

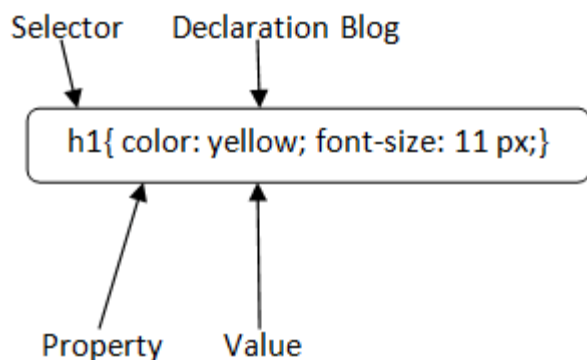
CSS Tutorial

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing.

Why use CSS

- Solves a big problem of writing a code
- Saves a lot of time we can modify the style of our webpage by just changing a single css file instead of changing style for every element
- Provide more attributes than html

CSS Syntax



-
- Each declaration block contains property name and value separated by colon
- `Selector{Property1: value1; Property2: value2;;}`

CSS Selector

- CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

- CSS Element Selector
- CSS Id Selector
- CSS Class Selector
- CSS Universal Selector
- CSS Group Selector

1) CSS Element Selector

- selects the HTML element by name
- ```
p{
```
- ```
  text-align: center;
```
- ```
 color: blue;
```
- ```
}
```
- **<p>**This style will be applied on every paragraph.**</p>**
- **Here styling is by using element**

2)CSS Id Selector

- The id selector selects the id attribute of an HTML element to select a specific element.
- An id is always unique within the page so it is chosen to select a single, unique element
- **<style>**
- ```
#para1 {
```
- ```
  text-align: center;
```
- ```
 color: blue;
```
- ```
}
```
- **</style>**
- **<p id="para1">**Hello Jeevan**</p>**

3) CSS Class Selector

- The class selector selects HTML elements with a specific class attribute.
- Class name shouldn't start with a number
- **<style>**
- ```
.center {
```
- ```
  text-align: center;
```
- ```
 color: blue;
```
- ```
}
```
- **</style>**
- **<h1 class="center">**This heading is blue and center-aligned.**</h1>**
- If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

4) CSS Universal Selector

- It selects all the elements on the pages.
- **<style>**
- ```
* {
```
- ```
  color: green;
```
- ```
 font-size: 20px;
```
- ```
}
```

- `</style>`

5) CSS Group Selector

- grouping selector is used to select all the elements with the same style definitions.
- Commas are used to separate each selector in grouping.
- `h1,h2,p {`
- `text-align: center;`
- `color: blue;`
- `}`

How to add CSS

There are three ways to insert CSS in HTML documents.

- Inline CSS
- Internal CSS
- External CSS

1) Inline CSS

- apply CSS on a single line or element.
- If you want to use inline CSS, you should use the style attribute to the relevant tag.
- `<p style="color:blue">Hello CSS</p>`
- `<h2 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h2>`

Disadvantages of Inline CSS

- cannot use quotations within inline CSS.
- cannot be reused anywhere else.
- tough to be edit

2) Internal CSS

- It can affect all the elements of the page.
- used to add a unique style for a single document.
- `<style>`
- `p{color:blue}`
- `</style>`

3) External CSS

- External CSS is used to apply CSS on multiple pages or all pages
- `<link rel="stylesheet" type="text/css" href="style.css">` link tag must be used inside head section of html.
- should not use a space between the property value and the unit.
- should be `margin-left:20px` not `margin-left:20 px`.

CSS Comments

- Comments are single or multiple lines statement and written within `/*.....*/`

CSS Background

- used to define the background effects on element

5 CSS background properties that affects the HTML elements:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

1) CSS background-color

- `<style>`
- `h2,p{`
- `background-color: #b0d4de;`
- `}`
- `</style>`

2) CSS background-image

- `<style>`
- `body {`
- `background-image: url("paper1.gif");`
- `margin-left:100px;`
- `}`
- `</style>`

3) CSS background-repeat

- image repeated horizontally only.
- `<style>`
- `body {`
- `background-image: url("gradient_bg.png");`
- `background-repeat: repeat-x;`
- `}`
- `</style>`
- Image repeated vertically
- `<style>`
- `body {`

- background-image: url("gradient_bg.png");
- background-repeat: repeat-y;
- }
- **</style>**

4) CSS background-attachment

- used to specify if the background image
- If you set fixed the background image then the image will not move during scrolling in the browser
- **<style>**
- body {
- background: white url('bbb.gif');
- background-repeat: no-repeat;
- background-attachment: fixed;
- margin-left:200px;
- }
- **</style>**

5) CSS background-position

- used to define the initial position of the background image
- default, the background image is placed on the top-left of the webpage.
- Center,top,bottom,left,right are possible background positions
- **<style>**
- body {
- background: white url('good-morning.jpg');
- background-repeat: no-repeat;
- background-attachment: fixed;
- background-position: center;
- }

CSS Border

- used to set the border on an element

1) CSS border-style

- none , dotted , dashed , solid , double , groove , ridge , inset , outset are values for the border-style
- p {
- border-style: none;
- }

2) CSS border-width

- border-width property is used to set the border's width

- You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.
- This tag is combinedly used with the border-style attribute otherwise it wont works

3) CSS border-color

- This tag is combinedly used with the border-style attribute otherwise it wont works
- p {
- border-style: solid;
- border-color: red;
- }

CSS Display

- used to control the layout of the element
- specifies how the element is displayed.
- display:value; is the basic syntax

There are following CSS display values which are commonly used.

- display: inline;
- display: inline-block;
- display: block;
- display: run-in;
- display: none;

1) CSS display inline

- doesn't force the line break
- so the flow of text doesn't break in inline example.
- **<style>**
- p {
- display: inline;
- }
- **</style>**
- **<p>**Java Tutorial.**</p>**
- **<p>**SQL Tutorial.**</p>**
- **<p>**HTML Tutorial.**</p>**
- **<p>**CSS Tutorial.**</p>**

2) CSS display inline-block

- **<style>**
- p {
- display: inline-block;

- }
- `</style>`
- `<p>Java Tutorial.</p>`
- `<p>SQL Tutorial.</p>`
- `<p>HTML Tutorial.</p>`
- `<p>CSS Tutorial.</p>`
- Inline block means it considers each element as a block so if page width is not sufficient then the whole block shifts to the next line

3) CSS display block

- CSS display block element takes as much horizontal space as they can.
- Means the block element takes the full available width.
- They make a line break before and after them.
- `<style>`
- `p {`
- `display: block;`
- `}`
- `</style>`
- `<p>Java Tutorial.</p>`
- `<p>SQL Tutorial.</p>`
- `<p>HTML Tutorial.</p>`
- `<p>CSS Tutorial.</p>`

CSS Float

- **CSS float property** is a *positioning property*
- used to *push an element to the left or right*
- generally used with images and layouts.
- Elements float from left or right, not up or down
- `<style>`
- `img {`
- `float: right;`
- `}`
- `</style>`

CSS Font

- By the use of CSS font property you can change the text size, color, style and more

1) CSS Font Color

- used to change the color of the text.
- We can define color name using a font color name, hexa decimal value, rgb color constant
- `<style>`

- body {
- font-size: 100%;
- }
- h1 { color: red; }
- h2 { color: #9000A1; }
- p { color:rgb(0, 220, 98); }
- }
- **</style>**

2) CSS Font Family

- Divided into two types
- Generic family: It includes Serif, Sans-serif, and Monospace.
- Font family: It specifies the font family name like Arial, New Times Roman etc.
- **<style>**
- body {
- font-size: 100%;
- }
- h1 { font-family: sans-serif; }
- h2 { font-family: serif; }
- p { font-family: monospace; }
- }
- **</style>**

3) CSS Font Size

- change the size of the font
- **style="font-size:x;"**
- x value can be xx-small,x-small,small,medium,large,x-large,xx-large,smaller,larger,size in pixels
- this style will be defined in tag

4) CSS Font Style

- what type of font you want to display
- **<style>**
- body {
- font-size: 100%;
- }
- h2 { font-style: italic; }
- h3 { font-style: oblique; }
- h4 { font-style: normal; }
- }
- **</style>**

5) CSS Font Variant

- `<style>`
- `p { font-variant: small-caps; }`
- `h3 { font-variant: normal; }`
- `</style>`

6) CSS Font Weight

- `<p style="font-weight:bold;">This font is bold.</p>`
- Values can be bold bolder lighter or either 100 to 900

CSS Line Height

- sets the differences between two lines of your content.

CSS line-height values

- you can set the line-height value in percentage
- `<style>`
- `h3.small {`
- `line-height: 70%;`
- `}`
- `h3.big {`
- `line-height: 200%;`
- `}`
- `</style>`

CSS Margin

- used to define the space around elements.
- clears an area around the element.
- Margin, margin-left, margin-right, margin-top, margin-bottom/
- Auto, length(px), %, inherit are the possible values of margin element
- `<style>`
- `p {`
- `background-color: pink;`
- `}`
- `p.ex {`
- `margin-top: 50px;`
- `margin-bottom: 50px;`
- `margin-right: 100px;`
- `margin-left: 100px;`
- `}`
- `</style>`

margin: 50px 100px 150px 200px;

- Top,right,bottom,left

margin: 50px 100px 150px;

- Top,left and right,bottom

margin: 50px 100px;

- Top and bottom, left and right

margin: 50px;

- Top right bottom left

CSS Opacity

- CSS opacity property is used to specify the transparency of an element

CSS Opacity Example

- `<style>`
- `img {`
- `opacity: 0.4;`
- `}`
- `</style>`

CSS Overflow

- **CSS overflow property** specifies how to handle the content when it overflows its block level container.
- The overflow property specifies what to do if the content of an element exceeds the size of the element's box
- `<style>`
- `div.scroll {`
- `background-color: #00ffff;`
- `width: 100px;`
- `height: 100px;`
- `overflow: scroll;`
- `}`
- `div.hidden {`
- `background-color: #00FF00;`
- `width: 100px;`
- `height: 170px;`
- `overflow: hidden;`
- `}`
- `</style>`

Value	Description
visible	It specifies that overflow is not clipped. it renders outside the element's box.this is a default value.
hidden	It specifies that the overflow is clipped, and rest of the content will be invisible.
scroll	It specifies that the overflow is clipped, and a scroll bar is used to see the rest of the content.
auto	It specifies that if overflow is clipped, a scroll bar is needed to see the rest of the content.
inherit	It inherits the property from its parent element.
initial	It is used to set the property to its initial value.

CSS Padding

- *define the space between the element content and the element border.*
- CSS margin defines the space around elements
- Top, bottom, left and right padding can be changed independently
- Padding-left,padding-right,padding-top,padding-bottom are the possible properties of padding
- The values can be in percentage or length(px)
- **<style>**
- p {
- background-color: pink;
- }
- p.padding {
- padding-top: 50px;
- padding-right: 100px;
- padding-bottom: 150px;
- padding-left: 200px;

- }
- `</style>`

CSS Position

- **CSS position property** is used *to set position for an element*
- used to place an element behind another
- position element's computed position property is relative, absolute, fixed or sticky.

CSS positioning:

- CSS Static Positioning
- CSS Fixed Positioning
- CSS Relative Positioning
- CSS Absolute Positioning

1) CSS Static Positioning

- always positions an element according to the normal flow of the page.

2) CSS Fixed Positioning

- fixed positioning property helps to put the text fixed on the browser.
- This fixed text is positioned relative to the browser window, and doesn't move even you scroll the window.

- `<style>`
- `p.pos_fixed {`
- `position: fixed;`
- `top: 50px;`
- `right: 5px;`
- `color: blue;`
- `}`
- `</style>`

3) CSS Relative Positioning

- used to set the element relative to its normal position

- `<style>`
- `h2.pos_left {`
- `position: relative;`
- `left: -30px;`
- `}`
- `h2.pos_right {`
- `position: relative;`
- `left: 30px;`
- `}`

➤ `</style>`

4) CSS Absolute Positioning

➤ absolute positioning is used to position an element relative to the first parent element that has a position other than static. If no such element is found, the containing block is HTML.

➤ `<style>`

➤ `h2 {`

➤ `position: absolute;`

➤ `left: 150px;`

➤ `top: 250px;`

➤ `}`

➤ `</style>`

CSS Vertical Align

value	description
baseline	It aligns the baseline of element with the baseline of parent element. This is a default value.
length	It is used to increase or decrease length of the element by the specified length. negative values are also allowed.
%	It is used to increase or decrease the element in a percent of the "line-height" property. negative values are allowed.
sub	It aligns the element as if it was subscript.
super	It aligns the element as if it was superscript.
top	It aligns the top of the element with the top of the tallest element on the line.
bottom	It aligns the bottom of the element with the lowest element on the line.
text-top	the top of the element is aligned with the top of the parent element's font.
middle	the element is placed in the middle of the parent element.
text-	the bottom of the element is aligned with the bottom of the parent

bottom	element's font.
initial	It sets this property to Its default value.
inherit	inherits this property from Its parent element.

CSS White Space

➤ used to handle the white spaces inside an element.

Value	Description
normal	This is a default value. in this value, text is wrapped when necessary. sequences of white space will collapse into a single whitespace.
nowrap	Sequences of white space will collapse into a single whitespace. in this value, text will never wrap to the next line and only break when <code>
</code> tag is used.
pre	Whitespace is preserved by the browser. it is act like html <code><pre></code> tag. text will only wrap on line breaks.
pre-line	Sequences of white space will collapse into a single whitespace. texts are wrapped when necessary, and on line break.
pre-wrap	Whitespace is preserved by the browser. texts are wrapped when necessary, and on line break.
initial	It sets this property to its default value.
inherit	It inherits this property from its parent element

➤ `<html>`

➤ `<head>`

➤ `<style>`

➤ `p {`

➤ `white-space: nowrap;`

➤ `}`

➤ `</style>`

➤ `</head>`

➤ `<body>`

➤ `<p>`

- `</p>`
- `</body>`
- `</html>`

CSS Width

- **CSS width property** is used to set the width of the content area of an element
- does not include padding borders or margins

- **<style>**
- p.ex {
 - height: 150px;
 - width: 150px;
- }
- **</style>**
-
- **<style>**
- img.small {
 - width: 10%;
- }
- **</style>**

CSS Word Wrap

- **CSS word wrap property** is used to break the long words and wrap onto the next line.
- Normal,break-word are two values for the word-wrap
- Normal is used to break the words only at breakable points
- Break-word is used to break the word anywhere as possible
- **<style>**
- p.test {
 - width: 11em;
 - word-wrap: break-word;
- }
- **</style>**
- **<body>**
- **<p class="test">** In this paragraph, there is a very long word:
iamsooooooooooooooooooooooooooooooollongggggggggggggggg.The long word will
break and wrap to the next line.**</p>**
- **</body>**

CSS Outline

- just like CSS border property
- draw an extra border around an element to get visual attention.
- **<head>**
- **<style type="text/css">**
- **.box {**
- **outline: 3px solid red;**
- **border: 3px solid lightgreen;**
- **padding: 5px 10px;**
- **}**
- **</style>**
- **</head>**
- **<body>**
- **<div class="box">Welcome Jeevan</div>**
- **</body>**
- **Not possible to apply a different outline width,style,color for the four sides individually**
- **<style type="text/css">**
- **.box {**
- background-color: #eee;
- border: 3px solid Lightgreen;
- padding: 5px 10px;
- outline-width: 3px;
- outline-style: solid;
- outline-color: red;
- **}**
- **</style>**
- **<div class="box">Welcome Jeevan</div>**
- outline style i.e. hidden, dotted, dashed, solid, double, groove, ridge, inset and outset

Outline offset

- outline offset is used to create a distance between outline and border.
- **<style type="text/css">**
- **.box {**
- background-color: #eee;
- outline: 3px solid red;
- outline-offset: 6px;
- border: 3px solid Lightgreen;
- padding: 5px 10px;
- **}**
- **</style>**
- **<div class="box">Welcome Jeevan</div>**

CSS Visibility

- used to specify whether an element is visible or not
- **Syntax:** visibility: visible|hidden|collapse|initial|inherit;
- `<head>`
- `<style>`
- `h1.visible {`
- `visibility: visible`
- `}`
- `h1.hidden {`
- `visibility: hidden`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1 class="visible">I am visible</h1>`
- `<h1 class="hidden">I am invisible</h1>`
- `<p>Note:` An invisible element also take up the space on the page.
- By using display property you can create invisible elements that don't take space.`</p>`
- `</body>`

CSS Tooltips

- display extra information about something when the user moves the mouse cursor over an element
- `<html>`
- `<style>`
- `.tooltip {`
- `position: relative;`
- `display: inline-block;`
- `border-bottom: 1px dotted black;`
- `}`
- `.tooltip .tooltiptext {`
- `visibility: hidden;`
- `width: 120px;`
- `background-color: red;`
- `color: #fff;`
- `text-align: center;`
- `border-radius: 6px;`
- `padding: 5px 0;`
- `position: absolute;`
- `z-index: 1;`

- opacity: 0;
- transition: opacity 5s;
-
- bottom: 100%;
- left: 50%;
- margin-left: -60px;
- }
- .tooltip:hover .tooltiptext {
- visibility: visible;
- opacity: 1;
- }
- </style>
- <body style="text-align:center;">
- <h2>Top Tooltip Example</h2>
- <p>Move your mouse cursor over the below heading</p>
- <div class="tooltip"> Welcome Jeevan
- A solution of all technology.
- </div>
- </body>
- </html>

CSS Tooltip Animation

- tooltip animation is used to fade in the tooltip text when it is about to visible
- CSS3 "transition" property along with "opacity" property is used o attain fade in tooltip or tooltip animation.
- The animation time from being completely invisible to 100% visible is specified in second

CSS Flexbox

- flexbox contains flex containers and flex items
- <!DOCTYPE html>
- <html>
- <head>
- <style>
- body {
- direction: rtl;
- }
- .flex-container {
- display: -webkit-flex;
- display: flex;
- -webkit-flex-direction: row-reverse;//To reverse the flex items
- flex-direction: row-reverse; //To reverse the flex items

- -webkit-justify-content: space-around;
- justify-content: space-around;
- -webkit-align-items: stretch;
- align-items: stretch;
- -webkit-flex-wrap: nowrap;
- flex-wrap: nowrap;
- -webkit-align-content: center;
- align-content: center;
-
- width: 400px;
- height: 200px;
- background-color: lightpink;
- }
- .flex-item {
- background-color: brown;
- width: 100px;
- height: 100px;
- margin: 10px;
- }
- </style>
- </head>
- <body>
- <div class="flex-container">
- <div class="flex-item">flex item 1</div>
- <div class="flex-item">flex item 2</div>
- <div class="flex-item">flex item 3</div>
- </div>
- </body>
- </html>

Flex Direction Property

- **Flex** default value is row (left-to-right, top-to-bottom).

The other possible values are:

- row-reverse
- column
- column-reverse
- these are all the possible keywords for the flex properties

Flexbox Justify-Content

- It is a sub-property of CSS3 Flexbox layout module.

Its possible values are:

- **flex-start:**It is the default value. It sets the items at the beginning of the container
- **flex-end:**It sets the items at the end of the container.
- **Center:**It sets the items at the center of the container.
- **space-between:**It sets the items with space between the lines.
- **space-around:**It sets the items with space before, between, and after the lines.

Flexbox align-items

- set the flexible container's items vertically align when the items do not use all available space on the cross-axis.
- **Its possible values are:**
- **stretch:**It is the default value. It specifies that Items are stretched to fit the container.
- **flex-start:**It sets the items at the top of the container.
- **flex-end:**It sets the items at the bottom of the container.
- **center:**It sets the items at the center of the container (vertically).
- **baseline:**It sets the items at the baseline of the container.
- Let's take some example to demonstrate the usage of above values.

Flexbox flex-wrap

- specifies if the flex-items should wrap or not
- in the case of not enough space for them on one flex line.

Its possible values are:

- **nowrap:** It is the default value. The flexible items will not wrap.
- **wrap:** It specifies that the flexible items will wrap if necessary.
- **wrap-reverse:** It specifies that the flexible items will wrap, if necessary, in reverse order.

Flexbox align-content property

- aligns flex lines instead of flex items.

possible values are:

- **stretch:**It is the default value. It specifies lines stretch to take up the remaining space.
- **flex-start:**It specifies that lines are packed toward the start of the flex container.
- **flex-end:**It specifies that lines are packed toward the end of the flex container.
- **center:**It specifies that lines are packed toward the center of the flex container.

- **space-between:** It specifies that lines are evenly distributed in the flex container.
- **space-around:** It specifies that lines are evenly distributed in the flex container, with half-size spaces on either end.

Responsive Web Design

- Smartphone: 320px
- Tablet: 768px
- Netbook: 1024px
- Desktop: 1600px
- `<html>`
- `<head>`
- `<meta name="viewport" content="width=device-width, initial-scale=1.0"/>`
- `<style>`
- `body {`
- `background-color:yellow;`
- `}`
- `@media only screen and (max-width: 500px) {`
- `body {`
- `background-color:green;`
- `}`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<p>` If you resize the browser window and the width of this document is less than 500 pixels, the background-color is "green", otherwise it is "yellow" `</p>`
- `</body>`
- `</html>`

CSS Transforms

- transform property facilitates you to translate, rotate, scale, and skews elements
- Transformation is an effect that is used to change shape, size and position

CSS 2D Transforms

translate() method

- `transform: translate(100px,80px); /* Standard syntax */`
- translate() method is used to move an element from its current position according to the given parameters

rotate() method

- `transform: rotate(30deg); /* Standard syntax */`

- rotate() method is used to rotate an element clockwise or anti-clockwise

scale() method

- transform: scale(2.5,2); /* Standard syntax */
- scale() method is used to increase or decrease the size of an element according to the given width and height.

skewX() method

- transform: skewX(30deg); /* Standard syntax */
- skewX() method is used to skew an element along the X axis according to the given angle

skewY() method

- transform: skewY(30deg); /* Standard syntax */
- skewY() method is used to skew an element along the Y axis according to the given angle.

skew() method

- transform: skew(30deg,20deg); /* Standard syntax */
- skew() method is used to skew an element along with X-axis and Y-axis according to the given angle

matrix() method

- parameters of matrix method:
matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())
- transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */
- transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */
- **<html>**
- **<head>**
- **<style>**
- div {
- width: 300px;
- height: 100px;
- background-color: lightpink;
- border: 1px solid black;
- }
- div#myDiv1 {
- transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */
- }
- div#myDiv2 {
- transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */
- }
- **</style>**
- **</head>**

- `<body>`
- `<p>`The matrix() method combines all the 2D transform methods into one.`</p>`
- `<div>`
- This is a normal div element.
- `</div>`
- `<div id="myDiv1">`
- Using the matrix() method.
- `</div>`
- `<div id="myDiv2">`
- Another use of the matrix() method.
- `</div>`
- `</body>`
- `</html>`

CSS Aural Media/ Style Sheets

- synthesis and sound effect to facilitate users to listen to information instead of reading them.

CSS User Interface

- resizing elements, outlines and box sizing.
- resize and outline-offset are the most important properties of the CSS user interface

CSS3 Resize property

- specifies that whether an element should be resized by the user or not

1)Horizontal Resize

- `<style>`
- `div {`
- `border: 2px solid;`
- `padding: 20px;`
- `width: 300px;`
- `resize: horizontal;`
- `overflow: auto;`
- `}`
- `</style>`
- Overflow is compulsory needed or resize option wont work

2)Vertical Resize

- `<style>`
- `div {`
- `border: 2px solid;`
- `padding: 20px;`
- `width: 300px;`

- `resize: vertical;`
- `overflow: auto;`
- `}`
- `</style>`
- If we need both horizontal and vertical then we need to use the value for resize as "both"

CSS3 Outline Offset

- `<style>`
- `div {`
- `margin: 20px;`
- `padding: 10px;`
- `width: 300px;`
- `height: 100px;`
- `border: 1px solid black;`
- `outline: 1px solid red;`
- `outline-offset: 10px;`
- `}`
- `</style>`

CSS Pagination

- (<https://www.javatpoint.com/css-pagination>)

LIST OF CSS PROPERTIES

- `color:white;`
- `background-color:red;`
- `padding:5px;`
- `text-align: center;`
- `margin-left:40px;`

- `background-image: url("paper1.gif");`
- `background-repeat: repeat-x;`

- `background-attachment: fixed;`
- `background-position: center;`
-