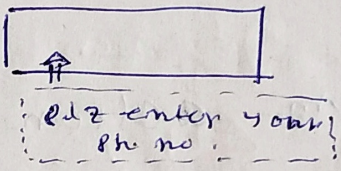
## Tooltips:-

```
<label>
    Phone no:
<input type="text" size="10"
name="Phone" title="Plz enter your
Phn no" />
</label>
```

o/p →

Phone no 


## CSS (cascading style sheet)

### Selectors:→

Simplest rules that can be applied to all occurences of a particular tag.

For eg,

To set the line spacing for all paragraph.

```
p { line-height : 150%; }
```

Set the Background colour of the entire doc,

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

Setting rules for multiple tag (grouping them with commas)

```
h1, h2, h3 { background: yellow, color: red; }
```



For making each heading have a different size

```
h1 { font-size : 200%; }
h2 {    "      : 150%; }
h3 {    "      : 100%; }
```

ID ~~Rules~~ rules :→

How can a particular style be applied to one particular tag without using it inline styles?

```
<h1 id="Heading1">welcome to JPL</h1>
```

we know that

```
<a href="#Heading1">GO TO TOP</a>
```

Through CSS

#Heading1 { background-color: pink; }

=

```
<html>
<head>
<title> ID Rules </title>
<style type="text/css">
#paragraph2 { background-color: gray; }
</style>
</head>
<body>
<p> First paragraph </p>
<p id="paragraph2"> This paragraph has style 1 </p>
</body>
</html>
```

## CLASS RULES :-

Used to define the names of the classes to which a particular tag belongs. Unlike Id values, class values don't have to be unique as many elements can be member of the same class.

We specify the class name with a dot before it as a selector.

```
<head>
  <title> classes ex </title>
  <style type = "text/css">
    .veryimp { background-color: skyblue }
  </style>
</head>
<body>
  <h1 class = "veryimp"> schedule </h1>
  <p class = "veryimp"> 2nd Sem exam's of
                        </p>
  <p> commence on 17th may' 08 </p>
  <p class = "veryimp"> All the best </p>
</body>
```

Inheriting

```
<head>
  <title> multiple classes ex. </title>
  <style type="text/css">
    .heading { font-family: Impact, sans-serif; }
    .veryimp { background-color: skyblue; }
    .stuff { color: red; }
    .veryimp.stuff { font-style: Italic; }
    .veryimp.examples.stuff { text-decoration: underline; }
  </style>
</head>
<body>
  <h1 class="veryimp heading stuff">
    Heading </h1>
  <p class="veryimp"> 1st para </p>
  <p class="stuff"> 2nd para </p>
```

 3rd para

</body> 

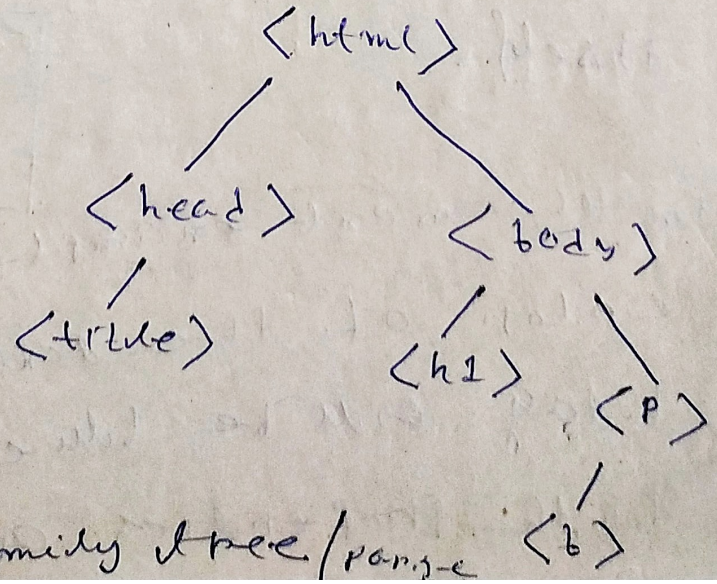
Heading -> Impact, grey blue, red, italics.

3rd para -> grey blue, red, italics.

Document structure & Inheritance :->



```
<html>
  <head>
    <title> Test file </title>
  </head>
  <body>
    <h1> Test </h1>
    <p> this is a <b> Test </b> </p>
  </body>
</html>
```




family tree / page

```
p { color: red;}
<b> this is bold & red </b>
</p>
```

---

```
p { border: solid 6;}
<b> this is within another border
</b>
</p>
```



Here the b tag inherits the color from it's parent.

Here b tag doesn't inherit from it's parent. Border is limited just to the paragraph itself.

```
p { color : red;
    font-size : 14pt;}
b { color : blue;}
```

In the next example the color of the text within the b tag will be blue & 14pts. both properties are inherited

but the color is overcomitten. the combination of multiple rules with elements inherit-ting some properties & overarriving other is the idea of the cascade that css is made for.

css generally follows the following rules →

(i) The closure to the tag the rule is, the more powerful it is.

(ii) the most specific the rule is, the more power-ful it is.

Schendo classes :-

They deal with style info. for contains which may change depending on user activ-ity.

```
<head>
  <style type = "text/css">
    a.link { color: blue; }
    a.active { color: red, background color:
                                    skyblue; }
    a.visited { color: purple; }
    a.hover { color: yellow; text-decoration:
              underline; font-size: larger; }
    a.focus { border.style: dashed;
              border.width: 1px;
              background.color: pink; }
  </style>
</head>
<body>
  <a href = "http://www.yahoo.com">
      yahoo </a>
</body>
```

CSS

pseudo

~~selected~~ element:-

These are used along with common block level text elements such as <P> to affect the presentation of the 1st letter or the 1st line of the enclosed text.

```
<Style type = "text/css" >
  P: first-line { background-color:
                  Skyblue; }
  P: first-letter { color: red;
                    font-size: 200%; }
</Style>
<pre>
  <P> This is very funny, .. </P>
</Pre>
```



Contextual Selections:-

To specify that all strong tag that occures within a P tag gets treated in a some what diff. way as compared to the same element occuring elsewhere

within a doc. To create such a rule we use contextual selection.

p strong { background-color : orange; }

eg:-

 This  is not  directly  within  the Paragraph .

p > strong { background-color : pink ; }

In CSS2 for the above eg.



! Important override :-

If a particular rule should never be over written by another rule, the ! Important should be needed just before ; of the rule.

```
<head>
  <title> Important override </title>
  <style type="text/css">
    p { color: red !important; font-size: 12Pt; }
  </style>
</head>

<body>
  <p> 1st para </p>
  <p style="color: green; font-size: 24Pt;">
    Inline style for green text
    override </p>
  <p style="color: blue !important,
            font-size: 20Pt; ">
    IS this a blue text? </p>
</body>
```