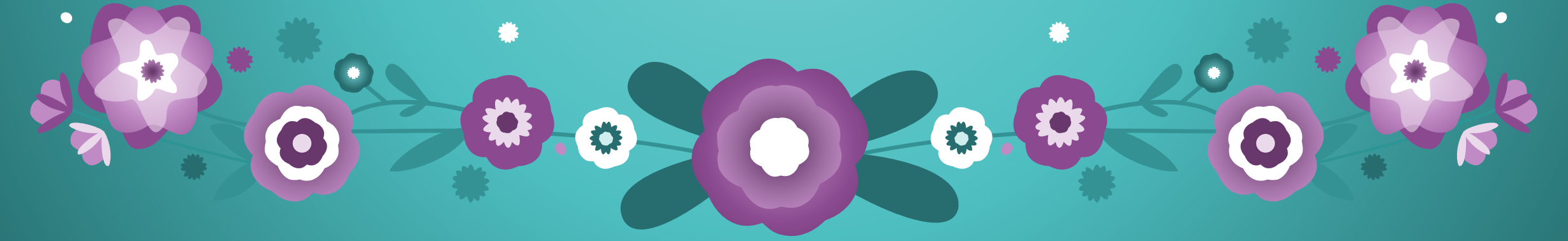
CSS

Facilitated By Krishna Prasad Acharya
Mechi Multiple Campus
Bhadrapur



CSS :Cascading Style Sheets

- CSS is a simple design language intended to simplify the process of making web pages presentable.
- CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.
- CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML

Why Use CSS?

- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Who create and maintain the CSS?

- CSS is created and maintained through a group of people within the W3C(The World Wide Web Consortium) called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.

CSS: **Advantages of CSS**

Advantages of CSS

- **CSS saves time** – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

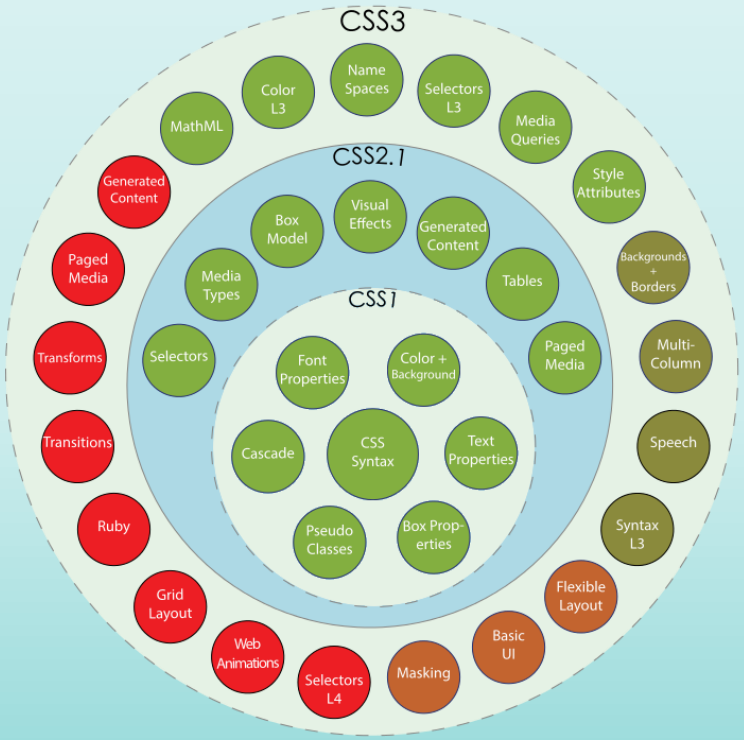
CSS: **Advantages of CSS**

Evaluation of CSS

CSS Level 1 (Dec 1996)
Font properties,
Text attributes
Alignment of text, tables, images and more
Colours of text and backgrounds,
Spacing of words, letters and lines,
Margins, borders, padding and positioning, and
Unique identification and classification of groups of attributes.

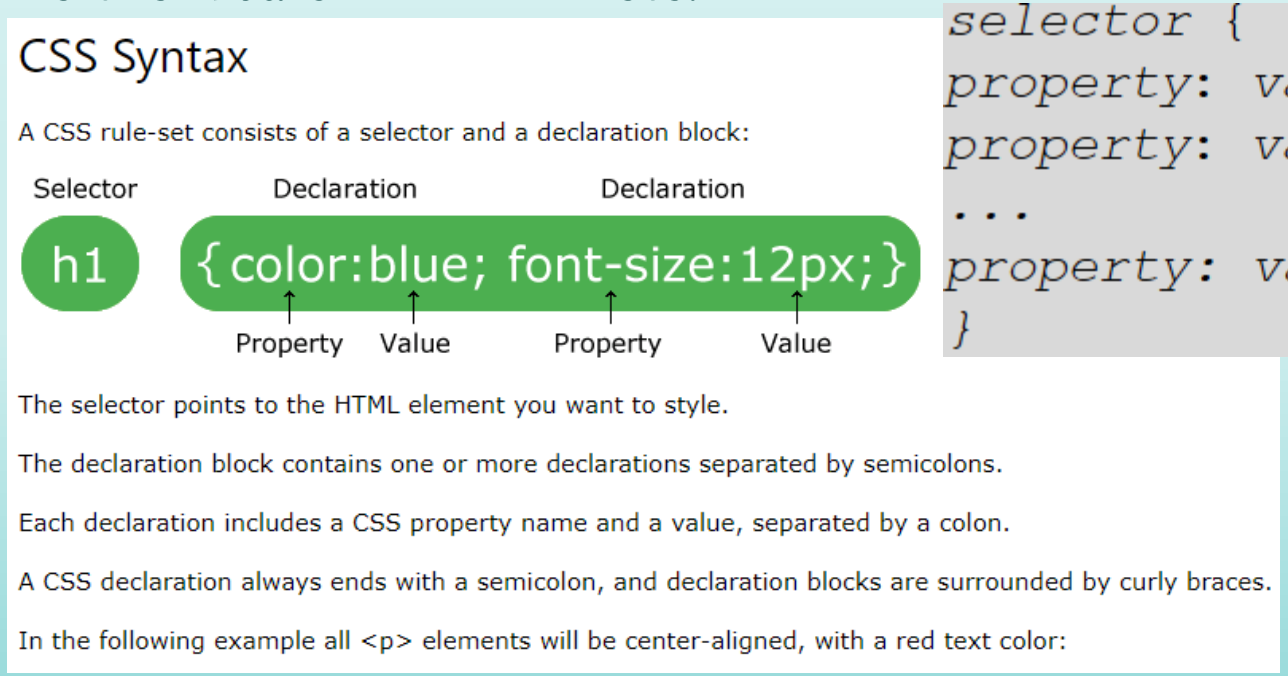
CSS Level 2 (May 1998)
z-index,
Media types
Bidirectional text
Absolute, relative and fixed positioning
Support for aural style sheets.

CSS Level 3 (2011/2012)
Introduction of Modules, namely:
Color
Selectors level 3,
Namespaces and
Media queries.



CSS Basic Syntax

- A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts –
- **Selector** – A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
- **Value** – Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.




```
selector {
property: value;
property: value;
...
property: value;
}
```

```
p {
font-family: sans-serif;
color: red;
}
```

```
p {
  color: red;
  text-align: center;
}
```

Linking Style Sheets to HTML

1. Inline Style

Inline Style is the simplest method of adding CSS styles to your HTML pages. An inline style is applied to an HTML document via its style attribute to specific tags within the document,

For example, If you want to add styles to < p > then you can code like this:

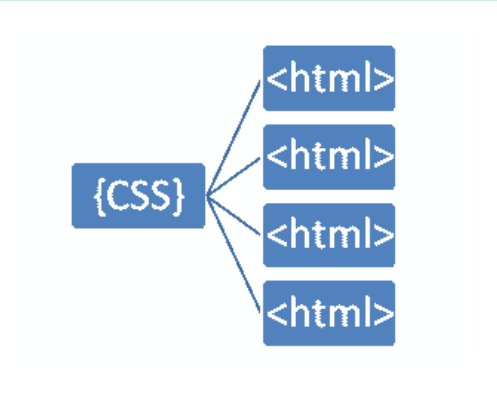
<p style="color: #0000FF">...<p>

<h1 style = "color:#36C; font-weight: normal; "> …</h1>

2. **Embedded Styles** allow you to implement any number of CSS styles by placing them between the opening and closing style tags.

<style>.....</style>

You can place Style Tag within the < head > ... < /head > section, just




```
<head>
        <title>Embedded Style Sample</title>
        <style type="text/css">
                h1{
                        color: #0000FF;
                }
                h2{
                        color: #00CCFF;
                }
        </style>
</head>
```

Linking Style Sheets to HTML

3. External Style

An external style sheet is a plain text file that contain CSS Style formats only. The extension of the external file should end with .css extension (e.g. pagestyle.css). This external file is referred to as an external style sheet.

The external Style Sheet (.css file) always seperate from HTML file.

```
<head>
    <link rel="stylesheet" type="text/css" href="styles.css " />
</head>
```

Alternet way

Imported CSS - @import Rule

- @import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

```
<head>
  @import url("mystyle.css“);
</head>
```

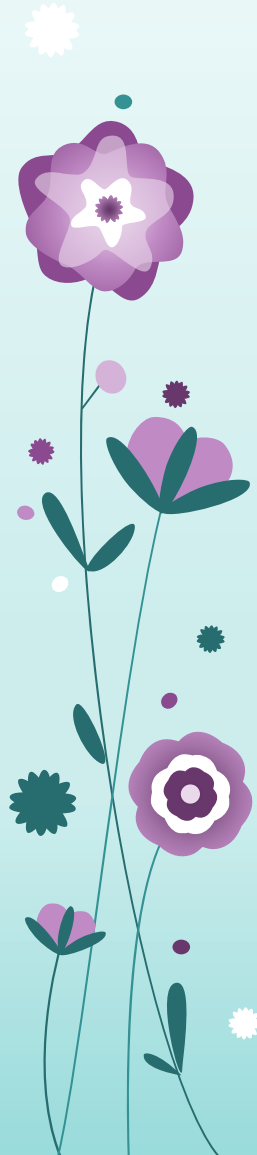
CSS Rules Overriding

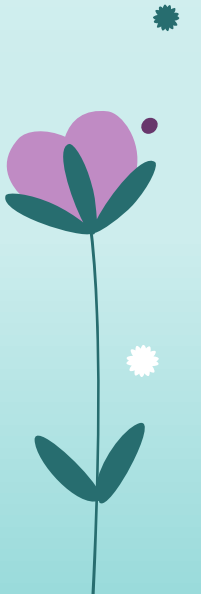
Here is the rule to override any Style Sheet Rule.

- Any **inline style** sheet takes highest priority. So, it will override any rule defined in <style>...</style> tags or rules defined in any external style sheet file.
- Any rule defined in <style>...</style> tags will override rules defined in any external style sheet file.
- Any rule defined in external style sheet file takes lowest priority, and rules defined in this file will be applied only when above two rules are not applicable.

CSS Comments

/*.....this is a comment in style sheet.....*/.





CSS Selectors

1.The element Selector:

The element selector selects elements based on the element name.

```
p {
  text-align: center;
  color: red;
}
```

2.The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- The style rule below will be applied to the HTML element with id="para1":

```
#para1 {
  text-align: center;
  color: red;
}
```

Mechi Multiple Campus

CSS Selectors

3.The class Selector

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.
- In the example below, all HTML elements with class="center" will be red and center-aligned:

```
.center {
  text-align: center;
  color: red;
}
```

```
<h1 class="center">Red and center-aligned heading</h1>
```

```
p.center {
  text-align: center;
  color: red;
}
```

- we can also specify that only specific HTML elements should be affected by a class.
- In the example below, only <p> elements with class="center" will be center-aligned:
- HTML elements can also refer to more than one class.

```
<p class="center large">This paragraph refers to two classes.</p>
```

CSS Selectors

4. Grouping Selectors

To group selectors, separate each selector with a comma.

```css
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}
```

```css
h1, h2, p {
  text-align: center;
  color: red;
}
```

```css
#content, #footer, .supplement {
  position: absolute;
  left: 510px;
  width: 200px;
}
```

CSS Selectors

4. Descendant Selector

Descendant means anywhere nested within it in the tag.

```
<style>
        p a {
                font-family: sans-serif;
                font-size: 15pt;
                line-height: 150%;
        }
</style>
```

```
ul li {  }
header h2  {  }
footer a  {  }
.module div {  }
#info-toggle span  {  }
div dl dt a  {  }
```

CSS Selectors

Selectors	Description
h1, p, span etc.	element selector
.className	class selector
#idName	id selector
*	Universal selector (selects everything)
h1.className	select h1 with class 'className'
h1#idName	select h1 with id 'idName'
p span	descendant selector (select span which is inside p)
p > span	child selector ('span' which is direct descendant of 'p')
h1, h2, p	group selection (select h1, h2 and p)
span[my_id]	select 'span' with attribute 'my_id'
span[my_id=m_span]	select 'span' with attribute 'my_id=m_span'

Pseudo Selectors

CSS pseudo-classes are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects.

selector:pseudo-class {property: value}

```
<head>
   <style type = "text/css">
      a:active {color: #FF00CC}
   </style>
</head>
```

```
<head>
   <style type = "text/css">
      a:hover {color: #FFCC00}
   </style>
</head>
```

:link
Use this class to add special style to an unvisited link.
:visited
Use this class to add special style to a visited link.
:hover
Use this class to add special style to an element when you mouse over it.
:active
Use this class to add special style to an active element.
:active
Use this class to add special style to an active element.
:first-child
Use this class to add special style to an element that is the first child of some other element.
:first-line
Use this element to add special styles to the first line of the text in a selector.
:first-letter
Use this element to add special style to the first letter of the text in a selector.
:before
Use this element to insert some content before an element.
:after
Use this element to insert some content after an element.

```
<style type = "text/css">
   div > p:first-child {
      text-indent: 25px;
   }
</style>
```

```
p:first-letter {
        font-size: 200%;
        font-weight: bold;
}
```

CSS UNITS - Sizes

- Relative length measurements:
  - px (pixels – size varies depending on screen resolution)
  - em (usually the height of a font's uppercase M)
  - ex (usually the height of a font's lowercase x)
  - Percentages (of the font's default size)
- Absolute-length measurements (units that do not vary in size):
  - in (inches)
  - cm (centimeters)
  - mm (millimeters)
  - pt (points; 1 pt = 1/72 in)
  - pc (picas; 1 pc = 12 pt)

- Generally 1em = 12pt = 16px = 100%

%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	p {
		font-size: 16pt;
		line-height: 125%;
		}

cm	Defines a measurement in centimeters.	div {margin-bottom: 2cm;}

em	A relative measurement for height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus 2em = 24pt.	p {
		letter spacing: 7em;
		}

ex	This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x.	p {
		font-size: 24pt;
		line-height: 3ex;
		}

in	Defines a measurement in inches.	p { word-spacing: .15in;}
mm	Defines a measure in millimeters.	p { word-spacing: 15mm;}
pc	Defines a measurement in picas. A pica is equivalent to 12 points. Thus, there are 6 picas per inch.	p { font-size: 20pc;}
pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	body {font-size: 18pt;}
px	Defines a measurement in screen pixels.	p {padding: 25px;}

CSS Font/text

property	values
font-family	font-family: arial, helvetica, serif
font-size	font-size: 16px; [*150%, larger, xx-small, small, medium*]
font-weight	font-weight: bold;[*Normal,bold,bolder,lighter,lightest*]
font-style	font-style: italic;[*Normal,italic,oblique]*
text-decoration	text-decoration:underline;[*none, underline, overline, line-through, blink*]
text-transform	text-transform:capitalize[none, capitalize, uppercase, lowercase]
letter-spacing	letter-spacing: 5px;
word-spacing	word-spacing:5px;
text-indent	text-indent:1cm;
color	*color:red*
text-align	*text-align:right[right/left/center]*
white-space	white-space:pre;[*normal, pre, nowrap*]
direction	direction:rtl;[*ltr/rtl*]
text-shadow	text-shadow:4px 4px 8px blue;[text-shadow: *h-shadow v-shadow blur-radius color*]

CSS List

property	values
list-style-type	none, disc (default), circle, square
list-style-position	None, inside, outside
list-style-image	ist-style: inside url("../img/images.png")
list-style	list-style: outside upper-alph
marker-offset	list-style: inside square; marker-offset:2em;

Value	Description	Example
decimal	Number	1,2,3,4,5
decimal-leading-zero	0 before the number	01, 02, 03, 04, 05
lower-alpha	Lowercase alphanumeric characters	a, b, c, d, e
upper-alpha	Uppercase alphanumeric characters	A, B, C, D, E
lower-roman	Lowercase Roman numerals	i, ii, iii, iv, v
upper-roman	Uppercase Roman numerals	I, II, III, IV, V
lower-greek	The marker is lower-greek	alpha, beta, gamma

CSS Link/Border

property	values
:link	a:link {color: blue} [unvisited hyperlinks]
:visited	a:visited {color:green}[visited hyperlinks]
:hover	a:hover {color:red}[mouse pointer hovering over it.]
:active	a:active {color: black} [user is currently clicking.]
Note	a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective. Also, a:active MUST come after a:hover in the CSS definition.

property	values
border-color border-bottom-color	[bottom, left, top, and right] border-bottom-color:#009900; /* Green */
border-style border-bottom-style	[None, solid, dotted, dashed, double, groove] [bottom, left, top, and right] border-style:solid;
border-width border-bottom-width	[bottom, left, top, and right] border-bottom-width:4px;border-top-width:10px;
Shorthand	border:4px solid red;

CSS - Backgrounds

property	values
background-color	set the background color of an element. <p **style = "background-color:yellow;"**>
background-image	set background image of an element background-image: url("/css/images/css.jpg"); background-color: #cccccc;
background-repeat	used to control the repetition of an image in the background background-repeat: repeat;[no-repeat,repeat-x,repeat-y]
background-position	used to control the position of an image in the background background-position:100px 200px; [from left hand side and top]
background-attachment	used to control the scrolling of an image in the background background-attachment: fixed;[fixed,scroll]
background	used as a shorthand to specify a number of other background properties background:red url(/images/pattern1.gif) repeat fixed

```
background-color: #000;
background-image: url(images/bg.gif);
background-repeat: no-repeat;
background-position: left top;
```

```
background: #000 url(images/bg.gif) no-repeat left top;
```

CSS – Table/Margin

property	values
border-collapse	collapse, separate(default)
border-spacing	border-spacing: 15px
empty-cells:	Hide/show(default)
caption-side	caption-side: bottom;[top, bottom]
table-layout	: auto | fixed | initial | inherit;

property	values
margin-top	margin-top: 100px;
margin-right	margin-right: 100px;
margin-bottom	margin-bottom: 100px;
margin-left	margin-left: 100px;
margin	margin: 25px 50px 75px 100px;[top,right,bottom,left] margin: 25px 50px 75px;[top,right-left,bottom] margin: 25px 50px;[top-bottom,right-left] margin: 25px;[all]
margin: auto	auto to horizontally center the element within its container

CSS – Padding

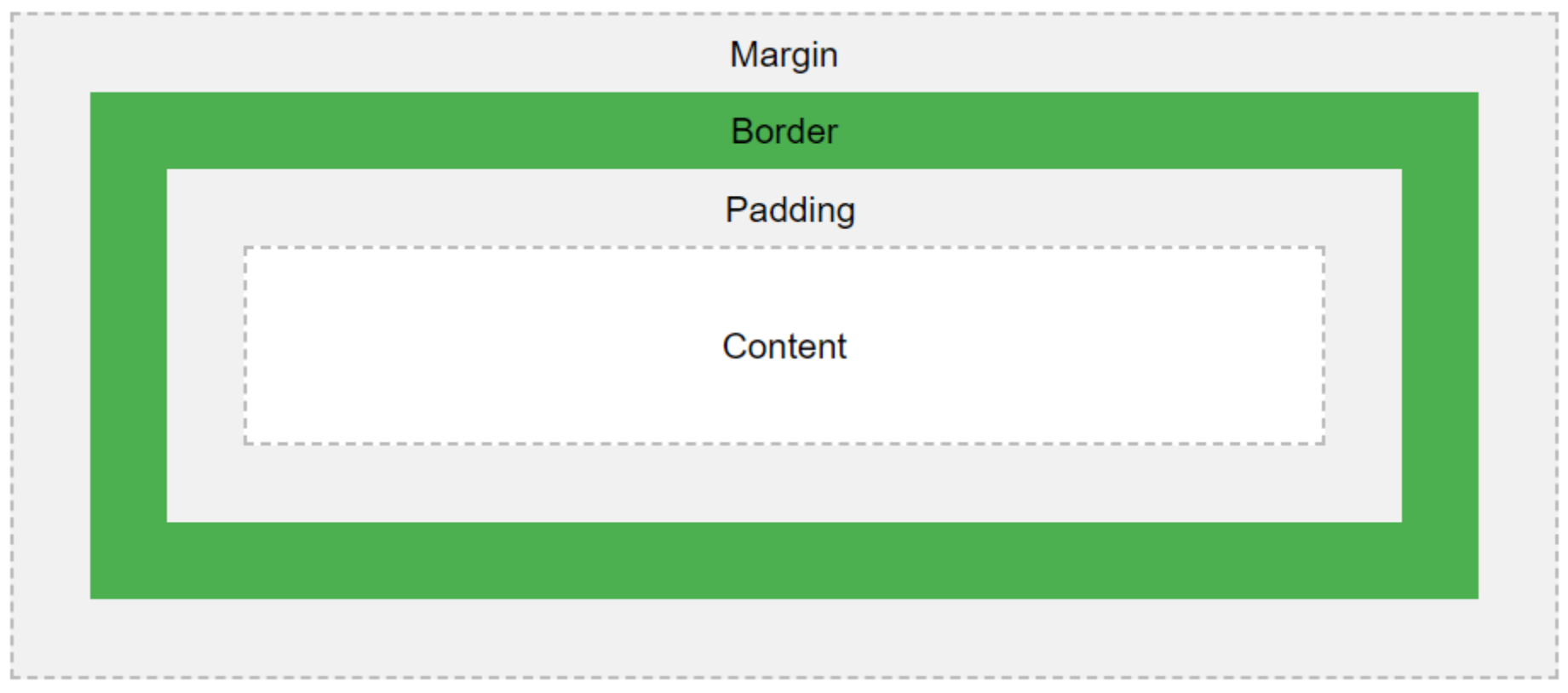
property	values
margin-top	padding-top: 100px;
padding-right	padding-right: 100px;
padding-bottom	padding-bottom: 100px;
padding-left	padding-left: 100px;
padding	padding: 25px 50px 75px 100px;[top,right,bottom,left] padding: 25px 50px 75px;[top,right-left,bottom] padding: 25px 50px;[top-bottom,right-left] padding: 25px;[all]
box-sizing	To keep the width at 300px, no matter the amount of padding, you can use the box-sizing property
	element is given a width of 300px. However, the actual rendered width of the element will be 350px (300px + 25px of left padding + 25px of right padding):

```
div {
  width: 300px;
  padding: 25px;
}
```

```
div {
  width: 300px;
  padding: 25px;
  box-sizing: border-box;
}
```

CSS box model

- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model
  - **Content** - The content of the box, where text and images appear
  - **Padding** - Clears an area around the content. The padding is transparent
  - **Border** - A border that goes around the padding and content
  - **Margin** - Clears an area outside the border. The margin is transparent



CSS box model

```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
```

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= **350px**

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS incline vs block level element

- An **inline element** does not start on a new line and only takes up as much width as necessary.
  This is an inline element inside a paragraph. Examples of inline elements:

  <a>
  <img>

  Block-level Elements

  A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

  The element is a block-level element.

  Examples of block-level elements:

  <h1> - <h6>
  <p>
  <form>
  <header>
  <footer>
  <section>

CSS Display /Visibility/overflow

- The display property is the most important CSS property for controlling layout.
- The display property specifies if/how an element is displayed.

- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

property	values
display	Display:block; [inline,block,none] The element will be hidden, and the page will be displayed as if the element is not there
Visibility	visibility:hidden; element will still take up the same space as before. The element will be hidden, but still affect the layout
overflow	Hidden,auto,visible,scroll