**Basic CSS**

1. **Module-01: CSS Introduction**
2. css is the language we use to style a web page.
3. **What is css?**
4. Css stands for cascading style sheets.
5. Css describes how html elements are to be displayed on screen,paper, or in other media.
6. Css saves a lot of work. It can control the layout of multiple web pages all at once.
7. **Why use css?**
8. Css is used to define styles for web pages, including the design, layouts and variants in display for different devices and screen sizes.
9. **Module-02: CSS Syntax**
10. a css rule consists of a selector and a declaration block.
11. The selector points to the html element you want to style.
12. The declaration block contains one or more declarations separated by semicolons.
13. Each declaration includes a css property name and a value, separated by colon.
14. **Module-03: CSS Selectors**
15. a css selector select the html elements you want to style.
16. **Kinds of css selectors**
17. simple selectors = general
18. combinator selectors = specific
19. pseudo-class selectors = specific
20. pseudo-element selectors = specific
21. attribute selectors = dynamic
22. universal selectors = general
23. grouping selector = general
24. **css simple selectors**
25. element selector: h1{}
26. id selector: #title{}
27. class selector: .title{}, h1.title{}
28. **css combinators selectors**
29. descendant selector(space) = the descendant selector matches all elements that are descendants of a specified element = div h1 = all h1
30. child selector = the child selector selects all elemens that are the children of a specified element = div>h1 = all h1 div
31. adjacent sibling selector = the adjacent sibling selector is used to select an element that is directly after another spacific element = div+h1 = h1 one of div
32. genral sibling selector = the general sibling selector selects all elemens that are next siblings of a specified element = div ~ h1 = div child all h1
33. **css pseudo class selectors**
34. :link
35. :visited
36. :active
37. :hover
38. :first-child
39. :last-child
40. :nth-child(nth), :nth-child(odd), :nth-child(even)
41. :nth-last-child(nth)
42. :focus
43. :checked
44. **css pseudo elements selectors**
45. a css pesudo-element is used to style specified parts of an element. Like, styles the first letter, or line, of an element.insert content before, or after the content of an element.
46. ::first-line
47. ::first-letter
48. ::before
49. ::after
50. ::marker
51. ::selection = color, background, cursor, outline
52. **css attribute selectors**
53. [attribute] = a[target] = attribute selector
54. [attribute= “value”] = a[target= “\_blank”] = attribute value selector
55. [attribute ~= “value”] = a[title ~= “green”] = attribute tild selector
56. [attribute |= “value”] = a[title |= “bd”] = attribute pipe selector
57. [attribute ^= “value”] = a[title ^= “bd”] = attribute carret selector
58. [attribute $= “value”] = a[title $= “bd”] = attribute dollar selector
59. [attribute \*= “value”] = a[title \*= “bd”] = attribute asterisk selector
60. **css universal selector**
61. \*{}
62. universal selector css applied all html elements
63. \* includes all htm,body, title all tags.
64. **Css grouping selector**
65. h1,h3,p{}
66. grouping selector is very usefull for repeated css in all elemnts.
67. **Module-04: CSS How To Add**
68. when a browser reads a style sheet, it will format the html document according to the information in the style sheet.
69. **Three ways to insert css**
70. external css
71. internal css
72. inline css
73. **cascading or priority order**
74. inline => external/internal => browser default
75. **Module-05: CSS Comments**
76. css comments are not displayed in the browser, but they can help document your source code.
77. Css comments appliend: /\*comment here\*/
78. **Module-06: CSS Colors**
79. colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.
80. In css, a color can be specified by using a predefined color name.
81. css/html support 140 standard color name.
82. **Css background color**
83. background-color: color\_name;
84. **css text color**
85. color: color\_name
86. **css border color**
87. border-color: color\_name
88. **color pickars** : most of the use color picker in hex code to implement in web page desing.
89. **Color values:** rgb(34,232,244) , #fffff , hsl(4,50%,77%), hsla(4,30%,30%,0,5), rgba(0, 34, 53, 0.3)
90. **rgb value:**  red green blue, 0-255
91. **hex value:** #rrggbb or #rgb(3 digits shorthand of #rrggbb)
92. **hsl value:** hue, saturation, lightness 0-357, 0-100%
93. **hsla value:** hue,saturation,lightness, 0-357, 0-100%, 0.0 – 1.00
94. **rgba value:** red, green, blue, alpha , 0-255, 0.0 1.00
95. **module-07: CSS Backgrounds**
96. the css background properties are used to add background effects for elements.
97. **Background color**
98. **Opacity**
99. opacity: value
100. rgba(23, 23, 23, 0.5)
101. hsla(9, 10%, 30%, 0.4)
102. **background image**
103. background-image: url();
104. **background repeat**
105. by default background-image property repeats an image both horizontally and vertically.
106. background-repeat: repeat,no-repeat,repeat-x,repeat-y
107. **background position**
108. the background-position property sets the starting position of a background image.
109. background-position: center top, right bottom
110. **background attachment**
111. the background-attachment property specifies whether the background image should scroll or be fixed
112. background-attachment: scroll, fixed
113. **background size**
114. background-size: auto, length(width(px,auto),height(px,auto)), cover, percentage(witdth(%,auto),height(%,auto))
115. **background origin**
116. origin of background
117. background-origin: content-box,padding-box,border-box
118. **background clip**
119. how far extends from origin background
120. background-clip: content-box,padding-box,border-box
121. origin and clip shouble be same value
122. **background shorthand**
123. background: url() fixed no-repeat center bottom
124. problem-photo not resize
125. **Module-08: CSS Borders**
126. the css border properties allow you to specify the style, width and color of element’s border.
127. **border style**
128. the border style property specifies what kind of border to display.
129. border-style = dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden
130. border-style = dotted dashed solid double
131. **border width**
132. the border-width property specifies the width of the four boarders.
133. border-width= value, value value value value, value value;
134. **border color**
135. border color= red,red green, red green blue gray;
136. the border color property is used to set the color of the four boarders.
137. **Border shorthand**
138. border: witdth style color
139. **rounded border**
140. border-radius = width
141. **Module-9: CSS Margin**
142. margins are used to create space around elements, outside of any defined borders.
143. **Margin individual sides**
144. margin-top = auto, length, %, inherit
145. margin-bottom = ||
146. margin-right = ||
147. margin-left = ||
148. **margin shorthand**
149. margin: width width width width
150. **margin auto value**
151. you can set the margin property to auto to horizontaly center the element within this container.
152. **Module-10: CSS Padding**
153. padding is used to create space around an element’s content, inside of any defined borders.
154. **Padding individual sides**
155. padding-top = length, % , inherit
156. padding-bottom = ||
157. padding-left = ||
158. padding-right = ||
159. **padding shorthand**
160. padding: width width width width
161. padding: width
162. **padding and element width**
163. width element if you set padding the width will measure with padding and increased it.
164. Solve this use box-sizing = border-box;
165. **Module-11: CSS Height, Width, Min,Max**
166. the css height and width properties are used to set the height and width of an element.
167. **Setting height and width**
168. height = auto,length, %, inherit
169. width = ||
170. min-height = ||
171. max-height = ||
172. min-width = ||
173. max-width = ||
174. this height and width are measurement without of padding and border measurement.
175. **Actual size of width and height**
176. you must be include padding and border for getting any element witdth and height value
177. **Module-12: CSS Box Model**
178. all html elements can be considered as boxes
179. **css boxes**
180. content => padding => border => margin
181. **Module-13: CSS Outline**
182. an outline is a line drawn outside the element’s border.
183. **Outline properties**
184. outline-style = dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden
185. outline-color = rgb,name,hex
186. outline-width = thin, medium, thick, specific(px,pt,cm,em)
187. outline-offset = space
188. **border vs outline**
189. the outline is drawn outside the element’s border, and may overlap other content. Also the outline is not a part of the element’s dimensions; the element’s total width and height is not effected by the width of the outline.
190. **Outline offset**
191. the outline-offset property adds space between an outline and the edge/border of an element.
192. Outline-offset = mesure
193. **Module-14: CSS Text**
194. css has a lot of properties for formatting text.
195. **Css text properties**
196. color = color\_name, color\_code
197. background-color = color\_name, color\_code
198. **css text alignment**
199. text-align = left,right,center,justify
200. text-align-last = left, right, center, justify
201. **css text direction**
202. direction = rtl,ltr
203. unicode-bidi = bidi-override
204. **css vertical alginment**
205. vertical-align = baseline,text-bottom,text-top,sub,super
206. **css text decoration**
207. text-decoration-line = overline, underline, line-through
208. text-decoration-style = dashed,dotted,solid,..
209. text-decoration-color = color\_name
210. text-decoration-thickness = measure
211. text-decoration = none
212. shorthand =
213. **css text transform**
214. the text-transform property is used to specify uppercase and lowercase letter in a text
215. text-transform = uppercase,lowercase,capitalize
216. **css text spacing**
217. text-indent = measure
218. letter-spacing = measure
219. line-height = measure
220. word-spacing = measure
221. white-space = nowrap,normal,pre
222. **css text shadow**
223. the text-shadow property adds shadow to text.
224. text-shadow = value\_color
225. **Module-15: CSS Font**
226. choosing the right font of your website is importan.
227. Choosing the right font has a huge impact on how the readers experience a website .
228. The right font can create a strong identity for your brand.
229. **Generic font families**
230. in css there are five generic font families:
231. serif = times new roman, georgia, garamond
232. sans-serif = arail,verdana, helvetica
233. monospace = courier new, lucida consol, monaco
234. cursive = brush script mt, lucida handwriting
235. fantasy = copperplate, papyrus
236. **css font family property**
237. in css we use the font-family property to specify the font of a text.
238. If the font name is more than one word, it must be in quotation marks, like: “Times New Roman”
239. the font-family property should hold several font name as a “fallback” system.like: “Times New Roman”, arial, monosapce.
240. font-family = “Times New Roman”, arial, times
241. font-style = normal,italic,oblique
242. font-weight = normal,bold
243. font-size = measure
244. font-variant = normal,small-caps
245. **css web safe fonts**
246. web safe fonts are fonts that are universally installed across all browsers and devices.
247. Generic fonts are called web safe fonts
248. **css font fallbacks**
249. however, there are no 100% completely web safe fonts. There is always a chance that a font is not found or is not installed properly.
250. Therefore, it is very important to always use fallbacks fonts.
251. This means that you should add a list of similar “backup fonts” in the font-family property. If the first font does not work, the browser will try next one, and so on.
252. **Css google fonts**
253. just easy to add google fonts
254. **css font shorthand**
255. font = “Times New Roman”, arial, sans-serif 30px bold
256. **Module-16: CSS Icons**
257. icons can easily be added to your html page, by using an icon library.
258. **Fontawesome icons**
259. **bootstrap icons**
260. **google icons**
261. **Module-17: CSS Links**
262. with css, links can be styled in many different ways.
263. **Four links states**
264. a:link
265. a:visited
266. a:hover
267. a:active
268. text-decoration: none,underline,over-line
269. **Module-18: CSS Lists**
270. **different list item markers**
271. list-style-type =
272. disc
273. circle
274. squre
275. none
276. decimal
277. decima-leading-zero
278. lower-alpha
279. upper-alpha
280. lower-roman
281. upper-roman
282. **An image as The List Item Marker**
283. list-style-image = url();
284. **position the list item markers**
285. list-style-position = inside,outside
286. **remove default settings**
287. list-style-type = none
288. margin = 0
289. padding = 0
290. **list shorthand property**
291. list-style = inside circle url()
292. **Module-19: CSS Tables**
293. **table borders**
294. table,th,td = border\_shorthand
295. **full width table**
296. width = 100%
297. **collapse double borders**
298. collapse-border = collapse
299. **table width and height**
300. th = width = 100px
301. **table alignment**
302. text-align = center
303. vertical-align = baseline
304. **table style**
305. table padding
306. hoverable table
307. striped table = tr:nth-child(even) = background-color = red
308. **responsive table**
309. overflow-x: auto
310. **Module-20: CSS Display**
311. the display property is the most important css property for controlling layout
312. block level element
313. inline element
314. **the display property**
315. display = inline, block, none, inline-block
316. **override the default display value**
317. display = inline, block, none
318. **Module-21: CSS Position**
319. the position property specifies the type of position method used for an element (static,relative,fixed,absolute,sticky)
320. **the position property**
321. static
322. relative
323. fixed
324. absolute
325. sticky
326. **position static**
327. html elements are positioned static by default.
328. Not working top,left,right,bottom property
329. not overflow
330. **Position relative**
331. working top,left,right,bottom porperty
332. not overflow
333. and absolute element includes it.
334. **Position absolute**
335. working top,left,right,bottom property
336. not move from into relative property block.
337. Over flow will be.
338. **Position fixed**
339. working top,left,right,bottom property
340. move from into relative property block
341. over flow will be
342. not scrolling
343. **position sticky**
344. working top,left,right,bottom property
345. working two way relative and fixed
346. until reach top,bottom width he is relative then fixed
347. **Module-22: CSS Z-Index**
348. the z-index property specifies the stack order of an element.
349. When elements are positioned, they can overlap other elements.
350. An element can have a positive or negetive stack order.
351. Z-index only works on positioned elements
352. **the z-index property**
353. z-index = -1,1
354. **Module-23: CSS Overflow**
355. the css overflow property controls what happens to content that is too big to fit into an area.
356. The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.
357. **Css overflow property**
358. overflow = visible, hidden, auto, scroll
359. overflow = visible is default
360. overflow-x = ||
361. overflow-y = ||
362. overflow-wrap = normal, break-word, anywhere
363. **Module-24: CSS Float and Clear**
364. the css float property specifies how an element should float.
365. The css clear property specifies what elements can float beside the cleared element and on which side.
366. **The float property**
367. float = left, right, none, inherit
368. using float position is will be absolute
369. **the clear property**
370. clear = left, right, both, none, inherit
371. the clear property will not be space of float element.
372. **Module-25: CSS Inline-Block**
373. **advantages**
374. can possible set width and height
375. respected top bottom margin and padings
376. display = inline-block
377. **Module-26: CSS Align**
378. **center align elements**
379. margin = auto
380. **center align text**
381. text-align = center
382. **center an image**
383. display = block
384. margin = auto
385. width = auto
386. **left and right align using position**
387. position = absolute
388. left = 0
389. right = 0
390. **left and rigth align using float**
391. float = left
392. float = right
393. **center vertically using padding**
394. padding: 90px 0;
395. **cener vertically using line-height**
396. height = 299px
397. line-height = 299px
398. **center vertically using position and transform**
399. **center vertically using flexbox**
400. **Module-27: CSS Opacity**
401. the opacity property specifies the opacity / transparancy of an element.
402. Opcacity value 0.0 – 1
403. opacity value 1 is default
404. **background opacity**
405. background = rgba(red,green,blue,alpha)
406. **Module-28: CSS Navigation**
407. vertical navigation
408. horizontal navigation
409. **Module-29: CSS Dropdown**
410. **Module-30: CSS Forms**
411. the look of an html form can be greatly improved with css.
412. :focus
413. **Module-31: CSS Layouts**
414. header
415. navigation menu
416. contetnt
417. footer
418. **Module-32: CSS Units**
419. **Module-33: CSS Specificity**
420. if there are two or more css rules that point to the same element, the selector with the highest specificity value will win, and it’s style declaration will be applied to that html element.
421. **Specificity hierarchy**
422. inline styles
423. id
424. classes, pseudo-classes, attribute selectors
425. elements and pseudo-elements
426. **Module-34: CSS !important**
427. the !important rule in css is used to add more importance to a property / value than value.
428. In fact, if you use the !important rule, it will override all previous styling rules for that specific property on that element!
429. P = style\_value !important
430. **Module-35: CSS Math Functions**
431. the css math functions allow mathematical expressions to be used as property values,
432. **math functions**
433. calc() = set a calculate expression
434. min() = set min value separated by comma
435. max() = set maximum value separated by comma

**Advanced CSS**

1. **Module-01: CSS Rounded Corners**
2. with the css border-radius property, you can give any element “rounded corners”
3. the css border-radius property defines the radius of an element’s corner.
4. **Border radius property**
5. border-top-left-radius = measure
6. border-top-right-radius = measure
7. border-bottom-left-radius = measure
8. border-bottom-right-radius = measure
9. **Border radius shorthand**
10. border-radius = measure = four round
11. border-radius = measure measure = top-left bottom-right
12. border-radius =
13. border-radius = measure measure measure measure
14. **Module-02: CSS Border Image**
15. with the css border-image property, you can set an image to be used as the border around an element.
16. **Css border image property**
17. border-image-source = url()
18. border-image-slice = measure
19. border-image-width = measure
20. border-image-outset = measure
21. border-image-repeat = repeat, stretch, round, space
22. **css border image shorthand**
23. border-image =
24. **Module-03: CSS Backgrounds**
25. css allows you to add multiple background images for an element, through the background image property.
26. The different background images are separated by commas, and the images are stacked on top of each other, where the first image is closest to the viewer.