

1 What is css? what is the purpose of css?

2 CSS stands for cascading style sheets. css is used for styling the elements of a web page, we can change the color, size and other aspects in rendering the elements of a web page using css.

3

4 Why do we need to use css, why not html provide tags for doing the same?

5 initially html has provided formatting tags like font, color, background-color etc. having these tags written at all the places of a web site in web pages is going to be complex and takes lot of time for developing the website.

6

7 For eg.. we wanted all the headings of our pages to be look alike "font: helvetica, font-size: 20px, color: red" if the same has to be applied for all the pages each heading should be surrounded with formatting html tags that takes more time in developing web pages by the web developers.

8

9 To solve this problem w3c has introduced style sheets. cascading style sheets, once written will be cascade to all the elements based on matching / hierarchy. So by use of css we can reduce greatly the amount of code we have to write for applying styles to the web pages.

10

11 How to write cascading style sheet?

12 One cascading style is called style rule. rule looks as below

13 selector {declaration; declaration}

14           prop: value; prop: value

15

16 selector = the elements on whom the style should be applied

17 all the declarations should be written in {} curly brackets

18 declaration = indicates a property and its value we want to apply on a element. The properties varies depends on the type of the element on whom we are applying.

19 prop = name of style property we want to modify

20 value = is a pre-defined value or a range that is allowed for that property

21

22 example:-

23 <style>

24 p {

25     font-family: fantasy;

26     font-size: 23px;

27 }

28 </style>

29 -----

-----

```
30
31  css selectors
32  -----
33  Using css selector we can tell the style rule has to be
    applied to which elements on our web page. We can
    classify a style selector into below types
34  1. element selector
35  2. id selector
36  3. class selector
37  4. block selector
38  5. psuedo-selector
39
40  #1 element selector
41  We target the style to a element of a type in web page by
    writing tag name of the element as shown below.
42
43  <html>
44      <head>
45          <style>
46              p {
47                  font-family: helvetica;
48                  font-size: 14px;
49              }
50          </style>
51      </head>
52      <body>
53          <p>Paragraph 1</p>
54          <p>Paragraph 2</p>
55      </body>
56  </html>
57
58  #2 id selector
59  It is used for applying a style to a specific element of
    a web page rather than all the elements.
60  For every html element we can give an id attribute that
    helps in uniquely identifying and accessing the element.
61
62  <html>
63      <head>
64          <style>
65              #para1 {
66                  font-family: helvetica;
67                  font-size: 14px;
68              }
69          </style>
70      </head>
71      <body>
72          <p id="para1">Paragraph 1</p>
73          <p>Paragraph 2</p>
74      </body>
```

```
75 </html>
76
77 #3 class selector
78 we bind html elements to a class name and we apply styles
  to all the elements bearing the class. In this approach we
  can select subset of elements on page to whom we want to
  apply the styles
79
80 <html>
81   <head>
82     <style>
83       .mainContent {
84         font-family: helvetica;
85         font-size: 14px;
86       }
87     </style>
88   </head>
89   <body>
90     <p class="mainContent">Paragraph 1</p>
91     <p>Paragraph 2</p>
92     <div class="mainContent">
93       This a section of content
94     </div>
95   </body>
96 </html>
97
98
99 #4 block/group selector
100 to all the group of elements we want to apply a style
101
102 <html>
103   <head>
104     <style>
105       p, a, div {
106         font-family: helvetica;
107         font-size: 14px;
108       }
109
110     </style>
111   </head>
112   <body>
113     <p>Paragraph 1</p>
114     <p>Paragraph 2</p>
115     <ul>
116       <li></li>
117     </ul>
118     <a href="">click me</a>
119     <div>
120       This is the content
121     </div>
```

```
122     </body>
123 </html>
124
125
126 $5 global selector
127 To every element on my page apply these styles
128
129 <html>
130     <head>
131         <style>
132             * {
133                 font-family: helvetica;
134                 font-size: 14px;
135             }
136
137         </style>
138     </head>
139     <body>
140         <p>Paragraph 1</p>
141         <p>Paragraph 2</p>
142         <ul>
143             <li></li>
144         </ul>
145         <a href="">click me</a>
146         <div>
147             This is the content
148         </div>
149     </body>
150 </html>
151
152 #6 psuedo selector
153 based on condition/criteria we can select the elements
154 for whom the style should be applied
155
156 <html>
157     <head>
158         <style>
159             p:nth-child(odd) {
160                 font-family: helvetica;
161                 font-size: 14px;
162             }
163
164         </style>
165     </head>
166     <body>
167         <p>Paragraph 1</p>
168         <p>Paragraph 2</p>
169         <ul>
170             <li></li>
171         </ul>
172         <a href="">click me</a>
```

```
171     <div>
172         This is the content
173     </div>
174 </body>
175 </html>
176 -----
177 -----
178 How many ways we can write css in web pages?
179 There are three ways in writing the css for web pages.
180 1. external stylesheet = we write all the style rules in
181    a external file and import into a web page
182 2. internal stylesheet = within the webpage we write
183    style rules
184 3. inline styles = embedded within the element
185
186 1. external stylesheet
187 -----
188 all the css rules for your website will be written in an
189 external file with extension ".css". The file in which we
190 write style rules is called "Stylesheet" | "css" file. We
191 import the css file into the webpage to apply styles.
192 We can easily change the look and feel of the controls of
193 the website by simply changing the external css file. It
194 is recommended to use external stylesheet for a web page.
195 It helps us in reusing the style rules.
196
197 project
198 |-css
199   |-default.css
200   |-dark.css
201   |-our-services.html
202
203 css/default.css
204 -----
205 body {
206     font-family: fantasy;
207     font-size: 14px;
208 }
209 .important {
210     color: red;
211 }
212
213 css/dark.css
214 body {
215     font-family: Tahoma;
216     font-size: 15px;
217 }
218 .important {
219     color: red;
220     font-style: italic
```

```

212     }
213
214
215 our-services.html
216 -----
217 <html>
218     <head>
219         <title>Our Services</title>
220         <link rel="css/stylessheet" href="css/dark.css">
221     </head>
222     <body>
223         <p class="important">
224         </p>
225     </body>
226 </html>
227
228
229 #2 Internal Stylesheet
230 If I have stylerules that has to applied to a specific
web page only. in such a case define the style local to
webpage using style tags
231
232
233 our-services.html
234 -----
235 <html>
236     <head>
237         <title>Our Services</title>
238         <style>
239             .important {
240                 font-family: monospace;
241             }
242         </style>
243     </head>
244     <body>
245         <p class="important">
246         </p>
247     </body>
248 </html>
249
250 #3 inline styles
251 Inline styles are written directly at the element level
to which want to apply the style. it is highly
dis-couraged to use inline styles. To write an inline
style at the element level we need use an attribute for
every element called style="declaration"
252
253 our-services.html
254 -----
255 <html>

```

```

256     <head>
257         <title>Our Services</title>
258     </head>
259     <body>
260         <p style="font-family: Tahoma; font-size: 18px;">
261         </p>
262     </body>
263 </html>
264 -----
265 color css style = in web pages
266 What are the different ways we can use colors in a web
267 page using css?
268 css background color
269 -----
270 <style>
271     .subheading {
272         background-color: cadetblue;
273     }
274 </style>
275 <p class="subheading">We are using background color for
276 this paragraph</p>
277 css text color
278 -----
279 used for changing the color of text display
280 <style>
281     p {
282         color: red;
283     }
284 </style>
285 <p>
286     Text in color
287 </p>
288
289 css border color
290 -----
291 border color will display the line around the element in
292 color specified
293 note:- border color is applied only when set the border,
294 border-style properties.
295
296 <style>
297     p {
298         border: 1px;
299         border-style: solid;
300         border-color: aqua;
301     }
302 </style>

```

```
301
302 <p>
303     Paragraph with colored border
304 </p>
305
306 we can apply border color to all sides also
307 border-left-color, border-right-color, border-top-color,
308 border-bottom-color
309 the shorthend for writing it is border-color: leftcolor
310 rightcolor topcolor bottomcolor;
311
312 css color choosing
313 -----
314 We can directly specify the color names.
315 there are other ways we can customize the color we want
316 to render using one of three ways
317
318 rgb();
319 HEX
320 HSL
321
322 <p style="color: rgb(255, 23, 200);"></p> = r=red,
323 b=blue, g=green using you can create different colors
324 <p style="color: rgba(255, 23, 200, 0.5);"></p>
325
326 HEX = Hexa Decimal Numbers
327 <p style="color: #ad9333;"></p>
328
329 HSL = hue, saturation, lightness
330 <p style="color: hsl(0, 100%, 100%)"></p>
331
332 -----
333 -----
334
335 css background
336 There are several css properties available regarding
337 managing the background for elements
338
339 1. background-color
340 2. background-image
341 3. background-repeat
342 4. background-attached
343
344 background-color = is to change the background color of
345 an element.
346
347 <div style="background-color: rgb(10,03,40); opacity: 0.2">
348 </div>
349
350 background-image = used for displaying an image as a
351 background for container if the image size is less than
352 the container automatically the image will be repeated
353 horizontal and vertical.
```



```
341
342 <div style="background-image: url(path)">
343 </div>
344
345 background-repeat = using this we can control the image
346 repetitions
347 note:- this property is applicable when we have
348 background-image specified
349
350 <div style="background-image: url(CHERRY PICK.png);
351 background-repeat: no-repeat"></div>
352
353 background-position:- position is used for placing the
354 background image within the container
355 note:- if we use background-image with background-repeat
356 : no-repeat then only background-position property is
357 applied.
358 for eg.. top right | top left | bottom right | bottom left
359 -----
360 -----
361
362 css borders
363 border is a line we want to draw around an element.
364 1. border
365 2. border-color
366 3. border-style
367    - solid
368    - dotted
369    - dashed
370 4. border-radius = is used for drawing rounded border
371 around an element
372 -----
373 -----
374
375 css margins
376 css margin is used to create space around the element
377 outside the border of the element
378
379 <style>
380   p {
381     border: 1px;
382     border-style: solid;
383     border-color: black;
384     margin: 20px, 20px, 20px, 20px;
385     margin-top: 20px;
386     margin-bottom: 20px;
387     margin-left: 20px;
388     margin-right: 20px;
389   }
390 </style>
391
392 <p>Top Tag</p>
```

```
381 <p>Bottom Tag</p>
382
383 -----
384
384 css padding
385 css padding creates a space around the contents of the
    element
386
387 <style>
388     .subsection {
389         border: 1px;
390         border-style: solid;
391         border-color: black;
392         padding-top: 10px;
393         padding-bottom: 10px;
394         padding-left: 10px;
395         padding-right: 10px;
396         padding: 10px, 10px, 10px, 10px;
397     }
398 </style>
399
400 <p class="subsection">Sed ut perspiciatis unde omnis iste
    natus error sit voluptatem accusantium doloremque
    laudantium, totam rem aperiam, eaque ipsa quae ab illo
    inventore veritatis et quasi architecto beatae vitae
    dicta sunt explicabo.</p>
401 -----
402
402 css height/width
403 css height and width is used for changing the heigh and
    with of an element in rendering.
404
405 for every element we can specific heigh and width
    property to render on the web page.
406 1. by default height and width of an element is auto
    (based on contents it will be rendered)
407 2. we can specify height and width in either cm, px or %
408
409 <style>
410     div {
411         height: 400px;
412         width: 60%
413     }
414 </style>
415
416 <div>
417     Sed ut perspiciatis unde omnis iste natus error sit
        voluptatem accusantium doloremque laudantium, totam rem
        aperiam, eaque ipsa quae ab illo inventore veritatis et
        quasi architecto beatae vitae dicta sunt explicabo.
```

```
418 </div>
419
420 in addition to height and width the css has added 2 more
properties
421 1. max-width
422 2. max-height
423 both these helps you in adjusting the element sizes based
on screen resolution.
424 -----
425
425 css outline
426 its an line that is drawn outside the border of the
element to make a element highlighted.
427 Below are the outline properties with which we can drawn
the outline of an element
428 1. outline-style:
429 2. outline-color:
430 3. outline-width:
431 4. outline
432
433 outline-style is mandatory to have other properties work
434
435 <style>
436     div {
437         border: 1px;
438         border-style: solid;
439         border-color: black;
440         outline-style: solid;
441         outline-width: 10px;
442         outline-color: red;
443     }
444 </style>
445
446 <div>
447     Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam
rem aperiam, eaque ipsa quae ab illo inventore
veritatis et quasi architecto beatae vitae dicta sunt
explicabo.
448 </div>
449 -----
450
450 css text
451 There are various css formatting options are available
for text content we render in a element
452
453 1. text color
454 2. background-color
455 3. text alignment
456 4. text transformation
```

```

457 5. text spacing
458 6. text shadow
459
460 <style>
461     body {
462         color: aquamarine; /** applies to only text we render
463         on the page */
464         text-align: justify;
465         direction: rtl;
466         vertical-align: top;
467         text-decoration: line-through;
468         text-transform: capitalize;
469         letter-spacing: 5px;
470         line-height: 5px;
471         word-spacing: 10px;
472         white-space: nowrap;
473         text-indent: 50px;
474         text-shadow: 2px 2px;
475     }
476     p {
477         color: red;
478     }
479 </style>
480 <body>
481     How are you
482     <p>
483         Good Morning How are
484         asdfafasfdafd
485     </p>
486     <p>
487         G o o d           M o r n i n g
488
489         H a v e           a           g r e a t           d a y
490     </p>
491 </body>
492 -----
493 -----
494
495 css fonts
496 using the font-family and font-size we can change the
497 font of a web page in css.
498 all the browsers supports predefined set of fonts which
499 are called "Font Web Safe"
500
501 "Font Web Safe" supported by all the browsers
502     Arial (sans-serif)
503     Verdana (sans-serif)
504     Helvetica (sans-serif)
505     Tahoma (sans-serif)
506     Trebuchet MS (sans-serif)

```

503 Times New Roman (serif)  
504 Georgia (serif)  
505 Garamond (serif)  
506 Courier New (monospace)  
507 Brush Script MT (cursive)  
508

509 but there is always a chance where due to different  
browsers and version few web safe fonts may not be  
supported by browser. due to this problem always use  
fallback fonts for your primary font as below.

510  
511 font-family: Helvetica; sans-serif  
512 The fallback fonts will be applied by the browser when  
the primary font is not available. and it is recommended  
to use fallback fonts as more generic fonts

513  
514 Always one font is the king. If at all required use  
different fonts at hierarchy level

515 Heading H1- Use one font

516 h2 = another font

517 h3 = another font

518 subsection

519 important

520 paragraph

521 default font body level

522 -----  
-----

523 css icons

524  
525 <head>  
526 <script src="https://kit.fontawesome.com/6ea0d26f8e.js"  
crossorigin="anonymous"></script>

527 </head>

528 <body>

529 <i class="fas fa-database">Database</i>

530 </body>

531  
532 -----  
-----

533 css links

534 we have various different styles we can apply for a  
anchor tag in a web page. by default anchor tag is  
displayed with underline.

535

536 default = blue

537 active=red

538 visited=purple

539  
540 all of the above default behaviours can be modified using  
css links

```

541
542 <head>
543   <style>
544     a {
545       text-decoration: none;
546       color: brown;
547     }
548     a:active {
549       color: chartreuse;
550     }
551     a:visited {
552       color: aqua;
553     }
554     a:hover {
555       color: aquamarine;
556     }
557   </style>
558 </head>
559 <body>
560   <a href="#">Home</a>
561 </body>
562 -----
563
564 css lists
565
566 Using css we can change the look and feel of a html list
567 like
568 we can change list icon, we can replace a list with
569 image, we remove bullets and we change the position of
570 the icons as well
571
572 <head>
573   <style>
574     ul.li {
575       list-style-type: circle; = we can change icon
576       list-style-image: url(''); = we can replace with
577       image
578       list-style-position: inside; = we can shift the
579       position li in the list
580       list-style-type: none; = we can hide the icon
581     }
582   </style>
583 </head>
584 <body>
585   <ul>
586     <li>Banana</li>
587     <li>Apple</li>
588     <li>Mango</li>
589     <li>Oranges</li>
590   </ul>

```

```
585 </body>
586 -----
587 CSS Table
588 -----
589 For a table we can apply
590     1. borders
591     2. size
592     3. style
593
594 <head>
595     <style>
596         table td, th {
597             border: 1px solid black;
598             border-collapse: collapse;
599         }
600     </style>
601 </head>
602 <body>
603     <table>
604         <tr>
605             <th>Job Title</th>
606             <th>Description</th>
607             <th>Organization</th>
608             <th>Experience</th>
609             <th>Location</th>
610         </tr>
611         <tr>
612             <td></td>
613             <td></td>
614             <td></td>
615             <td></td>
616             <td></td>
617         </tr>
618     </table>
619 </body>
620 #1
621 we apply border to a table at different level
622 table {
623     border: 1px solid black;
624 }
625 this will draw a border around the table but will not
    draw border for rows and columns
626
627 #2 we can apply border to only rows
628 table tr {
629     border: 1px solid black;
630 }
631 this will leaves table with no borders but tr drawn with
    borders
```

```
632 Note:- when we apply borders at td/th level we dont need
633 to write borders for tr
634 #3we can apply border for all the cells in the table
635 table, td, th {
636     border: 1px solid black;
637 }
638 Now for table, td, th borders are applied and appears as
639 double border around each cell, so we can collapse double
640 border by writing border-collapse: collapse
641
642 2. heigh and width of a table
643 <style>
644     table {
645         height: 100px;
646         width: 100%;
647         margin: auto;
648     }
649 </style>
650
651 3. text align of contents in table cells
652 we can align contents of the table cell by using
653 text-align property at td/th level
654
655 td {
656     text-align: center;
657 }
658
659 4. table styles
660 padding = can be given for each cell to create space
661 around the content
662 table td, th {
663     border-bottom: 1px solid black;
664 }
665
666 table tr:nth-child-element(even) {
667     background-color: grey;
668 }
669
670 table tr:hover {
671     backgroud-color: red;
672 }
673
674 th {
675     background-color: red;
676     color: white;
677 }
678 -----
679 -----
680
681 css display|visible
682
683 display: is an css property that can carry the below
```



```

values.
676 1.none
677 2.block
678 3.inline
679
680 <style>
681     #p1 {
682         display: none;
683     }
684 </style>
685 <p id="p1">Paragraph</p>
686
687 when we set display:none; property the element will be
    hidden in web page which can used to remove elements
    without removing them from page layout.
688
689 every element in html has a display attribute which is
    block-level or inline-level
690 block-level
691 - starts always in new line
692 - occupies to the max width of the page
693
694 inline-level
695 - relative page flow
696 - occupies only to the max of the content
697
698 we can change default display type of an element to
    block-inline and inline-block using display property
699
700 <style>
701     p {
702         display: inline;
703     }
704 </style>
705
706 <p>p1</p>
707 <p>p2</p>
708 <p>p3</p>
709
710 <style>
711     i {
712         display: block;
713     }
714 </style>
715 <i>i1</i><i>i2</i><i>i3</i>
716
717 Other css property is visibility which is used for
    controlling the display of the element whether it should
    be shown on the page or not. The possible values it can
    take is

```

718 1. visible  
719 2. hidden = within the element an empty space will be  
left the page will not adjusted.

720  
721 visibility: hidden = will hides the content but doesnt  
changes the page structure, so the when we hide the  
element any empty space will appear within its place  
722 display: none = hides and removes the space pushing next  
set of elements to occupy the space

723  
724 -----  
-----

725 css position property  
726 position property help us in positioning the html  
elements on a web page. it helps in managing the position  
method of html elements  
727 we can move the elements or position the elements of a  
web page using left, right, top, bottom properties of an  
element, but these attributes will be applied based on  
position property of the element. Based on the position  
property the behaviour of left/right/top/bottom  
attributes will change

728  
729 There are total 5 position methods are there allowed for  
position property

730 1. static  
731 2. fixed  
732 3. relative  
733 4. absolute  
734 5. sticky  
735

736 for e.g.. below is the snippet shows you how to apply  
position property for an HTML element.

737 <div style="position: static">  
738 </div>

739  
740 1. static  
741 HTML elements are positioned by default based on static  
position  
742 if an elements position is "static" then  
left|right|top|bottom will not have any affect. The  
elements are positioned according to the page flow  
relative to the page.

743  
744 <style>  
745 .static {  
746 position: static;  
747 border: 1px solid black;  
748 left: 30px;  
749 }

```
750 </style>
751
752 <div>
753     Introduction to css
754 </div>
755 <div class="static">
756     Paragraph with static position
757 </div>
758
759 3. relative
760 by default the elements with position: relative falls
    into the normal page flow. but left|top|right|bottom
    properties can be used for changing the display position
    of the element.
761 when we use position: relative if any space left towards
    the leftside of the element will be blank and cannot be
    filled with any other element.
762 The left, right, top and bottom attributes will be taken
    in to account relative to the current page flow.
763
764
765 <style>
766     div.relative {
767         position: relative;
768         left: 30px;
769         border: 1px solid black;
770     }
771 </style>
772
773 <div>
774     CSS Position Property
775 </div>
776 <div class="relative">
777     Position can be used for changing the position of page
        elements
778 </div>
779
780 4. fixed
781 fixed position elements are positioned relative to the
    viewport. these always stays in the fixed position even
    we scroll the page as well and will not leave any gap
    like relative elements
782 <style>
783     body {
784         margin-bottom: 2000px;
785     }
786     div.fixedposition {
787         position: fixed;
788         border: 1px solid black;
789         bottom: 0;
```

```
790         width: 55%;
791     }
792 </style>
793 <div class="fixedposition">
794     One day, a thief who knew the old miser's routine,
       waited for the old man to go back into his house
795 </div>
796
797 2. absolute
798 is positioned relative to the nearest parent and if there
    is no parent element then it will be placed relative to
    the viewport of the page and behaves as fixed but scrolls
    along with the page.
799
800
801 <div style="position: relative">
802     One day, a thief who knew the old miser's routine,
       waited for the old man to go back into his house
803     <div style="position: absolute; top: 30px">
804         He continued this routine every day, but not once
           did he spend the gold he saved.
805     </div>
806 </div>
807
808 5. sticky
809 sticky is a position used for sticking the page elements
    on user scroll position.
810 <style>
811     body {
812         margin-bottom: 2000px;
813     }
814     div.stickyposition {
815         position: sticky;
816         top: 0;
817         border: 1px solid black;
818         width: 100%;
819         height: 40px;
820     }
821 </style>
822
823 <body>
824     <div>
825         One day, a thief who knew the old miser's routine,
           waited for the old man to go back into his house
826     </div>
827     <div class="stickyposition">
828         He continued this routine every day, but not once did
           he spend the gold he saved.
829     </div>
830 </div>
```

```
831         is positioned relative to the nearest parent and if
            there is no parent element then it will be placed
            relative to the viewport of the page and behaves as
            fixed but scrolls along with the page.
832     </div>
833 </body>
834 -----
835 position = we can change the position of the page
            elements using css position property
836     - static = default, places the elements according to
            the page flow. left|right|top|bottom are not applied
837     - relative = places the elements according to the
            default page flow and left|right|top|bottom will
            changes the element position in the page. but any space
            created to the left of the element cannot be fixed.
838     - fixed = always fixed to the current viewport, if
            scrolled also there is no affect
839     - absolute = used for nested elements, where we want to
            place a child element relative to the parent we usually
            use nested. if not a parent element it is positioned to
            the viewport, but scrolls along with the page.
840     - sticky = element sticks to user scroll position. we
            tell at which place in the page the element should stick.
841 -----
842 css overflow
843 overflow css property is used for managing the contents
            of a block level element when the content is more than
            the size of the element.
844 note:-
845 overflow is applied only to block level elements only
846 for element to apply overflow property we should have
            fixed height
847
848 overflow property can have the below values
849 1. visible
850 2. hidden
851 3. scroll
852 4. auto
853
854 <style>
855     .para {
856         border: 1px solid black;
857         height: 200px;
858         width: 200px;
859         overflow: visible;
860     }
861 </style>
862
```

```

863 <p class="para">
864     Once upon a time, there lived a shepherd boy who was
        bored watching his flock of sheep on the hill. To amuse
        himself, he shouted, "Wolf! Wolf! The sheep are being
        chased by the wolf!" The villagers came running to help
        the boy and save the sheep. They found nothing and the
        boy just laughed looking at their angry faces
865 </p>
866
867 by default the content overflow of the element area will
        be visible.
868 hidden = will hides the content and displays only the
        content that fits to the area
869 scroll = will generates scroll bar if the contents of the
        element is more than the size of the element
870 auto = based on the contents it generates either
        veritical or horizantal scroll bar
871 overflow-x: can be used for enabling horizantal scrollbar
872 overflow-y: can be used for enabling vertical scrollbar.
873 -----
        -----
874 css float
875 to float the images around the content we use float
        property. we can float an image to the left of the text
        or right of the text using css float property. used for
        positioning and formatting.
876
877 <p>
878     
879     Once upon a time, there lived a shepherd boy who was
        bored watching his flock of sheep on the hill. To amuse
        himself, he shouted, "Wolf! Wolf! The sheep are being
        chased by the wolf!" The villagers came running to help
        the boy and save the sheep. They found nothing and the
        boy just laughed looking at their angry faces. "Don't
        cry 'wolf' when there's no wolf boy!", they said
        angrily and left. The boy just laughed at them.
880 </p>
881
882 -----
        -----
883 Horizantal Navigation bar
884
885
886 <body>
887     <ul>
888         <li>Our Services</li>
889         <li>Careers</li>
890         <li>Business Integrations</li>
891         <li>About Us</li>

```

```
892     <li>Contact Us</li>
893 </ul>
894
895 <h2>Bluedart</h2>
896 <p>
897     Paragraph
898 </p>
899 </body>
```

```
900 -----
901 -----
```

```
901 How to style a select control in html?
902 <select name="city">
903     <option value=""></option>
904 </select>
```

```
905
906 Intead of using the pre-defined select control of HTML we
907 can create our own dropdowns using css.
```

```
907
908 <div>
909
910 </div>
```

```
911
912 -----
913 -----
```

```
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
```

939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987