  
color: bisque;  
font-size: 15px;  
font-family:'Lucida Sans', 'Lucida Sans Regular',  
'Lucida Grande', 'Lucida Sans Unicode', Geneva,  
Verdana, sans-serif;  
}
```

The right side shows the rendered HTML with the following content:

Heading 1

Text

The heading and text elements are styled with brown background and bisque text.

## Id selector

Select an element based on the value of the elements id attribute

```
#idAttributeName{  
// properties  
}
```

## Class selector

It select an element based on their class attribute

```
.ClassAttributeName{  
    // properties  
}
```

The screenshot shows a browser window displaying a page at <http://127.0.0.1:5000/css/index.html>. On the left, the browser's developer tools are open, showing the CSS code:

```
/* id selector */  
#para{  
    background-color: orange;  
    color: purple;  
    font-size: 20px;  
    font-family: monospace;  
}  
  
/* class slector */  
  
.div1{  
    background-color: aqua;  
    font-size: 20px;  
    font-family: fantasy;  
    color: orangered;  
}
```

On the right, the browser displays two paragraphs of text. The first paragraph, which has the class ".div1", has an orange background, purple text, a font size of 20px, and a monospace font family. The second paragraph, which has the ID "#para", has a purple background, orange text, a font size of 20px, and a monospace font family.

## Pseudo elements

A keyword added to a selector that specifies a special state of that selected elements

:hover  
:active  
:checked  
:nth-of-type

The screenshot shows a browser window displaying a page at <http://127.0.0.1:5000/css/index.html>. On the left, the browser's developer tools are open, showing the CSS code:

```
p {  
    color: bisque;  
    background-color: gray;  
}  
  
p:hover{  
    background-color: black;  
}
```

On the right, the browser displays a single paragraph of text. When the mouse hovers over the paragraph, its background color changes from gray to black, while the text color remains white (bisque).

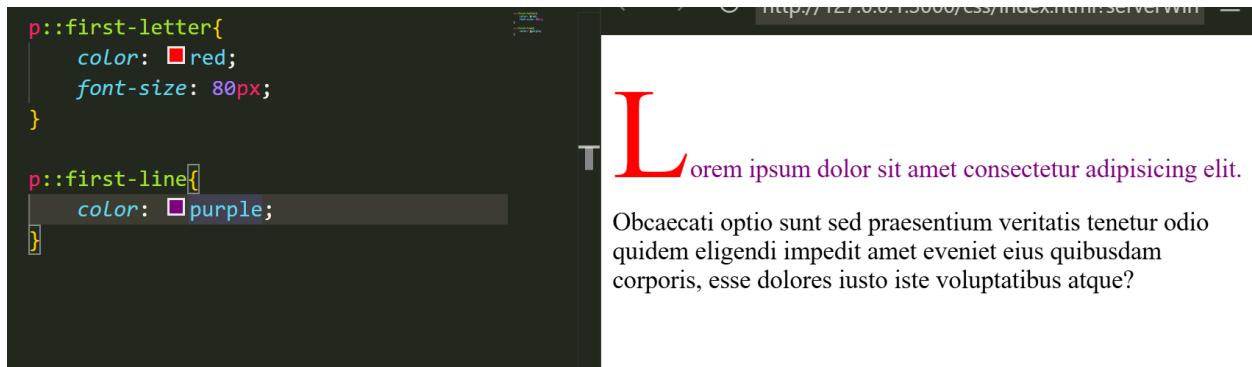
## Pseudo elements

A keyword added to a selector that lets you to style a specific part fo the selected element.

::frist-letter

::frist-line

::selection



The screenshot shows a browser window displaying a page at <http://127.0.0.1:5000/css/index.html>. The page contains the following CSS code:

```
p::first-letter{  
    color: red;  
    font-size: 80px;  
}  
  
p::first-line{  
    color: purple;  
}
```

The output on the page is: **L**orem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati optio sunt sed praesentium veritatis tenetur odio quidem eligendi impedit amet eveniet eius quibusdam corporis, esse dolores iusto iste voluptatibus atque?

## Cascading in css

The css cascading algorithm job is to select css declaration in order to determine the correct value for css property.

## Box model in css

**Height**

**Width**

**Border**

**Padding**

**Margin**

**Height - :** by default it sets the content height of the element .

**Width -:** it define the content area width of the element .

**border -:** used to set element border.

Border width  
Border style  
Border color

**Border radius** :- used to round the corner of an element outer border edge.

**Padding** :- it help to give the inner spacing in the box between border and the content

padding - left  
Padding- right  
padding - bottom  
Padding-top

### **Padding shortcuts**

Padding : top , right, left, bottom;  
padding : top-bottom , left- right;  
Padding top , left - right, bottom ;

**Margin** :- to set the spacing of out side of the border

Margin - top  
Margin - bottom  
Margin - left  
Margin - right

**Margin shorthand is same as padding shorthand.**

The screenshot shows a web browser window with the URL <http://127.0.0.1:3000/css/index.html>. Inside the browser, there are two orange rectangular boxes containing placeholder text (Lorem ipsum dolor...). To the left of the browser, a code editor displays the following CSS code:

```
/* style.css */ > div{  
    height: 200px;  
    width: 200px;  
    background-color: orange;  
    /* border-width: ;  
    border-style: ;  
    border-color: ; */  
    border: 4px solid red;  
    border-radius: 10px;  
    text-align: justify;  
    padding-top: 10px;  
    padding-left: 10px;  
    padding-bottom: 10px;  
    padding-right: 10px;  
  
    margin-top: 40px;  
    margin-left: 40px;  
    margin-bottom: 40px;  
}
```

**Task :-** create a simple traffic light ;

The screenshot shows a code editor with the following HTML code:

```
<div id="traffic">  
    <div id="red"></div>  
    <div id="yellow"></div>  
    <div id="green"></div>  
</div>
```

```
# style.css > #traffic
#traffic{
    height: 210px;
    width: 80px;
    background-color: black;
    padding: 10px 0px;
}

#red{
    height: 60px;
    width: 60px;
    background-color: red;
    border-radius: 50%;
    margin: 0px auto;
    border: 2px solid white;
}

#yellow{
    height: 60px;
    width: 60px;
    background-color: yellow;
    border-radius: 50%;
    margin: 10px auto;
    border: 2px solid white;
}

#green{
    height: 60px;
    width: 60px;
    background-color: green;
    border-radius: 50%;
    margin: 10px auto;
    border: 2px solid white;
}
```



## Task :- smily face

```
<div id="face">
    <div id="eye_container">
        <div id="left"></div>
        <div id="right"></div>
    </div>

    <div id="smile"></div>
```

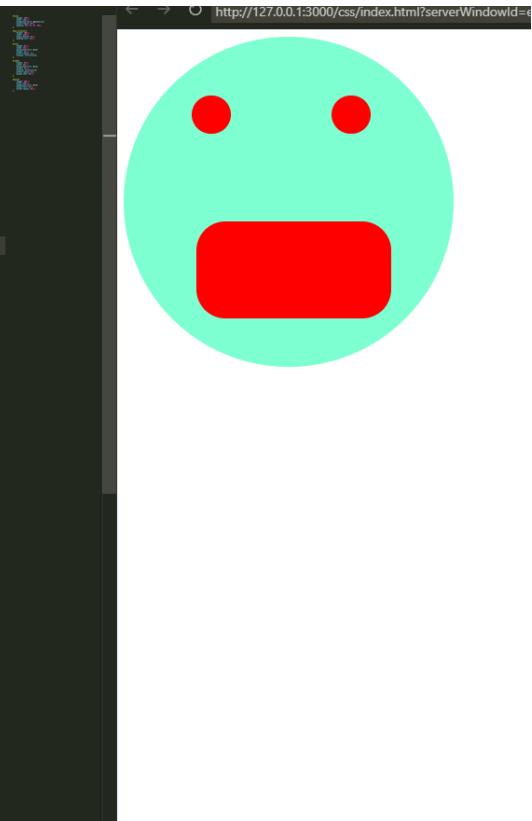
```
styles.css - eye_container
#face{
    height: 300px;
    width: 300px;
    background-color: aquamarine;
    border-radius: 50%;
    padding: 40px 0px 0px 40px;
}

#eye_container{
    height: 100px;
    width: 250px;
    margin-bottom: 50px;
    padding-left: 10px;
}

#left{
    height: 40px;
    width: 40px;
    background-color: red;
    margin: 20px;
    border-radius: 50%;
    display: inline-block;
}

#right{
    height: 40px;
    width: 40px;
    background-color: red;
    margin: 20px;
    display: inline-block;
    border-radius: 50%;
    margin-left: 80px;
}

#smile{
    height: 100px;
    width: 200px;
    background-color: red;
    margin-left: 35px;
    border-radius: 30px;
}
```



## Inline and block

**Block** - takes up the full width available on page and always start with new line.

**Inline** - they only take necessary width and don't start with new line.

## Css transitions.

transitions enable you to define the transition between 2 state of an element.

```
> # style.css > 9 div:hover
1 div{
2     height: 200px;
3     width: 200px;
4     background-color: orange;
5     transition: 2s;
6 }
7 }
8 div:hover {
9     background-color: greenyellow;
10    border-radius: 50%;
```



## Transition shorthand

Property name | duration | timing function | delay

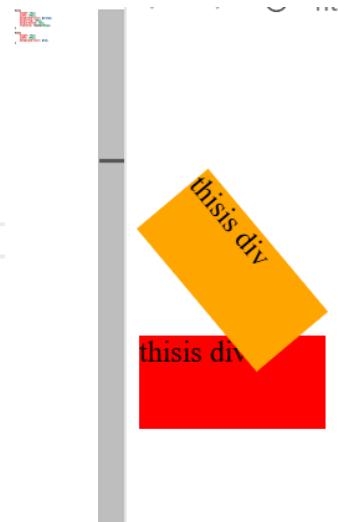
## Css transform

This property lets you **rotate, scale, skew or translate** an element

Rotate (45deg)

```
#one{
    height: 50px;
    width: 100px;
    background-color: orange;
    margin-top: 100px;
    margin-bottom: 10px;
    transform: rotate(50deg);
}

#two{
    height: 50px;
    width: 100px;
    background-color: red;
```



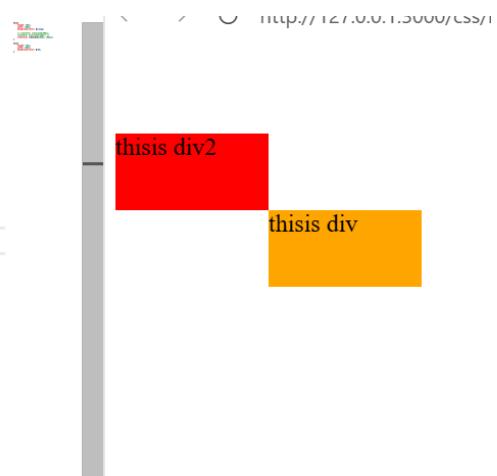
Scale :

```
#one{  
    height: 50px;  
    width: 100px;  
    background-color: orange;  
    margin: 10px auto;  
    transform: scale(2.5);  
}  
  
#two{  
    height: 50px;  
    width: 100px;  
    background-color: red;  
}
```



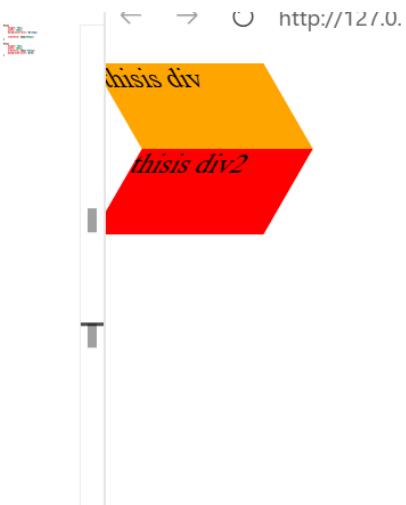
## Translate

```
#one{  
    height: 50px;  
    width: 100px;  
    background-color: orange;  
  
    /* transform: translateX(50px);  
     * transform: translateY(10px); */  
    transform: translate(100px, 100px);  
}  
  
#two{  
    height: 50px;  
    width: 100px;  
    background-color: red;  
}
```



## Skew

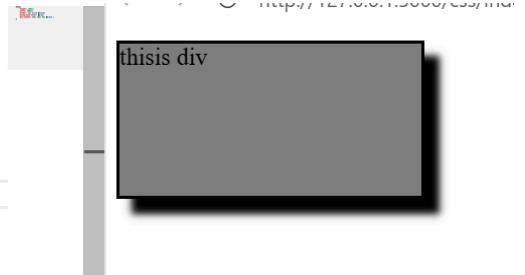
```
#one{  
    height: 50px;  
    width: 100px;  
    background-color: orange;  
  
    transform: skew(30deg);  
}  
  
#two{  
    height: 50px;  
    width: 100px;  
    transform: skew(-30deg);  
    background-color: red;  
}
```



## Box shadow

It adds shadow effect around an element frame

```
#one{  
    height: 100px;  
    width: 200px;  
    border: 2px solid black;  
    background-color: gray;  
    box-shadow: 10px 10px 5px black ;  
}
```



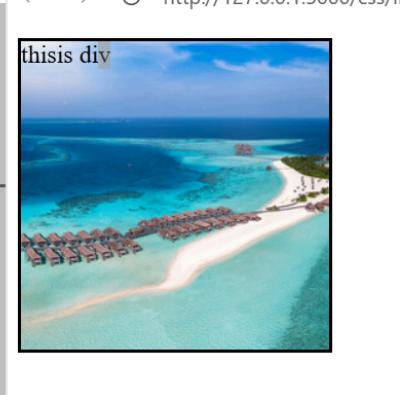
## Background image

Lets you set an image in a background

background-image : url("")

Background-size : contain / cover/ auto;

```
#one{  
    height: 200px;  
    width: 200px;  
    border: 2px solid black;  
    background-image: url('travel.jpg');  
    /* background-size: contain; */  
    /* background-repeat: no-repeat; */  
    background-size: cover;  
    background-position: center;  
}
```



Task - : make a card hover effect

```
#card{  
    height: 120px;  
    width: 100px;  
    border: 1px solid black;  
    border-radius: 4px;  
    background-image: url('travel.jpg');  
    background-size: cover;  
    background-position: center;  
    transition: 1s;  
}  
  
#card:hover{  
    box-shadow: 8px 8px 6px gray;  
}
```



Task : make a card rotate effect

style.css / ⑤ card:hover

```
#card{  
    height: 120px;  
    width: 100px;  
    border: 1px solid black;  
    border-radius: 4px;  
    background-image: url('travel.jpg');  
    background-size: cover;  
    background-position: center;  
    transition: 1s;  
}
```

```
#card:hover{  
    box-shadow: 8px 8px 6px gray;  
    transform: rotate(360deg);  
}
```



## Display property.

Display : Inline | block | inline-block

If display property is inline it will behave like inline element

If display property is block it will behave block element

If display is inline-block it will posses both inline and block element property .

```

body{
    width: 100vw;
    height: 100vh;
}

#one{
    background-color: red;
    display: inline-block;
    width: 100px;
    height: 40px;
    border: 2px solid black;
}

#two{
    height: 40px;
    width: 100px;
    border: 2px solid orange;
    display: inline-block;
}

span{
    border: 2px solid green;
    height: 40px;
    width: 100px;
    display: inline-block;
}

img{
    width: 100px;
    height: 100px;
    display: block;
}

```



## Positions in css

The position in css sets how element is positioned in a document

The top, left, right and bottom properties determine the final location of element.

- static
- relative
- absolute
- fixed

### Static

When position is static it will take its default value.

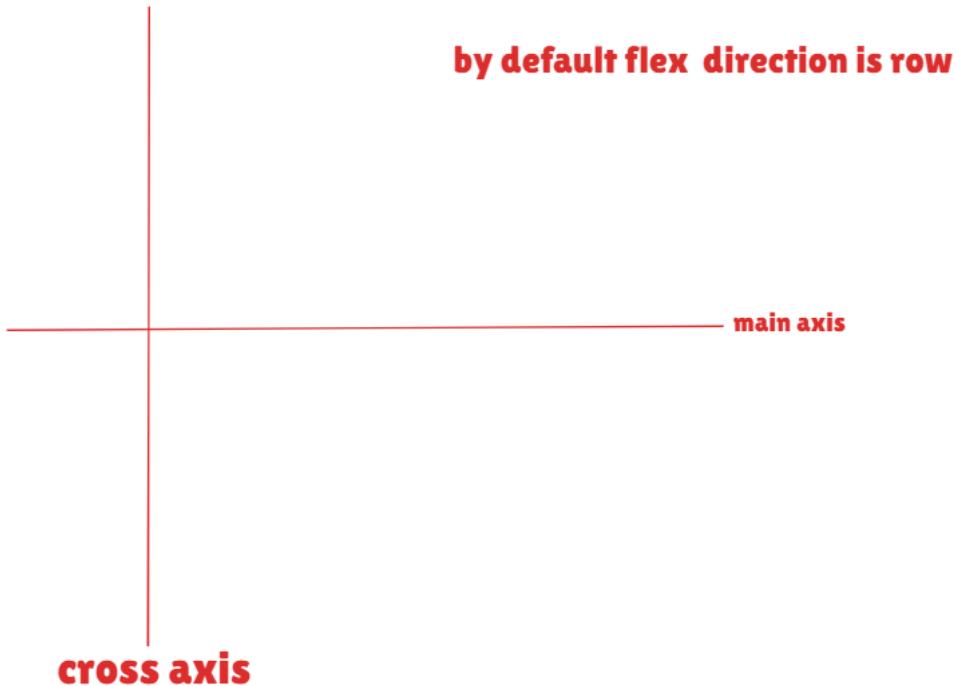
And all the top, right, bottom, left property will not work on it.

## Flex box

Flexible box layout

It is a one-dimensional layout method for arranging items in row or columns.

# the flex model



## Flexbox direction

It sets how flex items are placed in the flex container, along which axis and direction

- flexbox-direction: row (main, left to right) (default)
- flexbox-direction: row reverse (main, right to left)
- flexbox-direction: column (cross, top to bottom)
- flexbox-direction: column- reverse (main, bottom to top)

```

<body>

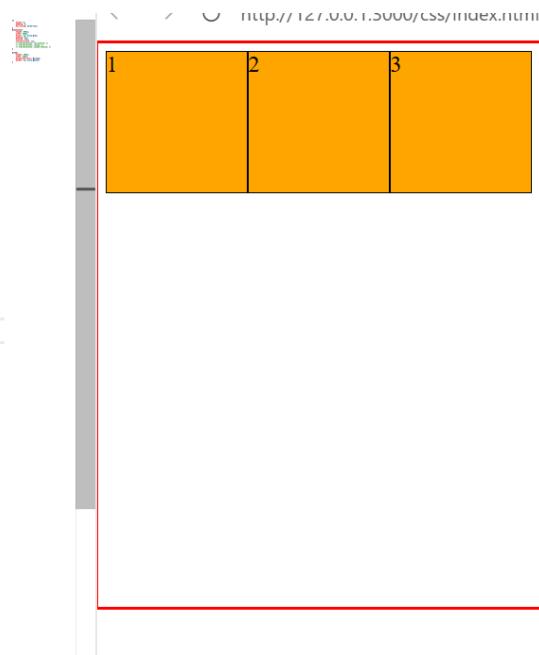
    <div id="container">
        <div id="item">1</div>
        <div id="item">2</div>
        <div id="item">3</div>
    </div>

</body>

/*
margin: 0;
padding: 0;
box-sizing: border-box;
}
#container{
height: 400px;
width: 400px;
border: 2px solid red;
padding: 5px;
display: flex;
flex-direction: row;
/* flex-direction: row-reverse; */
/* flex-direction: column; */
/* flex-direction: column-reverse; */
}

#item{
height: 100px;
width: 100px;
background-color: orange;
border: 1px solid black;
}

```



The screenshot shows a browser window displaying a simple flex layout. Inside a red-bordered container, there are three orange squares arranged horizontally. Each square contains a number: '1' in the first square, '2' in the second, and '3' in the third. This visual representation corresponds to the 'row' direction specified in the CSS code.

## Justify content.

It sets how flex items are placed in the flex container along your main axis (left to right or right to left)

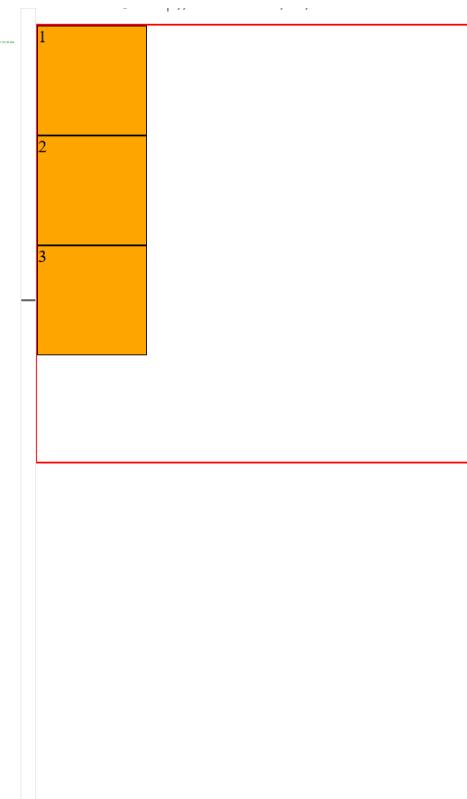
Justify-content : flex-start | flex-end | center

## Align items

It distribute our items along the cross axis.

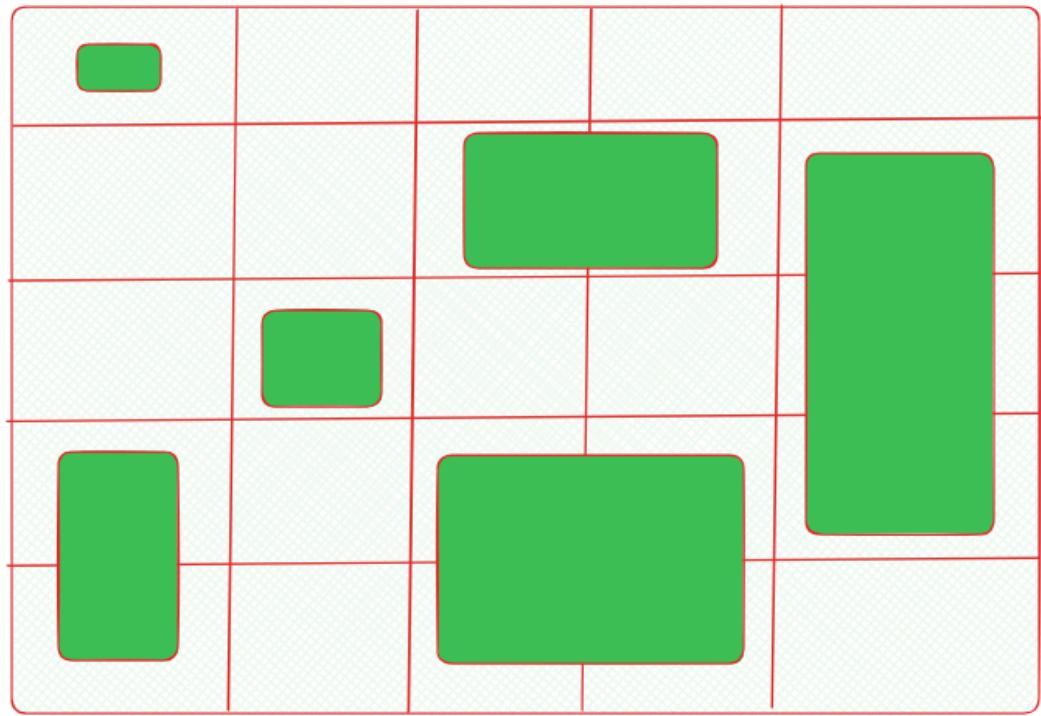
Align-item: flex-start | center | flex-end

```
*{  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
#container{  
    height: 400px;  
    width: 400px;  
    border: 2px solid red;  
    /* padding: 5px; */  
    display: flex;  
    /* flex-direction: row; */  
    /* flex-direction: row-reverse; */  
    flex-direction: column;  
    /* flex-direction: column-reverse; */  
  
    /* justify-content: flex-start; */  
    /* justify-content: flex-end; */  
    /* justify-content: center; */  
    /* in space around the sapce between the box is doubel form the space  
    in start and end */  
    /* justify-content:space-around; */  
    /* justify-content: space-between; */  
    /* justify-content: space-evenly; */  
  
    /* align-items: flex-start; */  
    /* align-items: flex-end; */  
    /* align-items: center; */  
}  
  
#item{  
    height: 100px;  
    width: 100px;  
    background-color: orange;  
    border: 1px solid black;  
}
```

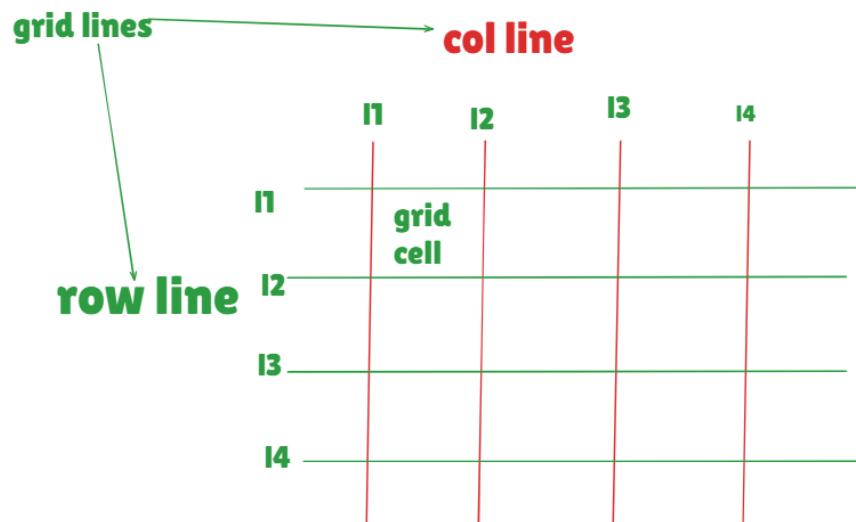


## grid

Setting a container display to grid will make all the items inside arranged in 2d layout



**grid model**



## **Grid templates :**

Used to define row and column to grid.

→grid-template-row:

→ grid-template-columns:

## **Grid row and grid column:**

They are used to define items starting and ending position in the column and row

grid-column-start: line number

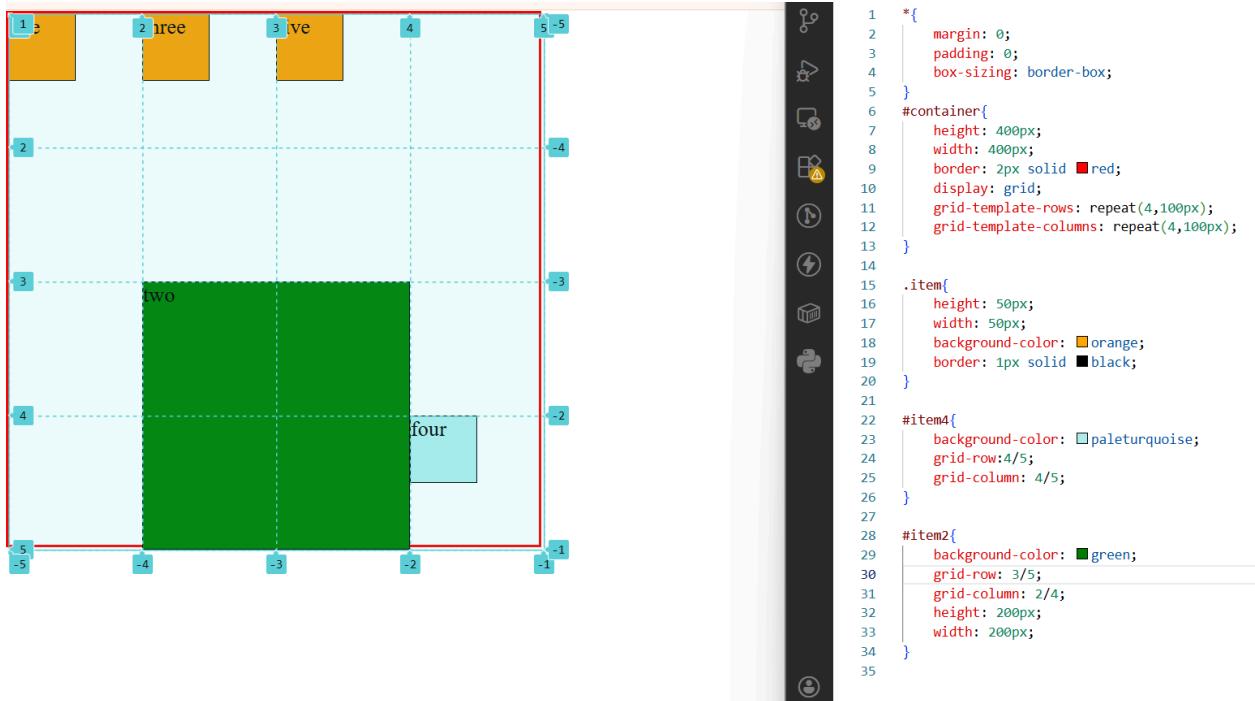
grid-column-end: line number

grid-column : start line / end line

grid-row-start: line number

grid-row-end: line number

grid-row: start line / end line



Task : hovering card

```

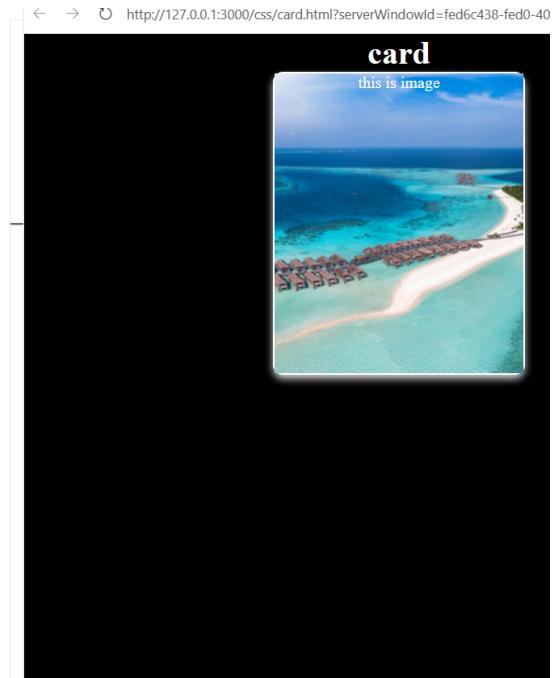
<body>
  <h1>card</h1>
  <div id="card">
    <div class="image">this is image</div>
  </div>
</body>

```

```

1  *{
2    margin: 0;
3    padding: 0;
4    box-sizing: border-box;
5    color: white;
6    background-color: black;
7    text-align: center;
8  }
9
10
11 #card{
12   height: 200px;
13   width: 250px;
14   border: 2px solid white;
15   border-radius: 10px;
16   background-color: white;
17   box-shadow: 0 4px 8px gainsboro;
18   margin: 0 auto;
19   transition: 0.5s ease;
20 }
21
22 .image{
23   width: 100%;
24   height: 100%;
25   background-image: url('travel.jpg');
26   background-position: center;
27   background-size: cover;
28   transition: 0.5s ease;
29 }
30
31 #card:hover{
32   height: 300px;
33 }
34

```



## Task design feedback for

```

*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body{
  font-family:cursive;
  padding: 20px;
  background-color: #e7e4e4;
}

h1, p{
  text-align: center;
  margin-bottom: 10px;
}

form{
  max-width: 400px;
  margin: auto;
  padding: 20px;
  background-color: #ffff;
  border-radius: 10px;
  box-shadow: 0px 5px 10px black;
}

label, p{
  font-weight: bold;
}

input, textarea{
  width: 100%;
  padding: 10px;
  margin-top: 5px;
  margin-bottom: 10px;
  border-radius: 10px;
}

button{
  width: 100px;
  padding: 10px;
  color: white;
  background-color: green;
  font-weight: bold;
  font-family: cursive;
  border-radius: 10px;
}

input[type="radio"]{
  margin-right: 5px;
}

```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Customer feedback</title>
7      <link rel="stylesheet" href="form.css">
8  </head>
9  <body>
10     <h1>Customer Feedback Form</h1>
11     <p>Please fill out this form to share your feedback with us.</p>
12
13     <form action="#">
14
15         <label for="name">Name : </label><br>
16         <input type="text" id="name" name="userName"><br>
17         <br>
18         <label for="email">Email : </label><br>
19         <input type="email" id="email" name="email"><br>
20         <br>
21         <label for="rating">Rate our service (1 to 10):</label><br>
22         <input type="range" min="1" max="10" id="rating" name="userRating"><br>
23         <br>
24         <label for="comment">comment : </label><br>
25         <textarea name="comment" id="comment"></textarea><br>
26         <br>
27         <p>Would you recommend us?</p>
28         <input type="radio" name="recommend" value="yes" id="yes">
29         <label for="yes">Yes</label>
30         <input type="radio" name="recommend" value="no" id="no">
31         <label for="no">No</label>
32
33         <br><br>
34         <button type="submit">Submit Feedback</button>
35
36     </form>
37 </body>
38 </html>
```

## Css animations.

To animate the elements

```
@keyframe name {  
    From{ // do something}  
    To{ // do something}  
}
```

animation-name  
animation-duration  
animation-timing-function  
animation-delay  
animation-iteration-function  
animation-direction

`animation-direction: normal;`

`animation-direction: reverse;`

`animation-direction: alternate;`

`animation-direction: alternate-rev`

>



```
*{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}

#container{
    height: 100px;
    width: 200px;
    background-color: greenyellow;
    animation-name:apple;
    animation-duration: 2s;
    animation-timing-function: ease-in;
    animation-delay:0;
    animation-iteration-count: 3;
    animation-direction: alternate-reverse;
    /* animation: animation-name duration timing function delay
    iteration-count direction */
}

@keyframes fontAnime{
    from{
        font-size: 20px;
    }
    to{
        font-size: 50px;
    }
}

@keyframes apple{
    from{
        background-color: greenyellow;
        height: 200px;
        width: 200px;
    }
    to{
        background-color: red;
        height: 500px;
        width: 500px;
    }
}
```

```
<body>

    <div id="container">
        Lorem ipsum dolor sit amet.
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Nesciunt,
        qui repellendus magni vero ratione, deserunt quis vitae laborum
        corrupti distinctio ad sunt! Autem placeat corporis maxime ea? Quis
        sed minima voluptatibus tempore alias culpa quibusdam at natus unde
        autem nemo perferendis quia, architecto, provident officia repellat
        commodi magnam error nam.|</div>

</body>
```

## Media queries.

Help create a responsive website.

```
@media(width (viewport width)){
    div{
        color= “red”
    }
}
```

hello

```
div{
    background-color: ■ red;
    color: □ white;
}

@media (width:500px) {
    div{
        background-color: ■ yellow;
        color: ■ black;
        height: 100px;
    }
}
```

Min-width  
Max-width

hello

292px × 632px

hello

455.33px × 632px

hello

762px × 632px

**Product catalog page.**

## Product Catalog

Explore our latest collection of amazing products. Click "Buy Now" to visit the product page.

Product Image	Product Name	Price	Buy Now
	Wireless Headphones	49.99	<a href="#">Buy Now</a>
	Smart Watch	79.99	<a href="#">Buy Now</a>
	Smart phone	39.99	<a href="#">Buy Now</a>

```
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
body{  
    font-family: Arial, sans-serif;  
    box-flex-group: #f4f4f4;  
    padding: 20px;  
}  
  
h1{  
    text-align: center;  
    color: #333;  
}  
  
p{  
    text-align: center;  
    color: #555;  
    margin-bottom:20px;  
}  
  
table{  
    width: 100%;  
    border-collapse: collapse;  
    background: white;
```

```
    box-shadow: 0px 0px 10px grey;
}

th{
    background-color: #333;
    color: #fff;
    padding: 12px;
    font-size: 20px;
}

td{
    text-align: center;
    padding: 13px;
    border-bottom: 1px solid #ddd;
}

img{
    width: 120px;
    height: auto;
    border-radius: 10px;
}

a{
    padding: 10px 14px;
    background-color: #4595eb;
    color: #fff;
    text-decoration: none;
    border-radius: 10px;
}

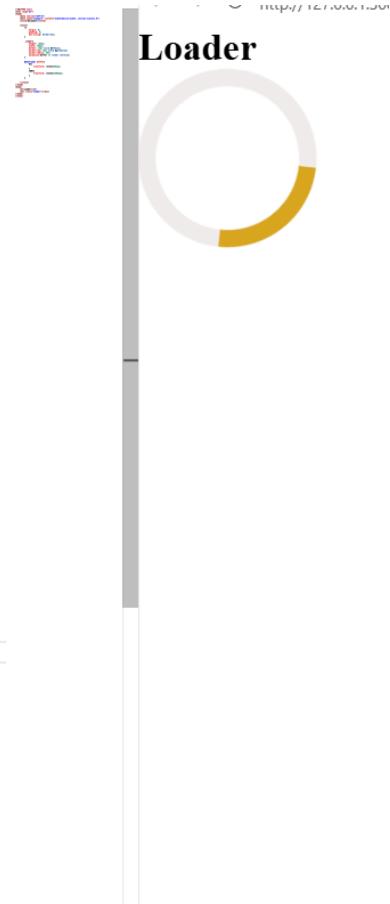
a:hover{
    background-color: #0e4b8d;
}
```

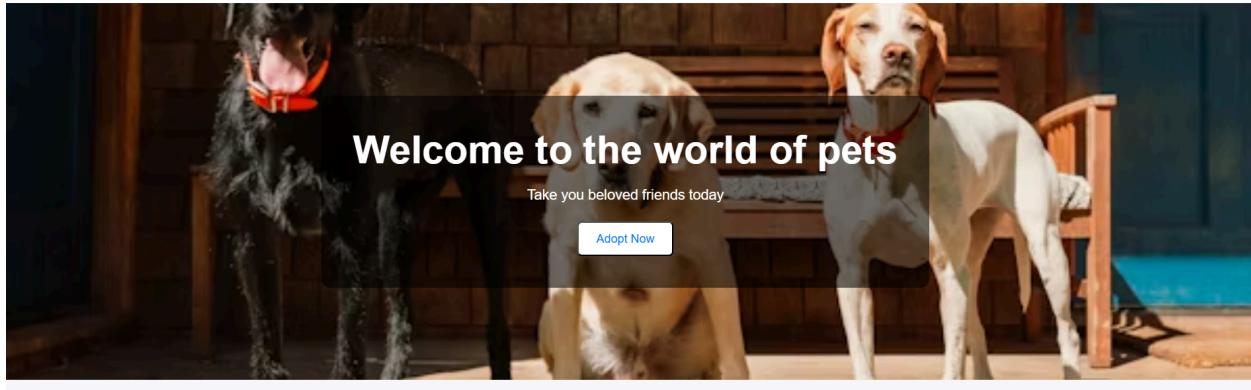
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        .loader{
            height: 150px;
            width: 150px;
            border: 15px solid #f4ecec;
            border-top: 15px solid goldenrod;
            border-radius: 50%;
            animation:buffer 2s linear infinite;
        }

        @keyframes buffer{
            0%{
                transform: rotate(0deg);
            }
            100%{
                transform: rotate(360deg);
            }
        }
    </style>
</head>
<body>
    <h1>Loader</h1>
    <div class="loader"></div>
</body>
</html>
```



**jerry**

A playful and loving dog, recently rescued and looking for a forever home.

**kenny**

A gentle rescued cat who enjoys cuddles and calm spaces.

**saidey**

A sweet rescued cat with a big heart and lots of affection to give.

## Css code

```
body{  
    margin: 0;  
    font-family: cursive;  
    background-color: #f8f9fa;  
}  
  
.navbar{  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    padding: 15px 40px;  
    background-color: #fff;  
    box-shadow: 0 2px 5px grey;
```

```
}

.logo{
    font-size: 25px;
    font-weight: bolder;
    color: #333;
}

.nav-links{
    list-style-type: none;
    display: flex;
    gap: 25px;
}

.nav-links a{
    text-decoration: none;
    color: #333;
    font-weight: 600;
}

.nav-links a:hover{
    color: #104c8c;
}

.hero{
    height: 60vh;
    background: url('thumb.avif') center/cover no-repeat;
    display: flex;
    justify-content: center;
    align-items: center;
}

.overlay{
    background-color: rgba(0, 0, 0, 0.55);
    padding: 40px;
    text-align: center;
    color: #fff;
    border-radius: 10px;
}
```

```
}

.hero h1{
    font-size: 45px;
    margin: 0 0 10px;
}

.hero p{
    margin-bottom: 25px;
}

.btn{
    background: #fff;
    color: #104c8c;
    padding: 10px 20px;
    border-radius: 10px;
    transition: 0.3s ease-in;
}

.btn:hover{
    background-color:#104c8c;
    color: white;
}

.cards{
    display: flex;
    justify-content: center;
    gap: 25px;
    margin: 50px;
    flex-wrap: wrap;
}

.card{
    width: 300px;
    background: #fff;
    border-radius: 12px;
    overflow: hidden;
    text-align: center;
    box-shadow: 0 4px 8px gray;
    transition: 0.3s;
```

```

}

.card:hover{
    transform: translateY(-5px);
}

.card img{
    width: 100%;
    height: 200px;
    object-fit: cover;
}

.card h3{
    margin: 15px 0 5px;
}

.card p{
    padding: 0 15px 20px;
}

footer{
    text-align: center;
    padding: 20px;
    box-shadow: 0 -2px 5px gray;
    background: #fff;
}

```

## Html code

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Modern Web Page</title>
    <link rel="stylesheet" href="style.css">
</head>

```

```
<body>

<nav class="navbar">
  <h2 class="logo">AdoptME</h2>
  <ul class="nav-links">
    <li><a href="#">Home</a></li>
    <li><a href="#cards">Adopt</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>

<section class="hero">
  <div class="overlay">
    <h1>Welcome to the world of pets</h1>
    <p>Take you beloved friends today</p>
    <button class="btn">Adopt Now</button>
  </div>
</section>

<section class="cards" id="cards">

  <div class="card">
    
    <h3>jerry</h3>
    <p>A playful and loving dog, recently rescued and looking for a forever home.</p>
  </div>

  <div class="card">
    
    <h3>kenny</h3>
    <p>A gentle rescued cat who enjoys cuddles and calm spaces.</p>
  </div>

  <div class="card">
    
    <h3>saidey</h3>
    <p>A sweet rescued cat with a big heart and lots of affection to give.</p>
  </div>
</section>
```

```
</div>

</section>

<footer id="contact">
  <p>© 2025 AdoptME | All Rights Reserved | adoptME@mail.com</p>
</footer>

</body>
</html>
```

# JavaScript

It is a programming language that is used give the instruction to our computer.

Input → computer → output.

The most simplest way to run our js code in browser console.

We can also run our js code in our computer system directly with help of nodeJS

nodeJS is environment in the which js code can run directly in our system.

To write our js code we need some IDE and here we are going to use the vscode.

With help of node js we can directly run our js code in our computer system

```
PS C:\Users\DELL\Desktop\resvision session\js> node .\script.js  
hello world
```

And if we want to run the code in our browser console we first have to inject our js in the html file with help of script tag

```
<script src="script.js"></script>
```

---

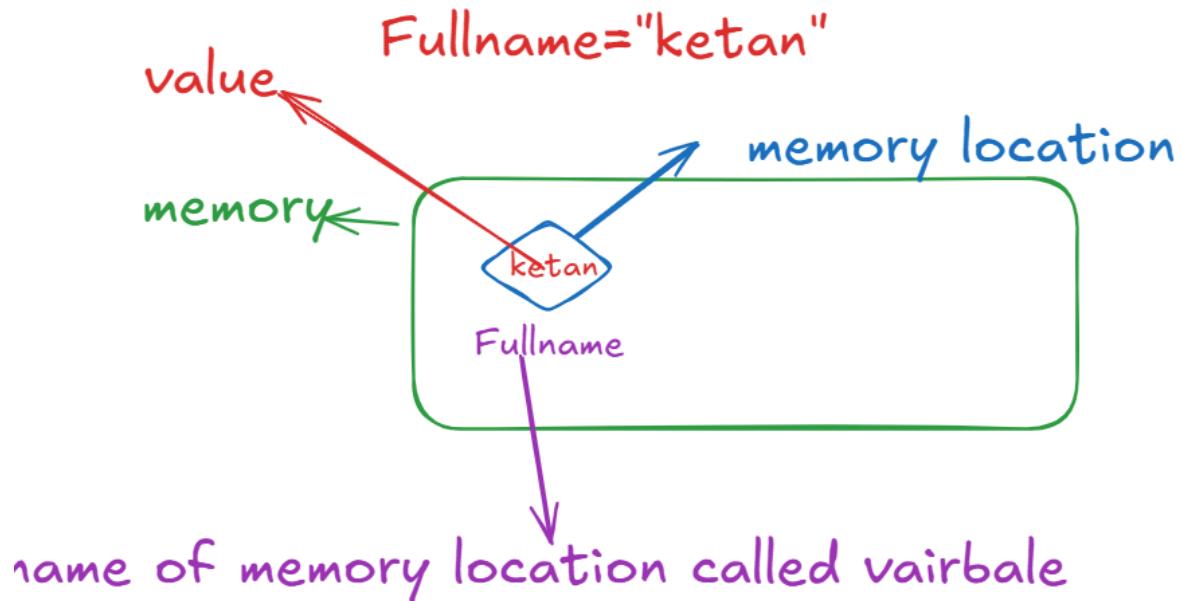
hello world

---

Live reload enabled.

### **Variables in js.**

Variables are simply names of the container location for data.  
For eg:-



```

fullname = "miraj" // string type of value
age = 23 // number type of value
price = 99.9 // number type of value
pi = 3.14 // number type of value
isGood = true // boolean type of value
a = null //the null value represent intentional absence of value
b = undefined // a variable the has not been assigmed any value is of type undefined.
c = NaN // the NaN gloabal property is a value the represent not a number

```

```
console.table([typeof fullname, typeof age, typeof price, typeof pi, typeof isGood, typeof a, typeof b, typeof c])
```

(index)	Values
0	'string'
1	'number'
2	'number'
3	'number'
4	'boolean'
5	'object'
6	'undefined'
7	'number'

## **Variable rules n js.**

- 1.Variable are case sensitive means “a” and “A” are different
- 2.Only letters, digits, underscore(\_) and \$ are allowed to be in variable name (not even a space also)
- 3.Only letters underscore, and \$ sign should be allowed as 1st character,
- 4.reserved keywords words cannot be variable name,

## **let, var and const keywords in js.**

**var :** variable can be re- assigned and updated in this and it is a global scope variable

```
var a = "mango"
console.log(a)      ● PS C:\Users\[
                      mango
var a = true        true
console.log(a)      ✦ PS C:\Users\[
```

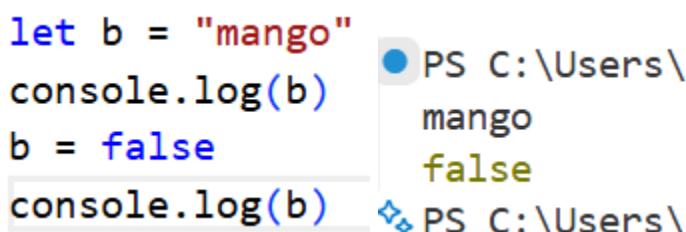
**let :** variables cannot be re-declared but they can be updated.  
And it is a block scope variable.

```
let b = "mango"
console.log(b)
let b = false
console.log(b)

let b = false
^

SyntaxError: Identifier 'b' has already been declared
  at wrapSafe (node:internal/modules/cjs/loader:1662:18)
  at Module._compile (node:internal/modules/cjs/loader:1704
  at Object . (node:internal/modules/cjs/loader:1005:10)
```

```
let b = "mango"
console.log(b)
b = false
console.log(b)  PS C:\Users\
```



```
mango
false
PS C:\Users\
```

**const :** variables can not be re-declared and cannot be updated in this const keyword and this is block scope.

```
const a = "mango"
console.log(a)
const a = false
console.log(a)

Uncaught SyntaxError: Identifier 'a' has already been declared
```

```
const a = "mango"
console.log(a)
a = false
console.log(a)

mango
▶ Uncaught TypeError: Assignment to constant variable.
  at <anonymous>:3:3
```

```
{  
var a = "mango" // gloal scope  
// console.log(a)  
let b = "apple" // block scope  
// console.log(b)  
const c = "banana"  
// console.log(c)  
}  
// console.log(a) // accessible  
// console.log(b) // not accesible  
// console.log(c)// not accessible
```

## Data types in js

2 types of data type in js

### Primitive data types.

Number, string, boolean, undefined, null, bigint, symbol,

### Non primitive data types.

Object and arrays (all about collection)

These collection are mostly stored in form key value pair.

Collection → key : value

Name : “raj”

AGE : 23

ISvOTE : true

```
const person = {  
    name : "raj",  
    age : 34,  
    isVote : true  
}
```

In the collection we have arrays also.

```
let person = ["jai", 34, true]
```

## Comments in js

We have two types of comments.

Single line comment // single line comment

Multiline comment

```
/*  
this  
is  
multi  
line  
comment  
*/
```

## Operators in js.

They are something which is used to perform some operation on our data.

### Arithmetic operator.

**+ , - , \* , / , modulus(%), exponential( \*\*), increment(++) , decrement(--)**

```
let a = 10;
let b = 5;
console.log("a+b = ",a+b)           a+b = 15
console.log("a-b = ",a-b)           a-b = 5
console.log("a*b = ",a*b)           a*b = 50
console.log("a/b = ",a/b)           a/b = 2
console.log("a%b = ",a%b)           a%b = 0
console.log("a**b = ",a**b)         a**b = 100000
```

```
// increment and decrement operator
// increment means increase value by one
// decrement means decrease value by one
let a = 10
a = ++a // a = a+1
console.log(a)                                PS (11

let b = 10
b = --b // b = b-1                           9
console.log(b)                                PS (
```

**In increment and decrement we have something called pre and post increment and decrement**

```
~/Desktop> ...
1 // pre increment and decrement
2 // in this value first updated and then used.
3
4 let a = 10;
5 let b = ++a // pre increment in the first value of a increased and then assigned in b
6 console.log(b) //11
7
8 let c = 10;
9 let d = --c // pre decrement in this first value is decreased and then assigned in d
0 console.log(d) // 9
```

```
//post increment and decrement
// in this value first used and then updated.
```

```
let a = 10;
let b = a++ // val of a assigned in b and then increased means b = 10 , a = 11
console.log("b = ", b , "a = ", a)

let c = 10;
let d = c-- //post decrement in this first value is assigned in d and then decreased means d = 10, c = 9
console.log("d = ", d , "c = ", c)
```

```
PS C:\Users\DELL\Desktop\resvision se
b = 10 a = 11
d = 10 c = 9
. PS C:\Users\DELL\Desktop\resvision se
```

## Assignment operator.

= , +=, -=, \*=, /=, %= , \*\*=

```
// assignment operator

let a = 10; // assigne value 10 in a
// console.log(a)

// a +=10 // a = a+10
// console.log(a) //20

// a -= 10 // a = a-10
// console.log(a) //0

// a *= 4 // a = a*4
// console.log(a) //40

// a /=5 // a = a/5
// console.log(a) //2

// a %= 3 //a = a%3
// console.log(a) //1

a **=10 // a = a **10


---

console.log(a) //10000000000
```

## Comparison operator.

`== , ===, != , !==, >, <, >=, <=`

```

let a = 10
let b = 5

console.log("a == b" , a == b)
console.log("a != b" , a != b)
console.log("a > b" , a > b)
console.log("a < b" , a < b)
console.log("a >= b" , a >= b)
console.log("a >= 10" , a >= 10)
console.log("a <= b" , a <= b)
console.log("5 <= b" , 5 <= b)

```

a == b	false
a != b	true
a > b	true
a < b	false
a >= b	true
a >= 10	true
a <= b	false
5 <= b	true

==== it check the value as well as type also.

```

let a = 2
let b = "2"
console.log("a == b" , a == b); //true
console.log("a === b" , a === b); //false  it check value and type both

```

---

```

- F5 C:\Users\DELL
  a == b true
  a === b false

```

!== return true when value or type both are different

Or false only when both value and type are same.

```

let a = 2
let b = "2"

console.log("a != b" , a!=b) // here "2" --> number 2 an the check 2 != 2 --> false
console.log("a !== b" , a !== b) // here "2" --> not convert in number 2 check 2 !== "2" --> true

```

## Logical operator

Logical AND &&  
 Logical Or ||  
 Logical NOT !

logical AND (&&)		
a	b	a && b
t	t	t
t	f	f
f	t	f
f	f	f

logical OR (  )		
a	b	a    b
t	t	t
t	f	t
f	t	t
f	f	f

logical NOT (!)	
a	!a
F	T
t	F

## Operator precedence.

↑  
 ()  
 \*\*  
 \*, /, %  
 +, -

```

let a = (2+1)*3
console.log(a) //9

let b = 4+1*6/2
console.log("b = ", b) // b = 7
    
```

## Conditional statements.

To implement some condition in the code .

## If statements.

```
script.js > ...
// age program

let age = 27

if(age >=18){
    console.log("vote")
}

if(age < 18){
    console.log("not vote")
}

let age = 12

if(age >=18){
    console.log("vote")
}else{
    console.log("not vote")
}
```

## else-if statements.

```
let age = 90

if(age > 18){
    console.log("vote")
}else if(age == 18){
    console.log("vote")
}else{
    console.log("not vote")
}
```

## Ternary operator.

Condition ? true output : false output;

```
let age = 9
age >= 18? console.log("vote") : console.log("not Vote");
```

## Switch case.

```
switch(condition){
```

```
case :
```

```
    //task
```

```
    Break;
```

```
case :
```

```
    //task
```

```
    Break;
```

Default:

```
    //task
```

```
    break;
```

```
}
```

```
// fruits price mango -- 100 apple-- 200 banana -- 300
```

```
let fruit = prompt("enter your fruit")
```

```
switch (fruit) {  
  case "mango":  
    console.log("100 rs")  
    break;  
  case "apple":  
    console.log("200 rs")  
    break;  
  case "banana":  
    console.log("300 rs")  
    break;  
  default:  
    console.log("item not present")  
    break;  
}
```

Task - : take a number using a prompt and check if the number is multiple of 5 or not.

Task -: write a program that grade student marks.

90 - 100 A

70 - 89 B

60 - 69 C

50 - 59 D

0 - 49 F

Task -: write a program to check leap year.

```
if((year % 4 === 0 && year % 100 != 0) || (year % 400 === 0)) {
    console.log("leap year")
}
else{
    console.log("not a leap year")
}
```

Task -: write a program that take number from **1 to 7** as input and prints the corresponding date of the week using the **switch** case statements

Task -: create a simple calculator using switch case that perform 4 basic operation (+,-,\*,/) and you have to take user input form number and operation both.

### **String methods.**

**Methods** - actions that can be performed on an object.

Synxta - : stringName.method();

## **Trim methods.**

```
let str = " hello "
console.log(str) //" hello "
let str2 = str.trim()
// console.log(str2) "hello"
```

## **toUpperCase and toLowerCase.**

```
// ----- P----- M----- P----- C----- S----- R----- E----- S----- T----- O----- U----- P----- L----- E-----
```

```
let str = "APPLE"
console.log(str.toLowerCase()) //apple
let str2 = "mango"
console.log(str2.toUpperCase()) //MANGO
```

**Note :-** string are immutable we can not make any direct changes in our string whatever methods we use it make a copy and then have operation on the and return the copy one.

## **String methods with an arguments.**

**Arguments-** : it is a some value we pass to the methods.

**Format :-** stringName.method(**args, args,.....**)

## **indexOf**

Return the first index of occurrence of some value in string. Or gives -1 if the value is not there.

```
let str = "this is javascript revision class"

console.log(str.indexOf('r')) // 14

console.log(str.indexOf('java')) //8
console.log(str.indexOf('w')) // -1
```

## **Method chaining.**

Using one methods after another. Order of execution if going to be left to right.

```
let str = "      this is javascript revision class      "
console.log(str)
let newStr = str.trim().toUpperCase()
console.log(newStr) //THIS IS JAVASCRIPT REVISION CLASS
```

## **Slice**

Return a part of the original string as a new string

```
let str = "hello this is js"
let str2 = str.slice(0,5)
console.log(str2)
console.log(str.slice(6)) // if only one value given then end value is
goint to be stringlength.
console.log(str.slice(-6)) // -ve value means string length - value
```

## **Replace**

Search for a value in the string and return a new string with the value replaced.

```
let str = "hello jhon"
```

```
str.replace("jhon", "shen") // hello shen
let str = "hello jhon"
let newStr = str.replace('jhon', 'shen')

console.log(newStr) //hello shen
```

## **repeat .**

Return a string with the numbers of copies of a string.

```
Let str = "Mango"  
str.repeat(3); //MangoMangoMango  
let str = "Mango"  
let newStr = str.repeat(3)  
console.log(newStr) //MangoMangoMango
```

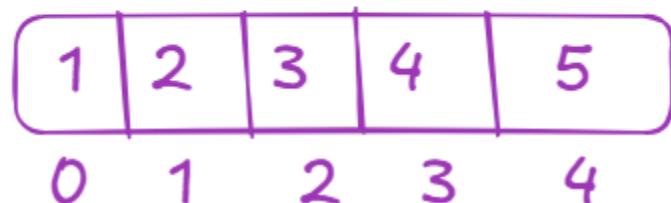
Task :- extract word "script" from string "javaScript"

## Arrays

Linear collection of things.

```
let arr = [1,2,3,4,5]
```

**let arr = [1,2,3,4,5]**



```
let num = [1,2,3,4]  
let ch = ["a", "b", "apple", "manog"]  
let mix = [1,2,3, "ramehs", "suresh", true, [12,13]]  
let empty = []
```

```

> num
< ▶ (4) [1, 2, 3, 4]

> ch
< ▶ (4) ['a', 'b', 'apple', 'manog']

> mix
< ▶ (7) [1, 2, 3, 'ramehs', 'suresh', true, Array(2)]

> empty
< ▶ []

```

## Arrays are mutable.

```

> num
< ▶ (4) [1, 2, 3, 4]

> num[10] = 10
< 10

> num
< ▶ (11) [1, 2, 3, 4, empty × 6, 10]

> num[0] = 100
< 100

> num
< ▶ (11) [100, 2, 3, 4, empty × 6, 10]

> num[2] = "two"
< 'two'

> num
< ▶ (11) [100, 2, 'two', 4, empty × 6, 10]

> num[2] = "three"
< 'three'

> num[] = "two"
✖ Uncaught SyntaxError: Unexpected token ']'

> num[1] = "two"
< 'two'

> num
< ▶ (11) [100, 'two', 'three', 4, empty × 6, 10]

```

## Array methods

**Push** → help you add in end

**Pop** → help to remove from end and return it

**Unshift** → help you add in start

**Shift** → help you to remove from start and return it.

```
> num
< ▶ (4) [1, 2, 3, 4]          > num
> num.push(5)                  < ▶ (4) [1, 2, 3, 4]
< 5                            > num.unshift(0)
> num                          < 5
< ▶ (5) [1, 2, 3, 4, 5]       > num
> num.pop()                   < ▶ (5) [0, 1, 2, 3, 4]
< 5                            > num.shift()
> num                          < 0
< ▶ (4) [1, 2, 3, 4]          > num
                                         < ▶ (4) [1, 2, 3, 4]
```

## indexOf()

Return the index of something.

```
> num
< ▶ (4) [1, 2, 3, 4]
> num.indexOf(44)
< -1
> num.indexOf(4)
< 3
```

## includes

Search for a value

```
> num
< ▶ (4) [1, 2, 3, 4]
> num.includes(3)
< true
> num.includes(33)
< false
> num.includes(-3)
< false
> num.includes('3')
< false
```

## concat

Merge two arrays.

**Syntax - : primaryarray.concat(secondarray)**

```
> num
< ▶ (4) [1, 2, 3, 4]
> num1
< ▶ (4) ['a', 'b', 'c', 'd']
> num.concat(num1)
< ▶ (8) [1, 2, 3, 4, 'a', 'b', 'c', 'd']
> num1.concat(num)
< ▶ (8) ['a', 'b', 'c', 'd', 1, 2, 3, 4]
```

## Reverse

Reverse an array.

```
> num1
< ▶ (4) ['a', 'b', 'c', 'd']
> num
< ▶ (4) [1, 2, 3, 4]
> num.reverse()
< ▶ (4) [4, 3, 2, 1]
> num1.reverse()
< ▶ (4) ['d', 'c', 'b', 'a']
```

## Slice

Copies of portion of array.

```
> num
< ▶ (4) [1, 2, 3, 4]
> num.slice(0,2)
< ▶ (2) [1, 2]
> num.slice(3)
< ▶ [4]
> num.slice(1)
< ▶ (3) [2, 3, 4]
> num.slice(-2)
< ▶ (2) [3, 4]
```

## Splice (special methods)

It is special methods it will help you to add remove replace and delete any element in array at any index.

Syntax :- **splice: start, delete count, item0, item1, ....., itemN**

If we pass only one value in the splice then it will work as splice.

```
> num
< ▶ (4) [1, 2, 3, 4]
> num.splice(1)
< ▶ (3) [2, 3, 4]
```

NOTE :- the main difference is that slice make copy array and do operation on that but splice make changes in the original array.

```
> num
< ▶ (4) [1, 2, 3, 4]
> num.splice(1,1)
< ▶ [2]
> num
< ▶ (3) [1, 3, 4]
> num.splice(1,0,2)
< ▶ []
> num
< ▶ (4) [1, 2, 3, 4]
> num.splice(2,0,2.3,2.4,2.5)
< ▶ []
> num
< ▶ (7) [1, 2, 2.3, 2.4, 2.5, 3, 4]
```

## Nested arrays

Arrays inside array

```
let na = [[1,2],[3,4],[5,6]]  
> na  
< ▶ (3) [Array(2), Array(2), Array(2)] i  
  ▶ 0: (2) [1, 2]  
  ▶ 1: (2) [3, 4]  
  ▶ 2: (2) [5, 6]  
  length: 3  
  ▶ [[Prototype]]: Array(0)  
  
> na[2][1]  
< 6  
  
> na[2][1] = 33  
< 33  
  
> na  
< ▶ (3) [Array(2), Array(2), Array(2)]
```

---

```
▼ (3) [Array(2), Array(2), Array(2)] i  
  ▶ 0: (2) [1, 2]  
  ▶ 1: (2) [3, 4]  
  ▶ 2: (2) [5, 6]  
  length: 3  
  ▶ [[Prototype]]: Array(0)
```

| Loops in js.

## For loop

```
for(initialisation; condition; updation){  
    //task  
}  
  
for(let i = 1; i<=5;i++){  
    console.log("hello everyone",i)  
}
```

```
hello everyone 1  
hello everyone 2  
hello everyone 3  
hello everyone 4  
hello everyone 5
```

## While loop.

```
while(conditon){  
    // task  
}
```

```
let i =1;  
while(i<=5){  
    console.log("hello everyone", i)  
    i++;  
}
```

```
hello everyone 1
hello everyone 2
hello everyone 3
hello everyone 4
hello everyone 5
```

## Nested for loop

Loops inside loops

```
for(let i = 0;i<=2;i++) {
    console.log("outer loop i = ", i)
    for(let j = 0;j<=1;j++) {
        console.log("inner loop j = ", j)
    }
}
outer loop i = 0
inner loop j = 0
inner loop j = 1
outer loop i = 1
inner loop j = 0
inner loop j = 1
outer loop i = 2
inner loop j = 0
inner loop j = 1
```

## Infinite loops.

```
for(let i = 0; i>=5; i++) {
```

```
}

for(let i = 1;i<=5; i--) {

}

for(let i = 0;;i++) {
```

## For of loop

```
for (elem of elements){
    //task
}

let arr = ["mango", "apple", "banana"]

for(fruit of arr){
    console.log(fruit)
}
mango
apple
banana
```

