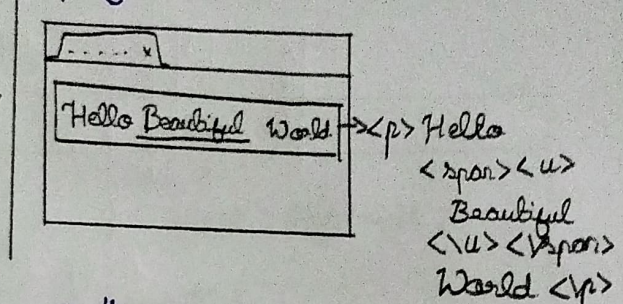
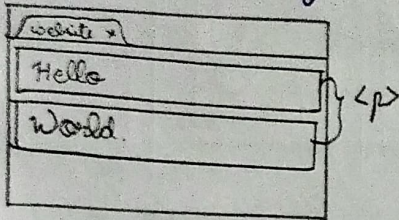


8.0. CSS Display :-

→ Block, inline and inline-block

→ Website layout.

→ Normally if we use `<p>` element or any, it will take full width of the display.



→ `<p> Hello <span><u> Beautiful <\u><\span> World <\p>`

→ The uniqueness of the `span` element is that it has unique display property.

→ `<h2> Harry <\h2>` | `h2 {`  
`<h2> Potter <\h2>` | `display: inline;`  
`}`

→ Here the 2 headings are displayed in the same line with the keyword ⇒ display: inline

output:

Harry Potter

→ if display: block for same one, then the output be:-

output:

Harry

Potter

⊗ default one.

→ if display: inline-block, here block element allows to set Height and width whereas inline element ~~can~~ can go in the same line.

output:

Block

inline

→ if display: none, there is no output.



### 8.1. CSS float:-

- Wrapping text using float and clear.
- used for website layout.
- It wraps text around the element.
- Newspaper is an best example.

How to use float properly?

index.html	style.css
<pre>&lt;img ... /&gt; &lt;p&gt; text... &lt;/p&gt; &lt;footer&gt; ..... &lt;/footer&gt;</pre>	<pre>img {   float : left/right ; } footer {   clear : left;</pre>

- float property is set on to the image and not on the text [Paragraph].
- clear property clears the any responsibility to wrap around things that are floating.



## 8.2- Responsive Websites:-

→ Making Websites looks good on all screen sizes.

4 main types:-

- ① Media Queries
- ② CSS grid
- ③ CSS flexbox
- ④ External frameworks, Eg:- Bootstrap.

### ① Media Query:-

```
@media (max-width: 600px) {
```

```
/* CSS for screens below or equal to 600px wide */  
}
```

→ instead of using selectors in CSS file, use **@media** to use media query and inside that give some condition **(max-width: 600px)**. it means that if the screen size is  $\leq 600px$ , then apply this CSS style inside the content.

### ② CSS Grid:-

→ Hard to understand but flexible.

index.html	style.css
<pre>&lt;div class="grid-container"&gt;   &lt;div class="first card"&gt;&lt;/div&gt;   &lt;div class="card"&gt;&lt;/div&gt;   &lt;div class="card"&gt;&lt;/div&gt;   &lt;div class="card"&gt;&lt;/div&gt;   &lt;div class="card"&gt;&lt;/div&gt; &lt;/div&gt;</pre>	<pre>.grid-container {   display: grid;   grid-template-columns: 1fr 1fr; <span style="margin-left: 20px;">2 columns</span>   grid-template-rows: 100px 200px 200px;   gap: 30px; }   .first {     grid-column: span 2;   }   .card {     background-color: blue;   }</pre>

→ fr  $\Rightarrow$  fraction [ratio]

→ span 2  $\Rightarrow$  combining 2 column of a row.



### ③ CSS flex box :-

- > grid used for 2D [↑ ↓]
- > flex used for 1D [↑ / →]

index.html	style.css
<pre> &lt;div class="flex-container"&gt;   &lt;div class="first-card"&gt;&lt;/div&gt;   &lt;div class="second-card"&gt;&lt;/div&gt;   &lt;div class="third"&gt;&lt;/div&gt;   &lt;div class="card"&gt;&lt;/div&gt; &lt;/div&gt; </pre>	<pre> .flex-container {   display: flex; } .card {   background: blue;   border: 30px solid white;   height: 100px;   flex: 1; } .first {   flex: 2; } .second {   flex: 0.5; } </pre>
<p>→ .card { ... flex: 1; }</p> <p>⇒ make all cards to divide in same ratio. [1:1:1:1]</p> <p>→ .first { flex: 2; }</p> <p>⇒ adjusted to 2x of normal card.</p> <p>→ .second { flex: 0.5; }</p> <p>⇒ 0.5x of normal card.</p>	

### ④ Bootstrap Framework :-

- > External one.
- > Imagine and dividing your screen as 12 divisions

```

<div class="container">
  <div class="row">
    <div class="card col-6"> 6 of 12 is modified
      | card
    </div>
    <div class="card col-2"> 2 of 12
      | card
    </div>
    <div class="card col-4"> 4 of 12
      | card
    </div>
  </div>
</div>

```



### 8.3. Media Query :-

- Adding breakpoints to define responsive layout.
- Simple to understand.

Example:-

```
@media (max-width: 600px) {  
    h1 {  
        font-size: 15px;  
    }  
}
```

- it means that "if the screen size is less than or equals to 600px, then the font size of  $h_1 = 15$ px else default".
- if we use minwidth then it is greater than or equal to 600px.
- we can able to combine min & max width using and logic. so that we can range the style.
  - ⇒ @media (max-width: 1000px) and (min-width: 600px) {  
 -----;  
 -----;  
}
- @media print(orientation: landscape) {....}  
style is applied if the screen is landscape.