

WEB PAGE WITHOUT CSS AND WITH CSS

News

Local and National News

- [Home](#)
- [Archives](#)
- [About](#)

Local News

Fire fighters rescue man from building

(author, date)

This is the story text. This is the story text.

This is the story text. This is the story text.

New Library to be built

(author, date)

This is the story text. This is the story text.

This is the story text. This is the story text.

National News

Snow storm is making travel difficult

(author, date)

News

Local and National News

[Home](#) [Archives](#) [About](#)

Local News

Fire fighters rescue man from building

(author, date)

This is the story text. This is the story text.

This is the story text. This is the story text.

New Library to be built

(author, date)

This is the story text. This is the story text.

This is the story text. This is the story text.

National News

Snow storm is making travel difficult

(author, date)

This is the story text. This is the story text.

This is the story text. This is the story text.

Thousands are without power

(author, date)

This is the story text. This is the story text.

This is the story text. This is the story text.

Be a news reporter

If you see news happening - Send us a Text.

Footer Information

CSS – Introduction

- **CSS** is used to control the style of a web document in a simple and easy way. **CSS** is the acronym for "**Cascading Style Sheet**".

CSS saves time - can write CSS once and then reuse same sheet in multiple HTML pages.

Pages load faster - 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.

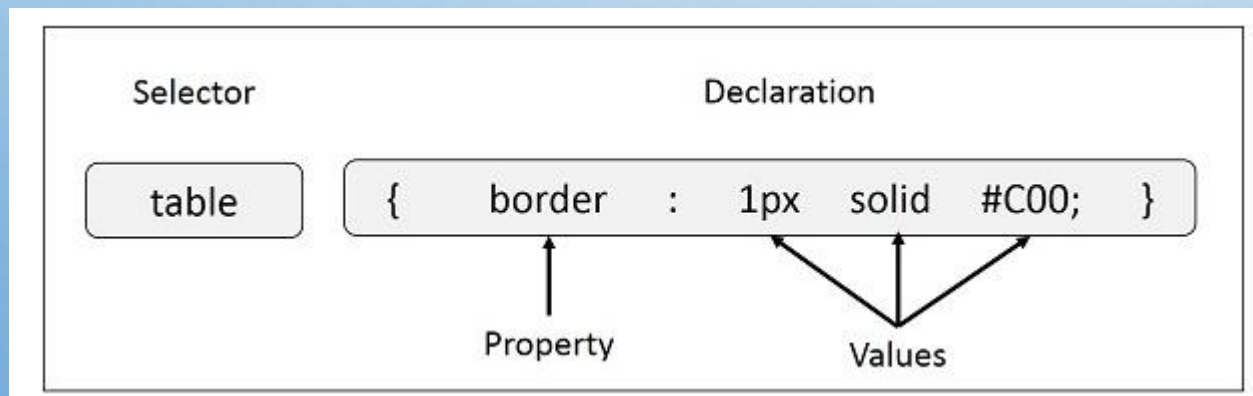
Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

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 – Syntax

- A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in the 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 }
```



CSS SELECTORS

- CSS SELECTORS ARE USED TO "FIND" (OR SELECT) THE HTML ELEMENTS YOU WANT TO STYLE.

WE CAN DIVIDE CSS SELECTORS INTO FIVE CATEGORIES:

- **SIMPLE SELECTORS** (SELECT ELEMENTS BASED ON NAME, ID, CLASS)
- COMBINATOR SELECTORS (SELECT ELEMENTS BASED ON A SPECIFIC RELATIONSHIP BETWEEN THEM)
- PSEUDO-CLASS SELECTORS (SELECT ELEMENTS BASED ON A CERTAIN STATE)
- PSEUDO-ELEMENTS SELECTORS (SELECT AND STYLE A PART OF AN ELEMENT)
- ATTRIBUTE SELECTORS (SELECT ELEMENTS BASED ON AN ATTRIBUTE OR ATTRIBUTE VALUE)

THE CSS ELEMENT SELECTOR

- `P {`
 `TEXT-ALIGN: CENTER;`
 `COLOR: RED;`
 `}`

THE CSS ID SELECTOR -> ID IS UNIQUE ELEMENT IN A WEB PAGE

- `#PARA1 {`
 `TEXT-ALIGN: CENTER;`
 `COLOR: RED;`
 `}`

THE CSS CLASS SELECTOR

- THE CLASS SELECTOR SELECTS HTML ELEMENTS WITH A SPECIFIC CLASS ATTRIBUTE.

`.CENTER {`

`TEXT-ALIGN: CENTER;`

`COLOR: RED;`

`}`

YOU CAN ALSO SPECIFY THAT ONLY SPECIFIC HTML ELEMENTS SHOULD BE AFFECTED BY A CLASS.

`P.CENTER {`

`TEXT-ALIGN: CENTER;`

`COLOR: RED;`

`}`

Selector – Types

- **Type Selector** – Basic and default which discussed earlier.

- **Universal Selector** – Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type.

E.g. `* { color: #000000; }`

Descendant Selectors – want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to `` element only when it lies inside `` tag.

E.g. `ul em { color: #000000; }`

Class Selectors - can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule. The following rule renders the content in black for only `<h1>` elements with class attribute set to *black*.

E.g. `h1.black { color: #000000; }`

CSS – Class Selector

- all HTML elements with class="center" will be red
- and center-aligned:

Sample Program

```
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head> <body>
<h1 class="center">Red and center-aligned
heading</h1>
<p class="center">This paragraph will be red and center-
aligned.</p>
</body> </html>
```

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

Output

Red and center-aligned heading

Red and center-aligned paragraph.

CSS – Class Selector

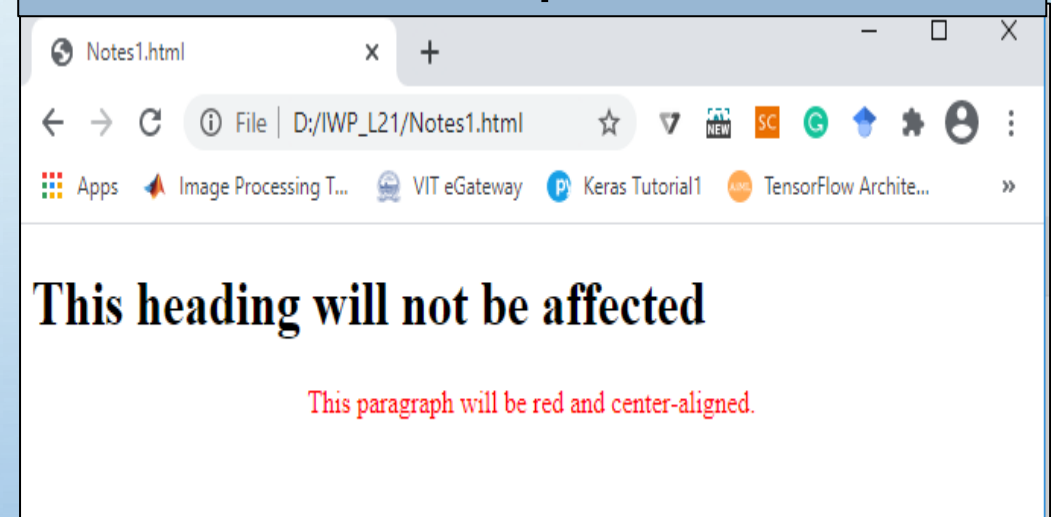
- Can also specify the class for specific HTML elements.

Sample Program

```
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
</style>
</head> <body>
<h1 class="center">This heading will not be
affected</h1>
<p class="center">This paragraph will be red and center-
aligned.</p>
</body> </html>
```

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

Output



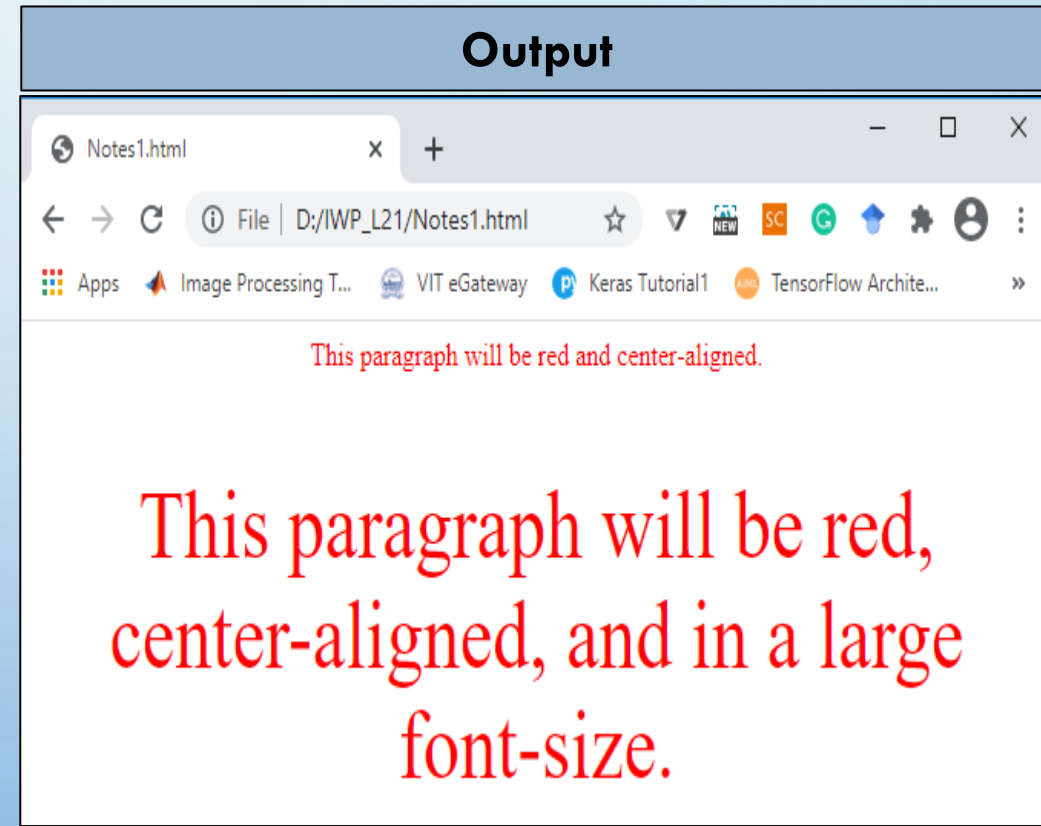
CSS – Class Selector

Can combine multiple class selectors for a single element.

Sample Program

```
<style>
p.center {
  text-align: center;
  color: red; }
p.large {
  font-size: 300%; }
</style>
<body>
<p class="center">This paragraph will be red and center-
aligned.</p>
<p class="center large">This paragraph will be red, center-
aligned, and in a large font-size.</p>
</body>
```

Output



CSS – ELEMENT, CLASS AND ID SELECTOR

```
< index.html > h1
1 <link rel="stylesheet" href="style.css">
2 <h1>Heading</h1>
3 <h1>How are you</h1>
```

```
# style.css > h1
1 h1{
2   color: red;
3 }
```

```
index.html > ...
1 <link rel="stylesheet" href="style.css">
2 <h1 class="one">Heading</h1>
3 <h1 class="two">How are you</h1>
4
```

```
# style.css > .two
1 .one{
2   color: red;
3 }
4
5 .two{
6   color: blue;
7 }
```

Heading

How are you

```
< index.html > h1#two
1 <link rel="stylesheet" href="style.css">
2 <h1 class="one">Heading</h1>
3 <h1 id="two">How are you</h1>
4
```

```
# style.css > #two
1 .one{
2   color: red;
3 }
4
5 #two{
6   color: blue;
7 }
```

CSS – Universal Selector

- The universal selector (*) selects all HTML elements on the page.

```
* {  
  text-align: center;  
  color: blue;  
}
```

```
<style>  
* {  
  text-align: center;  
  color: blue;  
}  
</style>  
</head>  
<body>  
<h1>Hello world!</h1>  
<p>Every element on the page will be affected by the  
style.</p>  
<p id="para1">Me too!</p>  
<p>And me!</p>  
</body>
```

Hello world!

Every element on the page will be affected by the style.

Me too!

And me!

CSS – Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions.
- Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

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



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


DESCENDANT SELECTORS

```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>Descendant Selector</h2>
<p>The descendant selector matches all elements that
are descendants of a specified element.</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
</body>
</html>
```

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

CSS – Insertion Models

There are three ways of inserting a style sheet:

1. **Inline CSS** – An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
2. **Internal CSS** – An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the `<style>` element, inside the head section.
3. **External CSS** – An external style sheet can be written in any text editor, and must be saved with a .css extension. The external .css file should not contain any HTML tags. Each HTML page must include a reference to the external style sheet file inside the `<link>` element, inside the head section.

CSS – Insertion Models

In-Line Model

```
<html>
<body>

<h1 style="color:blue;text-align:center;">
This is a heading</h1>
<p style="color:red;">This is a
paragraph.</p>

</body>
</html>
```

Internal Model

```
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

External Model

```
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body> </html>
```

Mystyle.css

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

h1 {
  color: navy;
  margin-left: 20px; }
```

CSS – Measurements

- CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units. These values are needed while specifying various measurements in the style rules like **border = "1px solid red"**.

Unit	Description	Example
%	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;}
in	Defines a measurement in inches.	p {word-spacing: .15in;}
mm	Defines a measurement in millimeters.	p {word-spacing: 15mm;}
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 – Insertion Models

- There are three ways of inserting a style sheet:

- 1. Inline CSS** – An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
- 2. Internal CSS** – An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the `<style>` element, inside the head section.
- 3. External CSS** – An external style sheet can be written in any text editor, and must be saved with a .css extension. The external .css file should not contain any HTML tags. Each HTML page must include a reference to the external style sheet file inside the `<link>` element, inside the head section.

CSS – Insertion Models

In-Line Model

```
<html>
<body>

<h1 style="color:blue;text-align:center;">
This is a heading</h1>
<p style="color:red;">This is a
paragraph.</p>

</body>
</html>
```

Internal Model

```
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

External Model

```
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body> </html>
```

Mystyle.css

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

h1 {
  color: navy;
  margin-left: 20px; }
```

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and **much more!**

IN-LINE

This is a heading

This is a paragraph.

INTERNAL

This is a heading

This is a paragraph.

EXTERNAL

This is a heading

This is a paragraph.

Apple

Orange

Mango

```
<h1 style="background-color:  red;">Apple</h1>  
<h1 style="background-color:  orange;">Orange</h1>  
<h1 style="background-color:  yellow;">Mango</h1>
```


Create a **div** tag with background color red (**internal css**)
Inside div tag create a **h1** tag with color white(**inline css**)
Set width(300px) and height(100px) to div tag(**internal css**)
Create **P** tag after h1 tag and set its color to blue(**inline css**)

```
<style>
  div{
    background-color: red;
    height: 100px;
  }
</style>

<div>

</div>
```



```
<style>
  div{
    background-color: red;
    height: 100px;
    width: 300px;
  }
</style>

<div>
  <h1 style="color: white;">Error Makes Clev
  <p style="color: blue;">Pudukkottai</p>
</div>
```

HTML COLORS

- HTML COLORS ARE SPECIFIED WITH PREDEFINED COLOR NAMES, OR WITH RGB, HEX, HSL, RGBA, OR HSLA VALUES.
- COLOR NAMES
 - TOMATO
 - ORANGE
 - LIGHTGRAY ETC..
 - `<H1 STYLE="BACKGROUND-COLOR:TOMATO;">TOMATO</H1>`

- **BACKGROUND COLOR**

- YOU CAN SET THE BACKGROUND COLOR FOR HTML ELEMENTS



- **TEXT COLOR**

- YOU CAN SET THE COLOR OF TEXT `<H1 STYLE="COLOR:TOMATO;">HELLO
WORLD</H1>`

- **BORDER COLOR**

- `<H1 STYLE="BORDER:2PX SOLID TOMATO;">HELLO WORLD</H1>`

- **COLOR VALUES**

- IN HTML, COLORS CAN ALSO BE SPECIFIED USING RGB VALUES, HEX VALUES, HSL VALUES, RGBA VALUES, AND HSLA VALUES.

- HSL -> HUE, SATURATION & LIGHTNESS

- ALPHA CHANNEL TO THE COLOR (ADDS TRANSPARENCY).

- `<H1 STYLE="BACKGROUND-COLOR:RGB(255, 99, 71);">...</H1>`

- `<H1 STYLE="BACKGROUND-COLOR:#FF6347;">...</H1>`

- `<H1 STYLE="BACKGROUND-COLOR:HSL(9, 100%, 64%);">...</H1>`

- `<H1 STYLE="BACKGROUND-COLOR:RGBA(255, 99, 71, 0.5);">...</H1>`

- `<H1 STYLE="BACKGROUND-COLOR:HSLA(9, 100%, 64%, 0.5);">...</H1>`

`rgb(255, 99, 71)`

RED



255

GREEN



99

BLUE



71

`hsl(0, 100%, 50%)`

HUE



0

SATURATION



100%

LIGHTNESS



50%

`rgba(255, 99, 71, 0.5)`

RED



255

GREEN



99

BLUE



71

ALPHA



0.5

- RGB(RED, GREEN, BLUE)

- HEX: #RRGGBB

- TO DISPLAY BLACK, SET ALL COLOR PARAMETERS TO 00, LIKE THIS: #000000.

- TO DISPLAY WHITE, SET ALL COLOR PARAMETERS TO FF, LIKE THIS: #FFFFFF.

- HSL COLOR VALUES

- HUE IS A DEGREE ON THE COLOR WHEEL FROM 0 TO 360. 0 IS RED, 120 IS GREEN, AND 240 IS BLUE.

- SATURATION IS A PERCENTAGE VALUE. 0% MEANS A SHADE OF GRAY, AND 100% IS THE FULL COLOR.

- LIGHTNESS IS ALSO A PERCENTAGE VALUE. 0% IS BLACK, AND 100% IS WHITE.

CSS – Background

- The CSS background properties are used to add background effects for elements.
- 1. The **background-color** property specifies the background color of an element. Values are valid color name like red, green, etc. or valid hexadecimal code.
- 2. The **background-image** property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element. Value is `url("image file name")`.
- 3. By default, the **background-repeat** property repeats an image both horizontally and vertically. Values are ***repeat-x***, ***no-repeat***, ***repeat-y***.
- 4. The **background-position** property decides the location of background image like ***top***, ***bottom***, ***left*** or ***right***. Value is right top
- 5. The **background-attachment** property specifies whether the background image should scroll or be fixed. Value is ***fixed*** or ***scroll***.

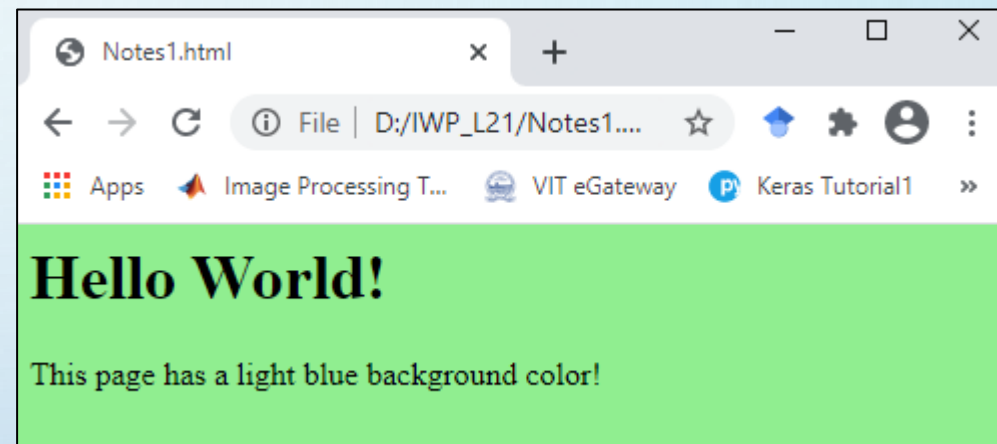
CSS – Background Samples

```
<html><head>
<style>
body {
  background-color: lightgreen;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has a light blue background color!</p>

</body>
</html>
```

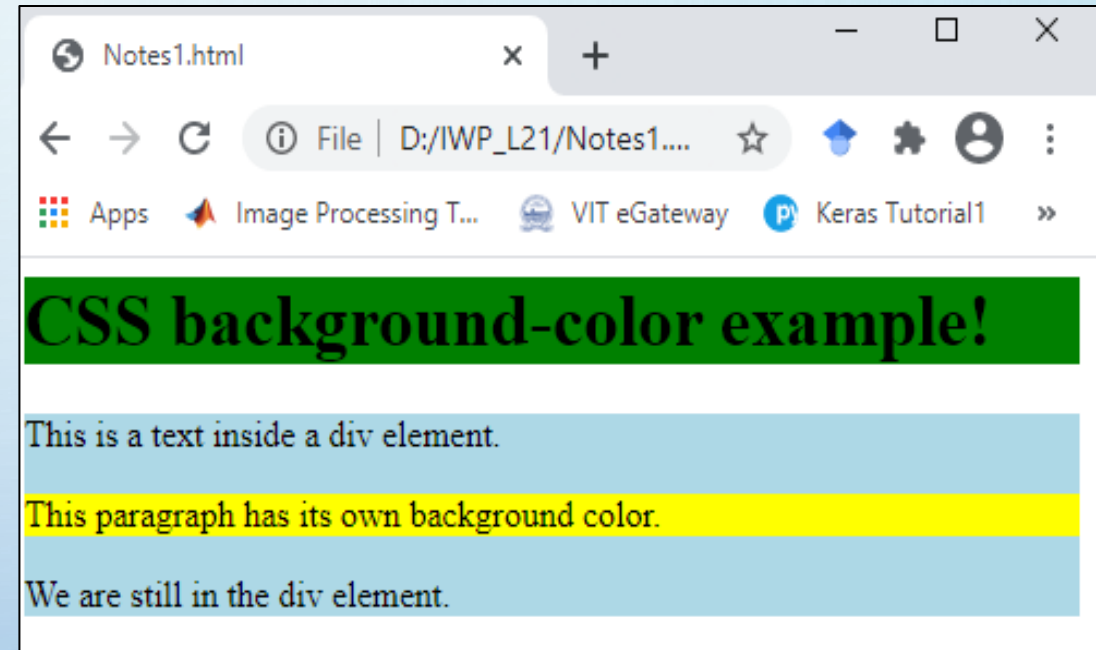


CSS – Background Samples

```
<style>
h1 {
  background-color: green; }

div {
  background-color: lightblue; }

p {
  background-color: yellow; }
</style> </head>
<body>
<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>
</body>
```



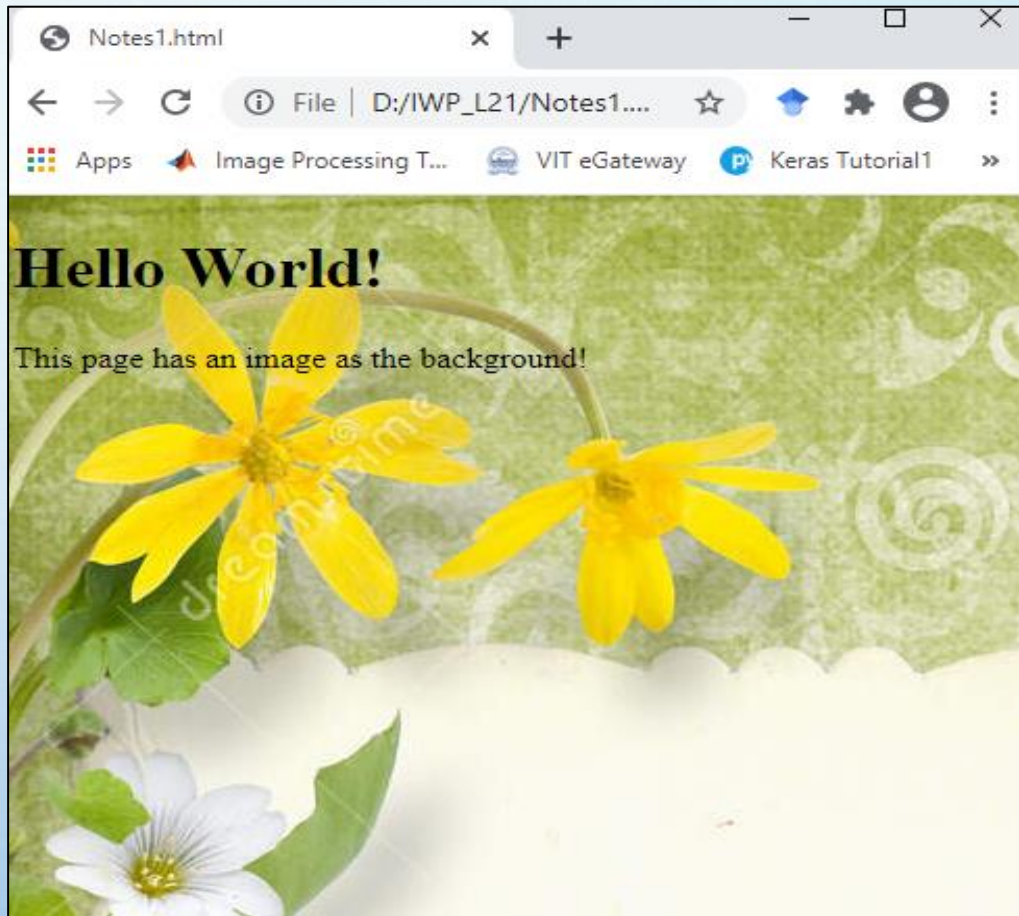
CSS – Background Samples

```
<head>
<style>
body {
  background-image: url("background1.jpg");
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the
background!</p>

</body>
```



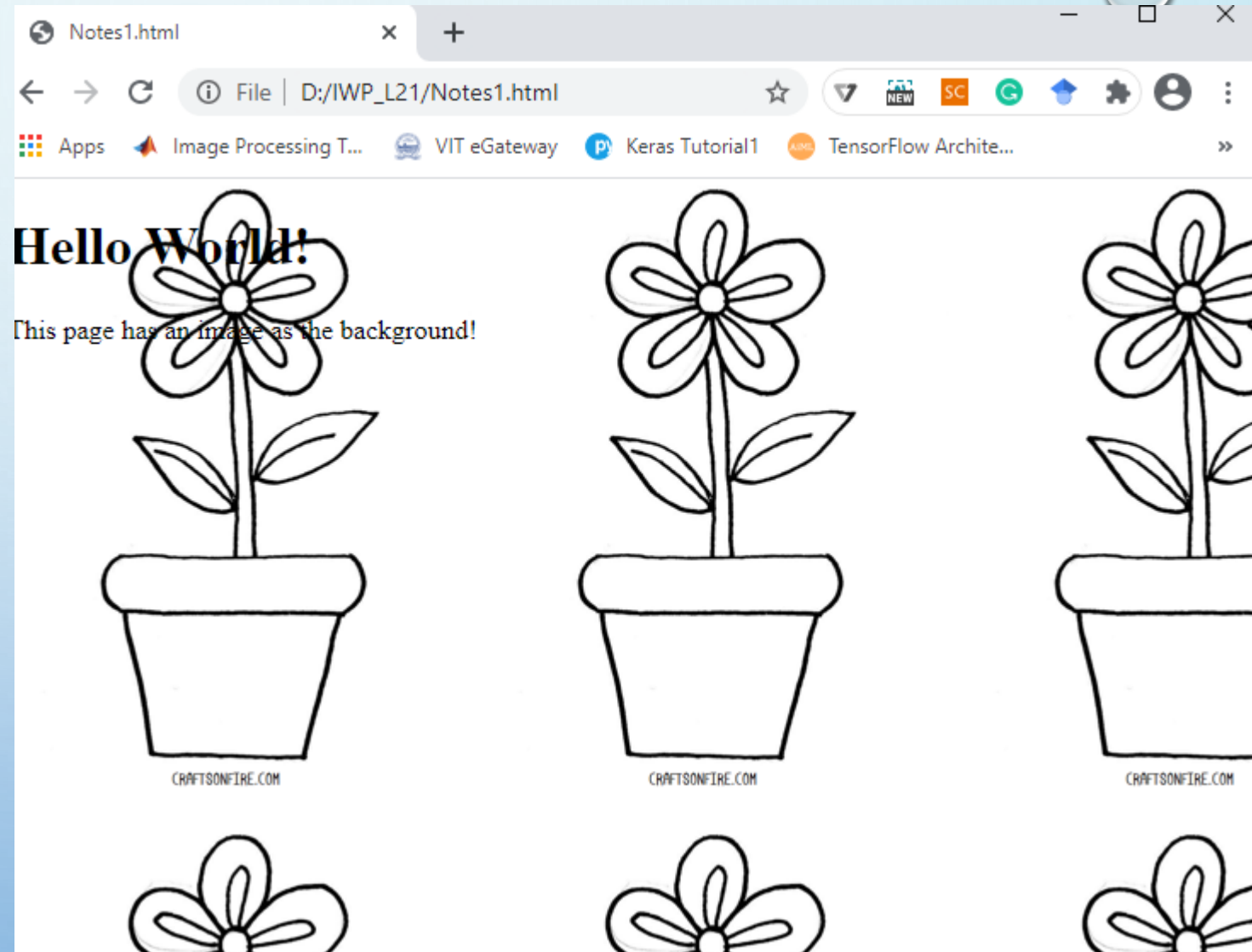
CSS – Background Samples

```
<head>
<style>
body {
  background-image: url("flower_small.png");
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the
background!</p>

</body>
```



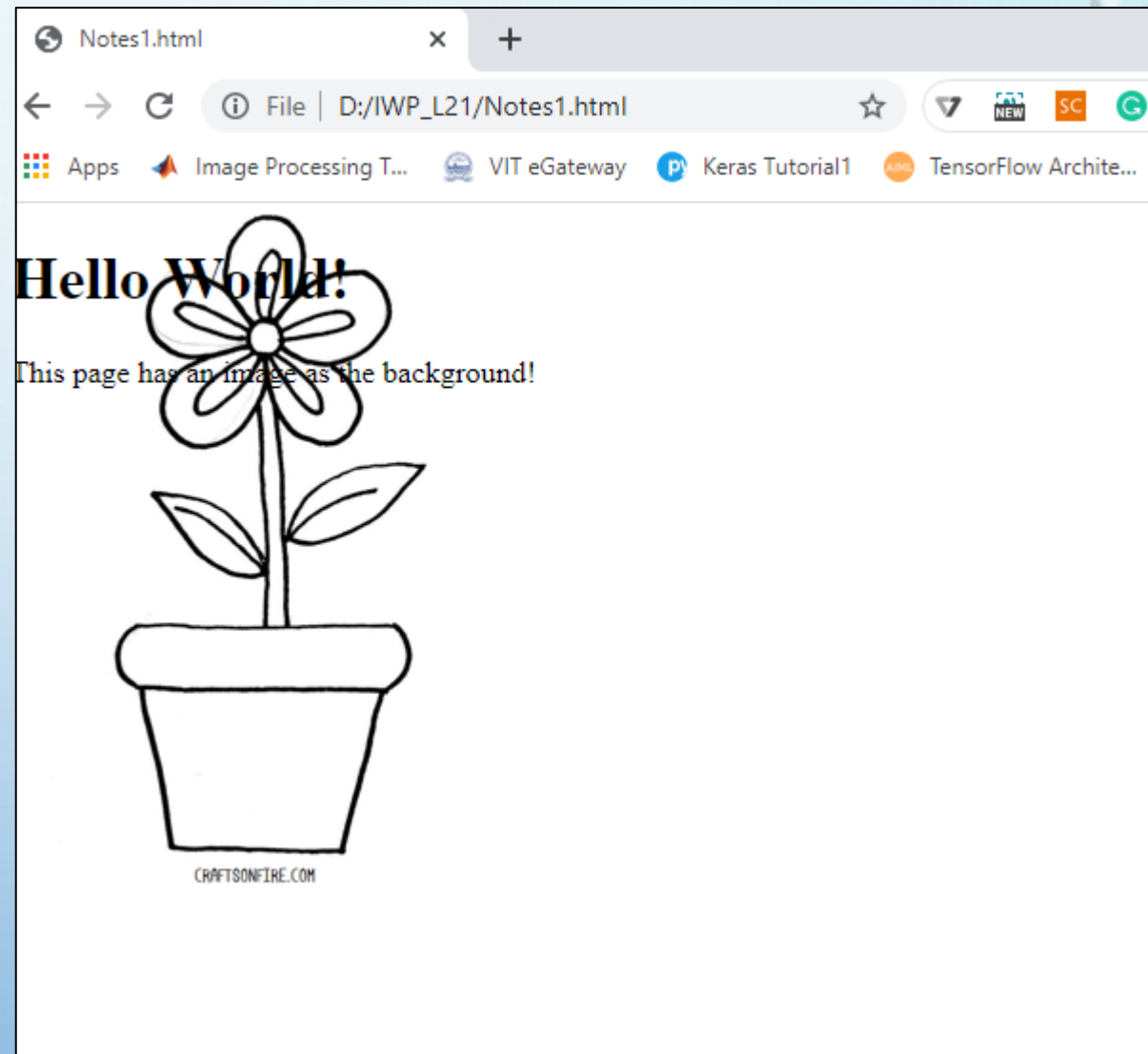
CSS – Background Samples

```
<head>
<style>
body {
  background-image: url("flower_small.png");
  background-repeat: no-repeat;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the
background!</p>

</body>
```



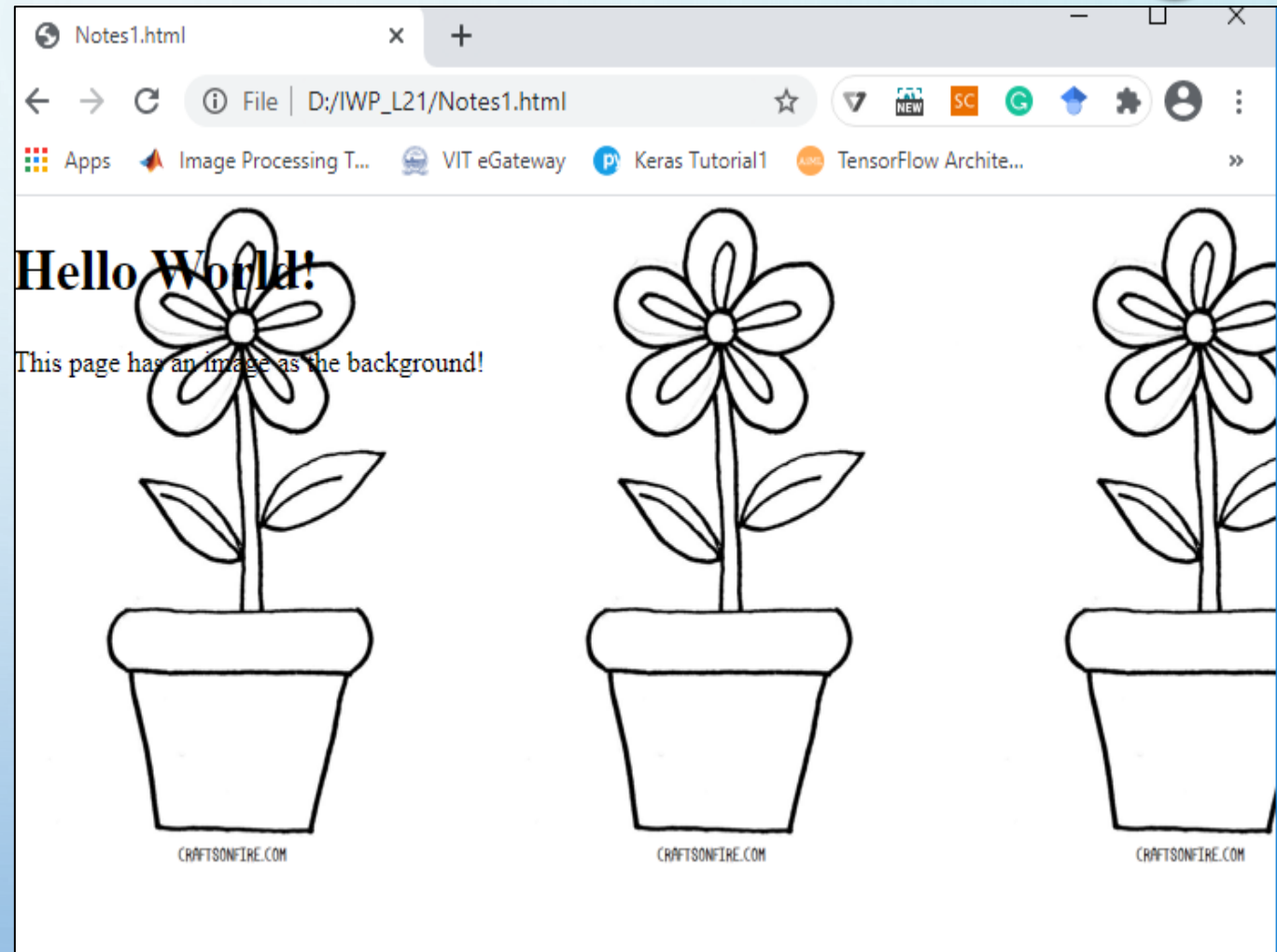
CSS – Background Samples

```
<head>
<style>
body {
  background-image: url("flower_small.png");
  background-repeat: repeat-x;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the
background!</p>

</body>
```



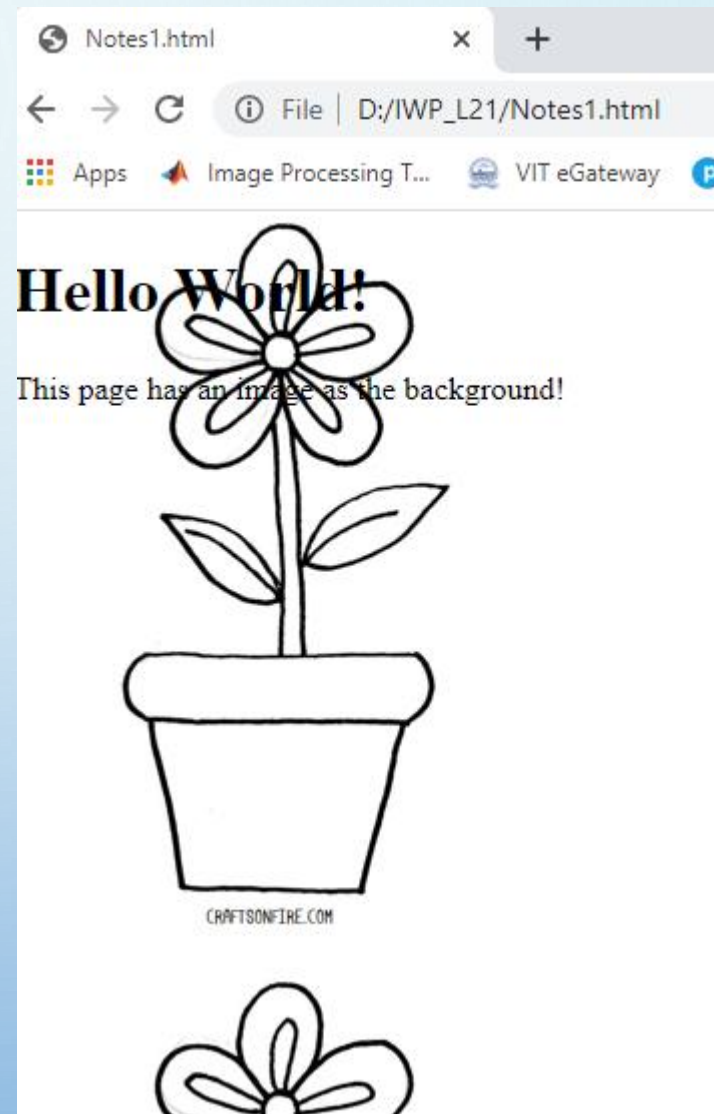
CSS – Background Samples

```
<head>
<style>
body {
  background-image: url("flower_small.png");
  background-repeat: repeat-y;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the
background!</p>

</body>
```



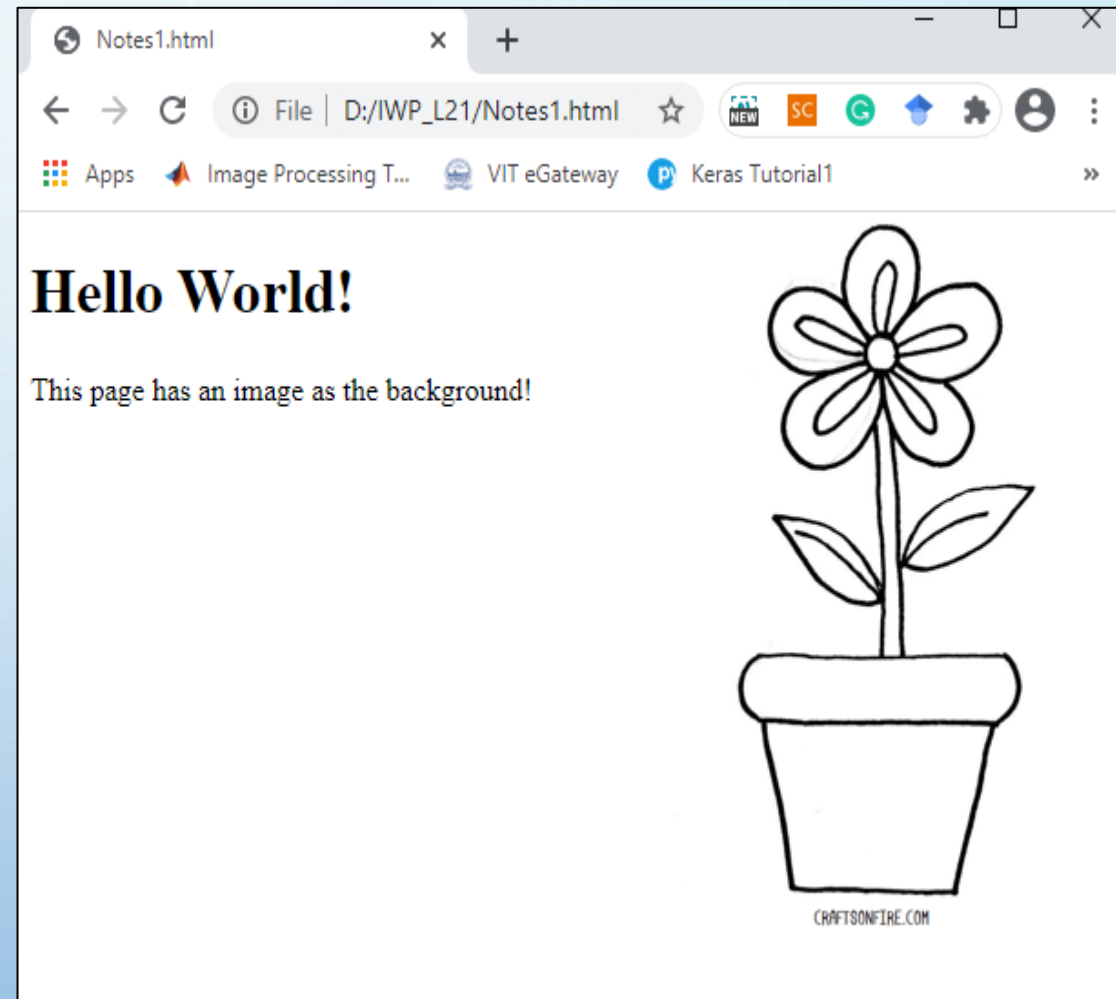
CSS – Background Samples

```
<head>
<style>
body {
  background-image: url("flower_small.png");
  background-repeat: no-repeat;
  background-position: right top;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the
background!</p>

</body>
```

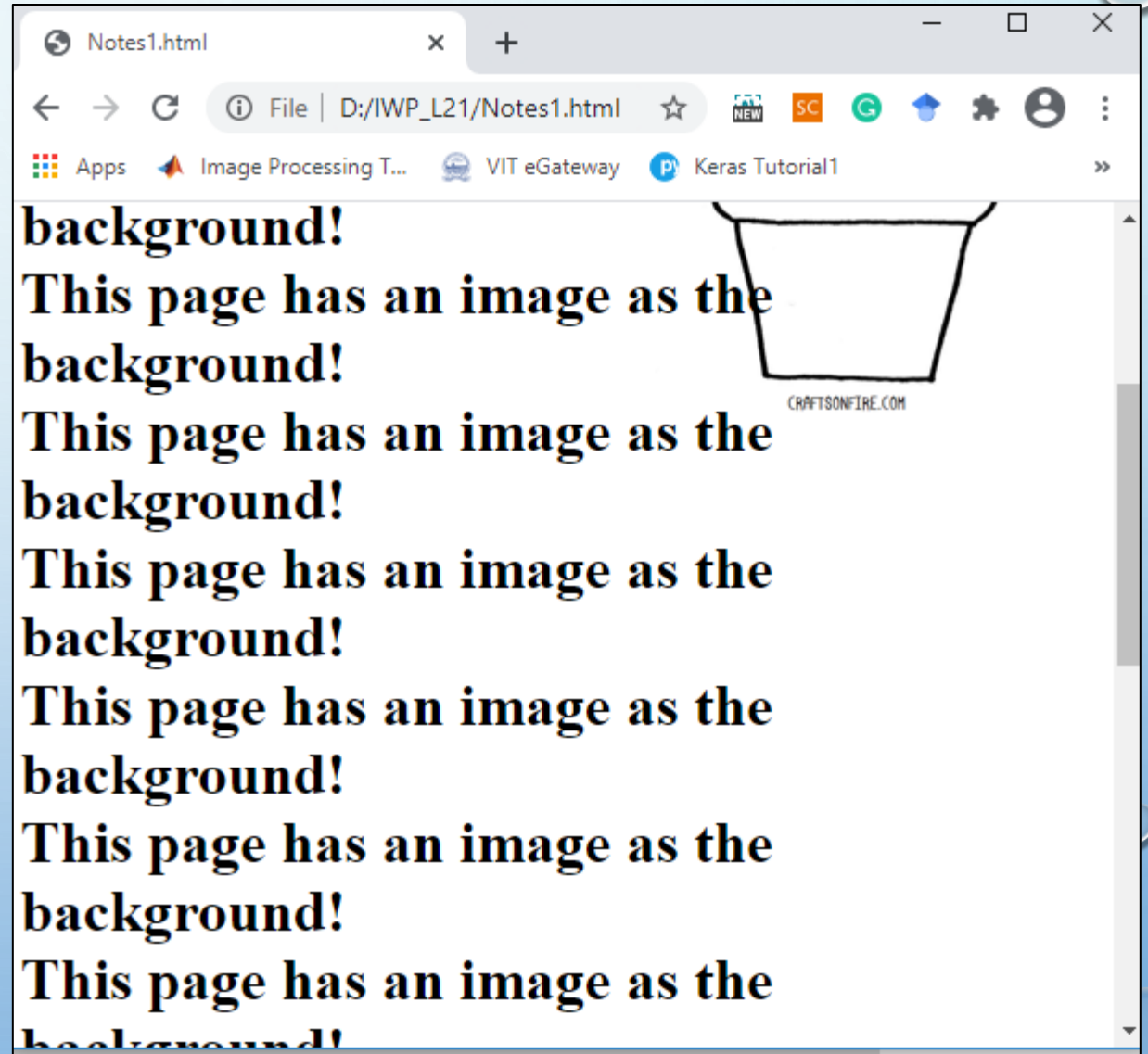


CSS – Background Samples

```
<head>
<style>
body {
  background-image: url("flower_small.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: scroll;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<h1>
<p>This page has an image as the
background!<br>
This page has an image as the
background!<br>
.....</p>

</body>
```



CSS – Borders

- The CSS border properties allow to specify the style, width, and color of an element's border. Properties are:

- **dotted** - Defines a dotted border
- **dashed** - Defines a dashed border
- **solid** - Defines a solid border
- **double** - Defines a double border
- **groove** - Defines a 3D grooved border. The effect depends on the border-color value
- **ridge** - Defines a 3D ridged border. The effect depends on the border-color value
- **inset** - Defines a 3D inset border. The effect depends on the border-color value
- **outset** - Defines a 3D outset border. The effect depends on the border-color value

CSS – Borders

```
<html>
<head>
<style>
p.dotted {border-style: dotted;
border-color: red;}
p.dashed {border-style: dashed;
border-color: green;}
p.solid {border-style: solid;
border color: lightblue;}
</style>
</head>
<body>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
</body>
</html>
```



CSS – Text

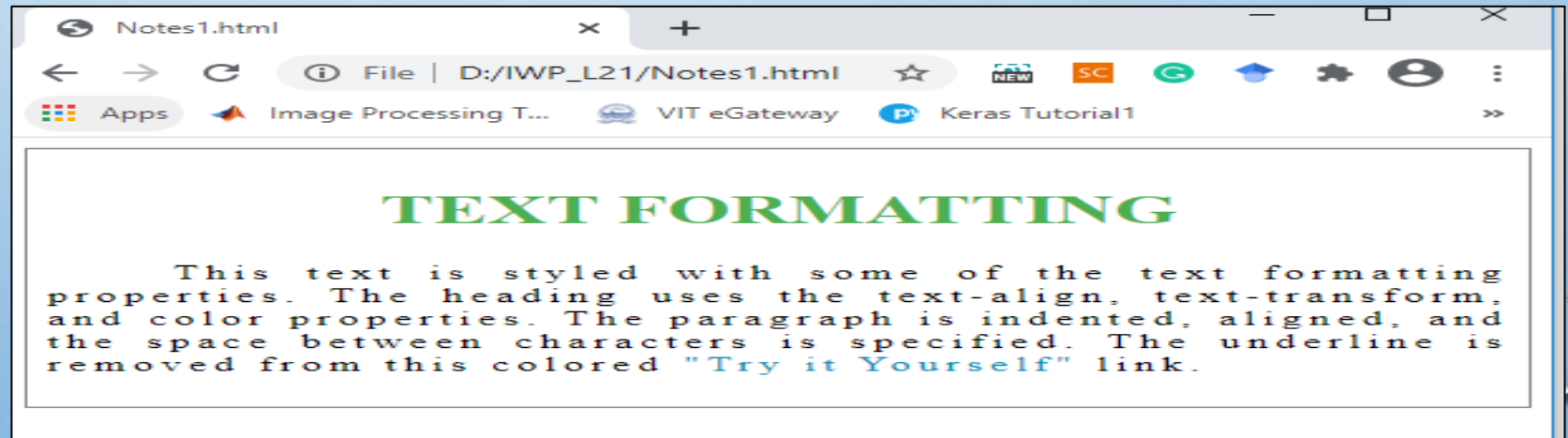
- CSS has a lot of properties for formatting text.

1. The **color** property is used to set the color of the text. The color is specified by a valid color name or hexadecimal code.
2. The **text-align** property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified.
3. The **text-indent** property is used to specify the indentation of the first line of a text.
4. The **letter-spacing** property is used to specify the space between the characters in a text.
5. The **line-height** property is used to specify the space between lines .
6. The CSS **text-overflow** property specifies how overflowed content that is not displayed should be signaled to the user.
7. The CSS **word-wrap** property allows long words to be able to be broken and wrap onto the next line. If a word is too long to fit within an area, it expands outside. Values are keep-all and break-all.

CSS – Text

```
<head>
<style>
div {
  border: 1px solid gray;
  padding: 8px; }
h1 {
  text-align: center;
  text-transform: uppercase;
  color: #4CAF50; }
p {
  text-indent: 50px;
  text-align: justify;
  letter-spacing: 3px; }
a {
  text-decoration: none;
  color: #008CBA; }
</style>
</head>
```

```
<body> <div>
  <h1>text formatting</h1>
  <p>This text is styled with some of the text formatting properties. The heading
uses the text-align, text-transform, and color properties.
  The paragraph is indented, aligned, and the space between characters is
specified. The underline is removed from this colored
  <a target="_blank" href="tryit.asp?filename=trycss_text">"Try it Yourself"</a>
link.</p>
</div> </body>
</html>
```



CSS – TEXT EFFECTS

```
<html>
<head>
<style>
div {
  width: 150px;
  border: 1px solid #000000;
}

div.a {
  word-wrap: normal;
}

div.b {
  word-wrap: break-word;
}
</style>
</head>
<body>

<h1>The word-wrap Property</h1>

<h2>word-wrap: normal (default):</h2>
<div class="a"> This div contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will not break and wrap to the
next line.</div>

<h2>word-wrap: break-word:</h2>
<div class="b"> This div contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next
line.</div>

</body>
</html>
```

The word-wrap Property

word-wrap: normal (default):

This div contains a very long word:
thisisaveryveryveryveryveryverylongword.
The long word will not break and wrap to the next line.

word-wrap: break-word:

This div contains a very long word:
thisisaveryveryveryveryveryverylongword.
The long word will break and wrap to the next line.

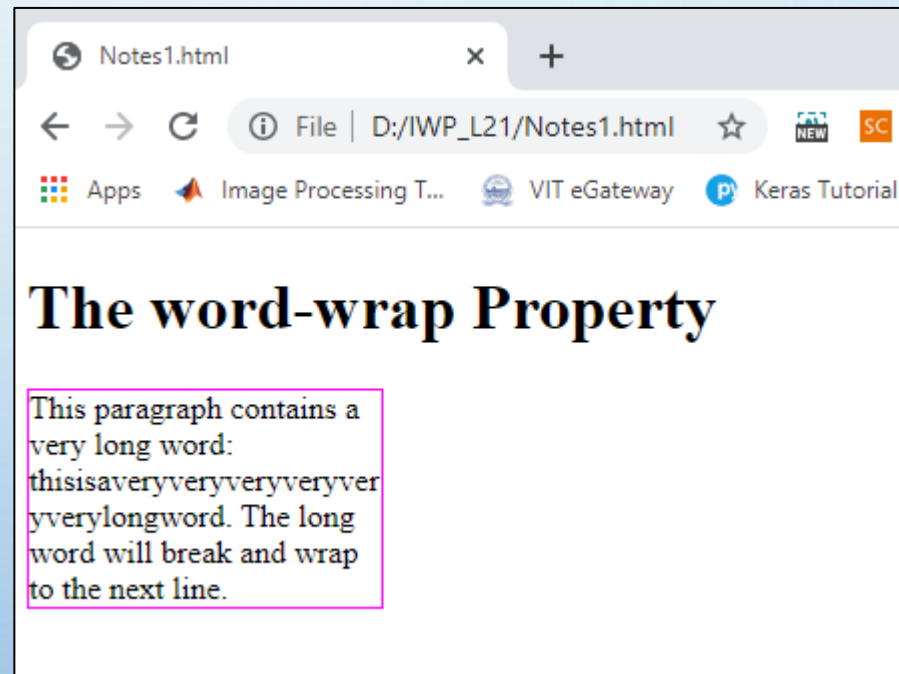
CSS – Text Effects

```
<head>
<style>
p.test {
  width: 11em;
  border: 1px solid #FF00FF;
  word-wrap: break-word;
}
</style>
</head>
<body>

<h1>The word-wrap Property</h1>

<p class="test"> This paragraph contains a
very long word:
thisisaveryveryveryveryveryverylongword. The
long word will break and wrap to the next
line.</p>

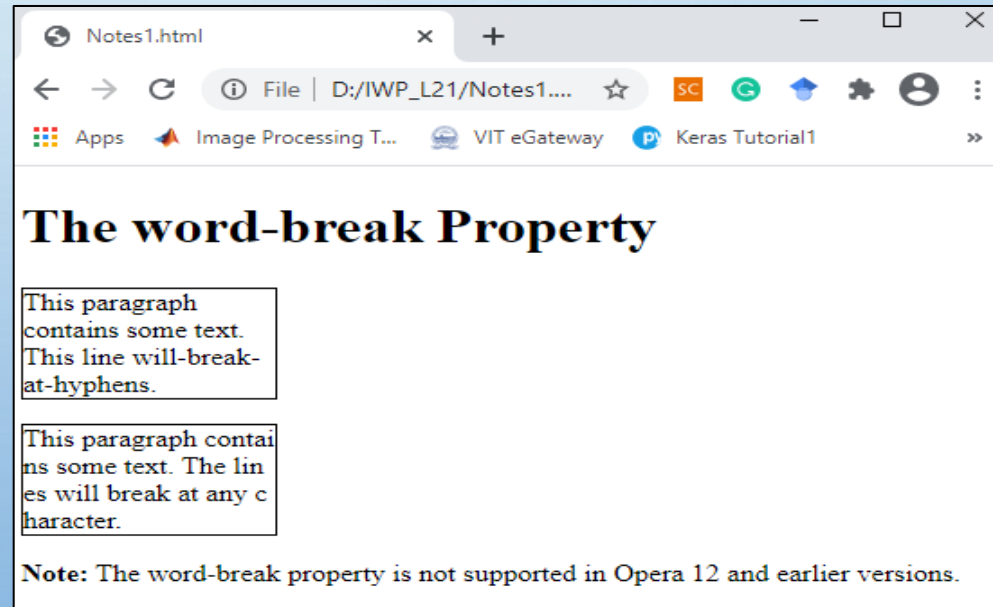
</body>
```



CSS – Text Effects

```
<head>
<style>
p.test1 {
  width: 140px;
  border: 1px solid #000000;
  word-break: keep-all; }
p.test2 {
  width: 140px;
  border: 1px solid #000000;
  word-break: break-all; }
</style>
</head>
<body>
```

```
<h1>The word-break Property</h1>
<p class="test1">This paragraph contains some text. This line will-break-
at-hyphens.</p>
<p class="test2">This paragraph contains some text. The lines will break at
any character.</p>
<p><b>Note:</b> The word-break property is not supported in Opera
12 and earlier versions.</p>
</body> </html>
```



CSS – Pseudo Class

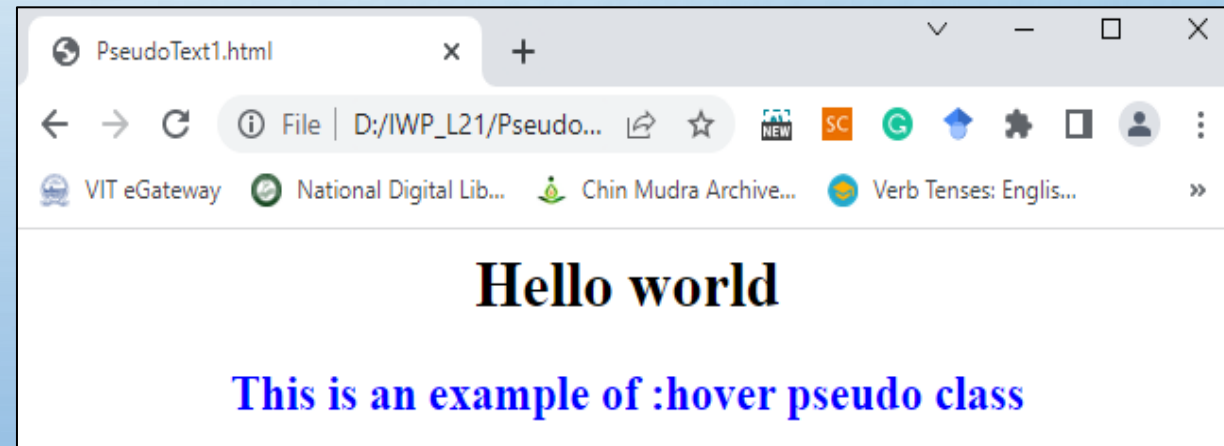
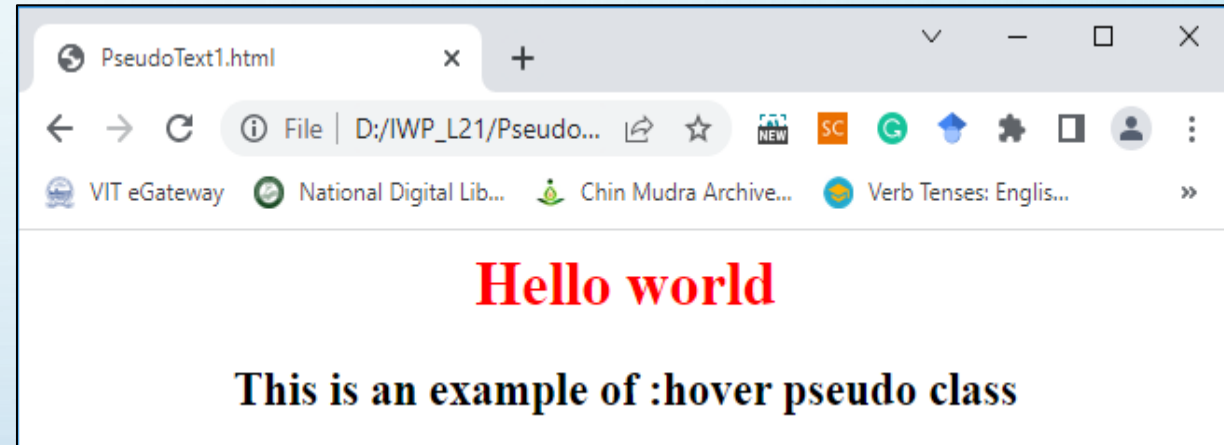
- A **pseudo-class** can be defined as a keyword which is combined to a selector that defines the **special state** of the selected elements.
- It is added to the selector for adding an effect to the existing elements based on their states.
- The names of the pseudo-class are not case-sensitive.

pseudo-class	Description
:active	It is used to add style to an active element.
:hover	It adds special effects to an element when the user moves the mouse pointer over the element.
:link	It adds style to the unvisited link.
:visited	It adds style to a visited link.
:lang	It is used to define a language to use in a specified element.
:focus	It selects the element which is focused by the user currently.

CSS – Pseudo Class

```
<html>
<head>
  <style>
    body{
      text-align:center;
    }
    h1:hover{
      color:red;
    }
    h2:hover{
      color:blue;
    }
  </style>
</head>

<body>
  <h1>Hello world </h1>
  <h2>This is an example of :hover pseudo class</h2>
</body>
</html>
```



CSS – Pseudo Class

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {
  color: red;
}
a:visited {
  color: green;
}
a:hover {
  color: hotpink;
}
a:active {
  color: blue;
}
</style>
</head>
<body>
```

```
<body>
<h2>Styling a link depending on state</h2>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in
order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be
effective.</p>

</body></html>
```

CSS – Pseudo Class

Normal Link



The screenshot shows a web browser window with the title "PseudoText2.html". The address bar displays "D:/IWP_L21/PseudoText2.html". The browser's toolbar includes navigation buttons (back, forward, refresh) and a search bar. Below the toolbar, there are several search engine shortcuts: VIT eGateway, National Digital Lib..., Chin Mudra Archive..., Verb Tenses: Englis..., and IAPR TC-. The main content area displays the heading "Styling a link depending on state" followed by the text "This is a link" in red, underlined font. Below this, two notes are provided: "Note: a: hover MUST come after a: link and a: visited in the CSS definition in order to be effective." and "Note: a: active MUST come after a: hover in the CSS definition in order to be effective."

Hover



The screenshot shows a web browser window with the title "PseudoText2.html". The address bar displays "D:/IWP_L21/PseudoText2.html". The browser's toolbar includes navigation buttons (back, forward, refresh) and a search bar. Below the toolbar, there are several search engine shortcuts: VIT eGateway, National Digital Lib..., Chin Mudra Archive..., Verb Tenses: Englis..., and IAPR TC-. The main content area displays the heading "Styling a link depending on state" followed by the text "This is a link" in pink, underlined font. Below this, two notes are provided: "Note: a: hover MUST come after a: link and a: visited in the CSS definition in order to be effective." and "Note: a: active MUST come after a: hover in the CSS definition in order to be effective."

Visited



The screenshot shows a web browser window with the title "PseudoText2.html". The address bar displays "D:/IWP_L21/PseudoText2.html". The browser's toolbar includes navigation buttons (back, forward, refresh) and a search bar. Below the toolbar, there are several search engine shortcuts: VIT eGateway, National Digital Lib..., Chin Mudra Archive..., Verb Tenses: Englis..., and IAPR TC-. The main content area displays the heading "Styling a link depending on state" followed by the text "This is a link" in green, underlined font. Below this, two notes are provided: "Note: a: hover MUST come after a: link and a: visited in the CSS definition in order to be effective." and "Note: a: active MUST come after a: hover in the CSS definition in order to be effective."

CSS – Box Model

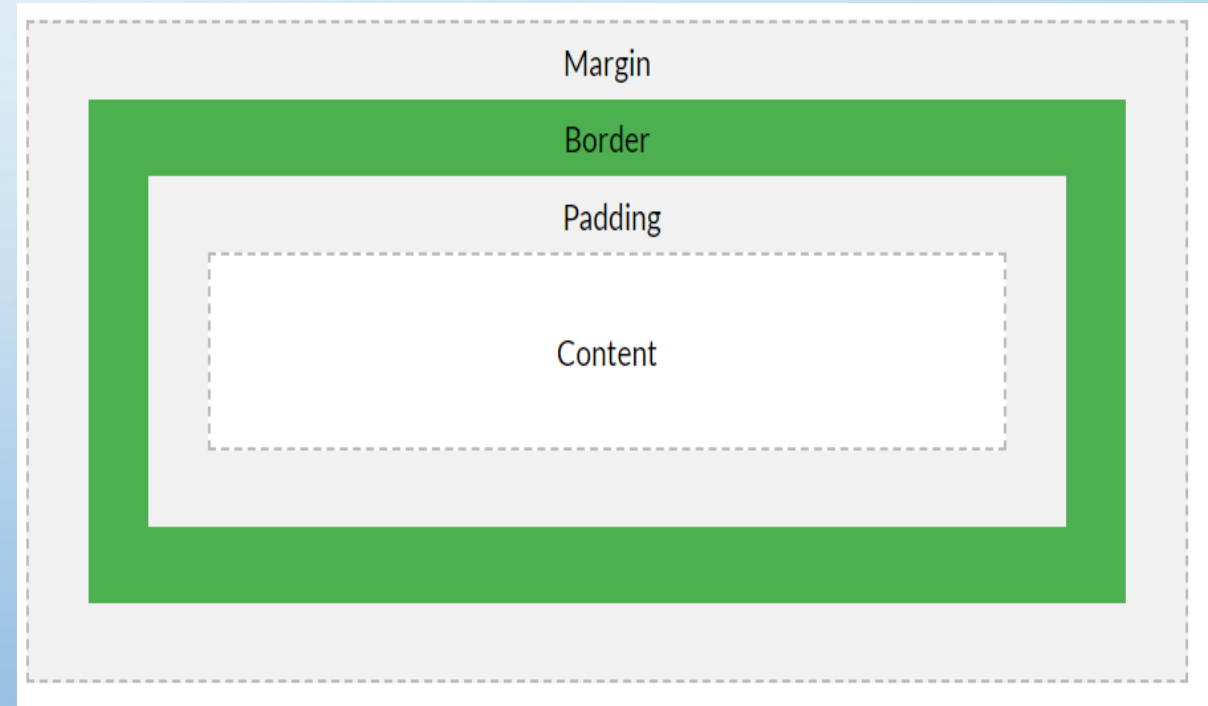
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The box model allows us to add a border around elements, and to define space between elements. This can be created using div tag. The image below illustrates 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 Example

```
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>

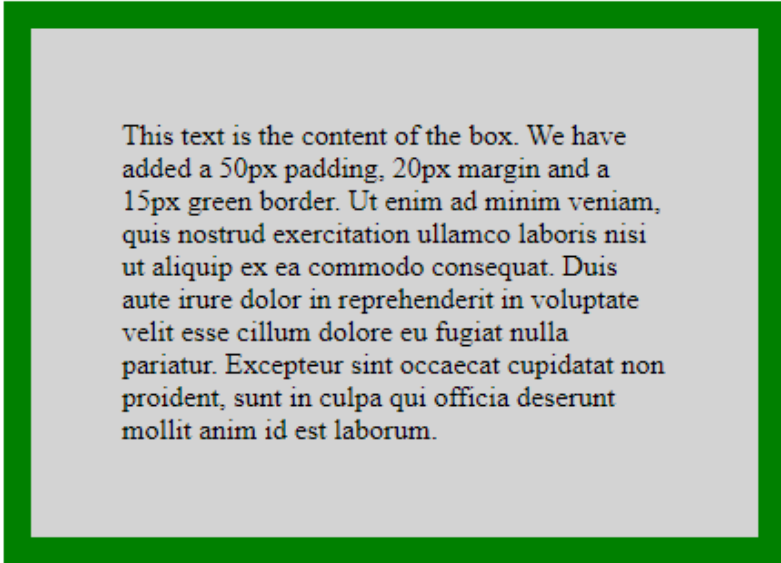
<p>The CSS box model is .....</p>

<div>This text is the content of the box..... </div>

</body>
```

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.



This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CSS – Box Width Height Calculation

- If a box is set with the width 300px then the actual occupying width is

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

The total width of an element should be calculated like this:

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

The total height of an element should be calculated like this:

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

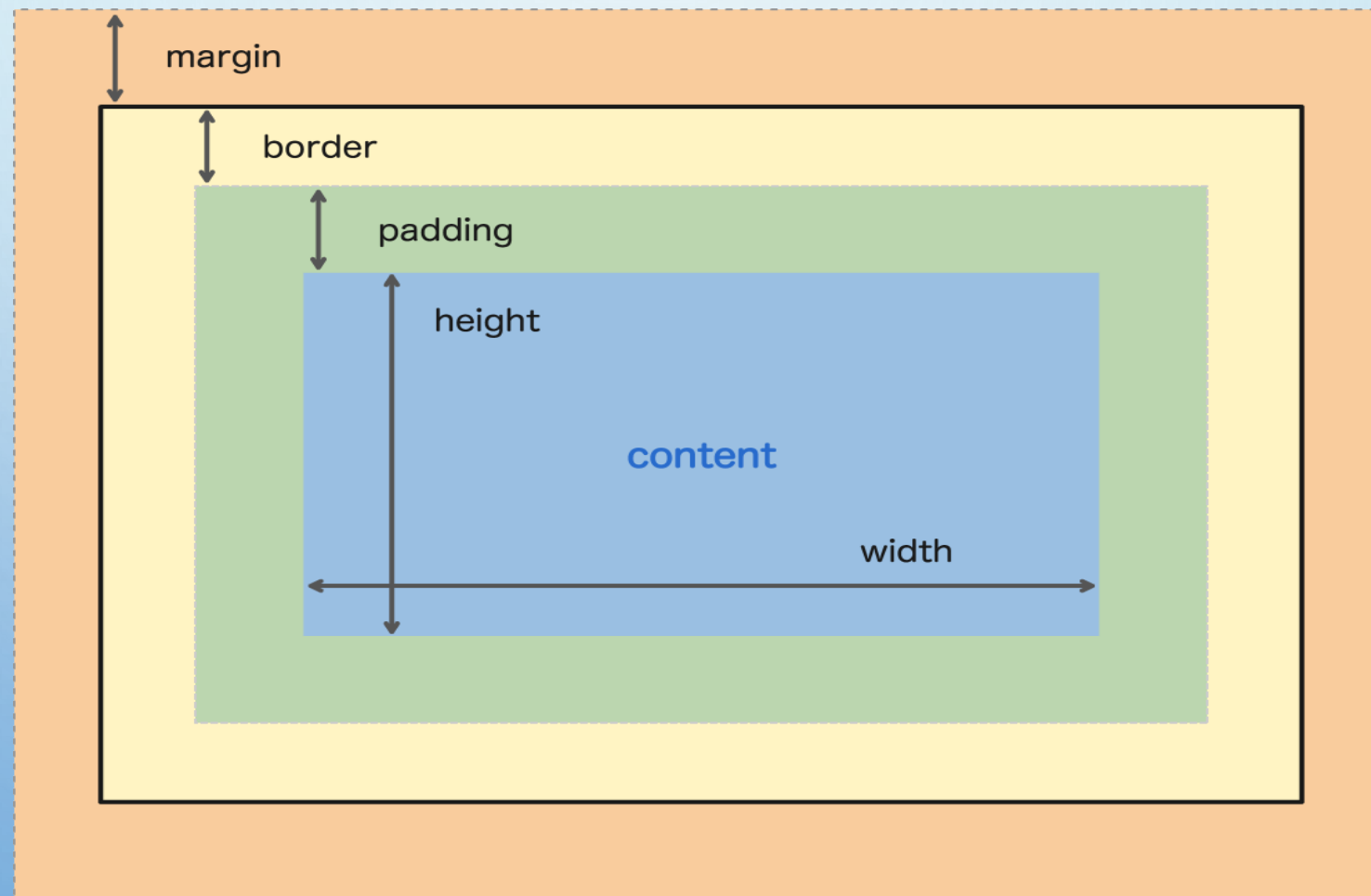
BOX MODEL

THE CSS BOX MODEL IS A FUNDAMENTAL CONCEPT THAT DESCRIBES HOW HTML ELEMENTS ARE RENDERED AND HOW THEIR DIMENSIONS ARE CALCULATED.

IT CONSISTS OF SEVERAL COMPONENTS THAT DEFINE THE SPACE AN ELEMENT OCCUPIES WITHIN THE LAYOUT OF A WEBPAGE. THE BOX MODEL CONSISTS OF FOUR MAIN PARTS:

1. CONTENT
2. PADDING
3. BORDER
4. MARGIN

BOX MODEL



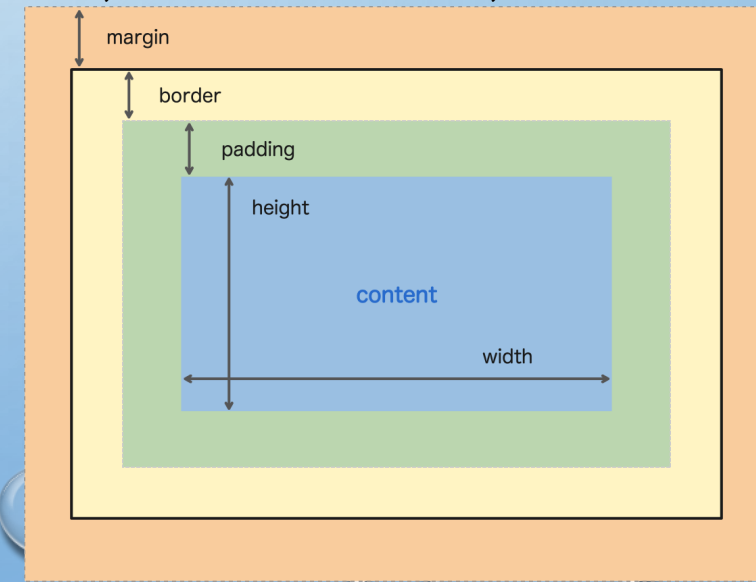
BOX MODEL

CONTENT:

THE INNERMOST PART OF THE BOX MODEL IS THE CONTENT AREA, WHICH CONTAINS THE ACTUAL CONTENT OF THE HTML ELEMENT. THIS CONTENT CAN BE TEXT, IMAGES, VIDEOS, OR ANY OTHER MEDIA.

PADDING:

PADDING IS THE SPACE BETWEEN THE CONTENT AND THE ELEMENT'S BORDER. IT ADDS INTERNAL SPACE WITHIN THE ELEMENT, SEPARATING THE CONTENT FROM THE BORDER. PADDING CAN BE SPECIFIED USING VARIOUS UNITS LIKE PIXELS, PERCENTAGES, OR EMS.

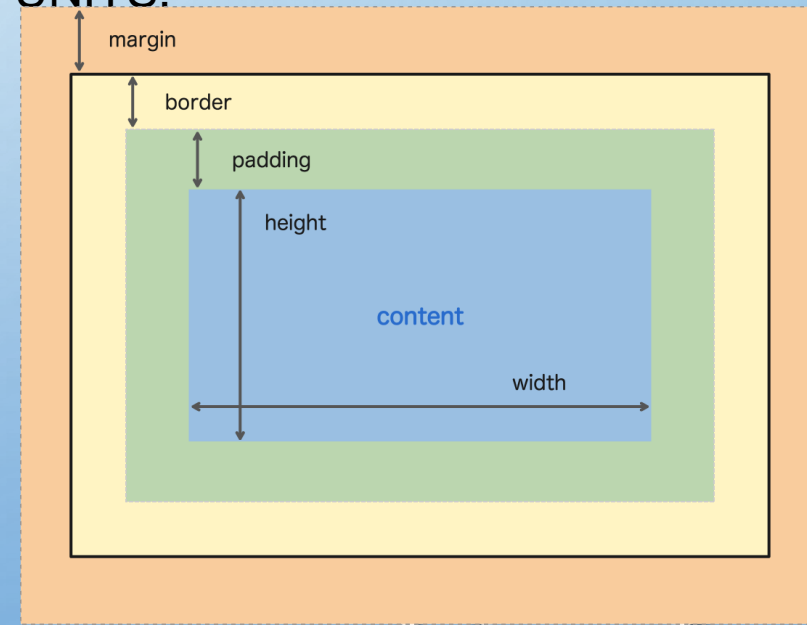


BORDER:

THE BORDER SURROUNDS THE PADDING AND CONTENT OF THE ELEMENT. IT IS A VISIBLE OR INVISIBLE BOUNDARY THAT DEFINES THE EDGES OF THE ELEMENT. BORDERS CAN HAVE DIFFERENT STYLES (SOLID, DASHED, ETC.), COLORS, AND WIDTHS.

MARGIN:

THE MARGIN IS THE SPACE OUTSIDE THE ELEMENT'S BORDER. IT CREATES SPACE BETWEEN THE ELEMENT AND ITS NEIGHBORING ELEMENTS IN THE LAYOUT. LIKE PADDING, MARGINS CAN BE SPECIFIED USING DIFFERENT UNITS.



MORE ON PADDING

HTML

```
<DIV>THIS ELEMENT HAS A PADDING OF 70PX.</DIV>
```

CSS

```
DIV {
```

```
    PADDING: 70PX;
```

```
    BORDER: 1PX SOLID #4CAF50;
```

```
}
```

OUTPUT:

This element has a padding of 70px.

MORE ON PADDING

CSS HAS PROPERTIES FOR SPECIFYING THE PADDING FOR EACH SIDE OF AN ELEMENT: PADDING-TOP, PADDING-RIGHT, PADDING-BOTTOM, PADDING-LEFT

DIV {

BORDER: 1PX SOLID BLACK;

BACKGROUND-COLOR: LIGHTBLUE;

PADDING-TOP: 25PX;

PADDING-RIGHT: 30PX;

PADDING-BOTTOM: 75PX;

PADDING-LEFT: 180PX;

}

This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

PADDING - SHORTHAND PROPERTY

TO SHORTEN THE CODE, IT IS POSSIBLE TO SPECIFY ALL THE PADDING PROPERTIES IN ONE PROPERTY.

THE PADDING PROPERTY IS A SHORTHAND PROPERTY FOR THE FOLLOWING INDIVIDUAL PADDING PROPERTIES:

PADDING-TOP, PADDING-RIGHT, PADDING-BOTTOM, PADDING-LEFT

SO, HERE IS HOW IT WORKS:

IF THE PADDING PROPERTY HAS FOUR VALUES:

```
DIV {
```

```
PADDING: 25PX 50PX 75PX 100PX;
```

```
}
```

MEANS TOP PADDING IS 25PX, RIGHT PADDING IS 50PX, BOTTOM PADDING IS 75PX, LEFT PADDING IS 100PX

BOX MODEL EXAMPLE

```
<!DOCTYPE HTML>
```

```
<HTML LANG="EN">
```

```
<HEAD>
```

```
<META CHARSET="UTF-8">
```

```
<META NAME="VIEWPORT" CONTENT="WIDTH=DEVICE-WIDTH, INITIAL-SCALE=1.0">
```

```
<LINK REL="stylesheet" href="styles.css">
```

```
<TITLE>BOX MODEL EXAMPLE</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>CSS BOX MODEL EXAMPLE</H1>
```

```
<!-- THE BOX ELEMENT WITH CONTENT, PADDING, BORDER, AND MARGIN -->
```

```
<DIV CLASS="BOX"> THIS IS THE CONTENT INSIDE THE BOX. </DIV>
```

```
<P>MORE CONTENT OUTSIDE THE BOX.</P>
```

```
</BODY>
```

```
</HTML>
```



```
/* APPLY STYLES TO THE BOX */
```

```
.BOX {
```

```
  WIDTH: 200PX;
```

```
  HEIGHT: 100PX;
```

```
  PADDING: 20PX;
```

```
  BORDER: 2PX SOLID #3498DB;
```

```
  MARGIN: 20PX; }
```

```
/* APPLY SOME ADDITIONAL STYLES FOR CLARITY */
```

```
BODY {
```

```
  TEXT-ALIGN: CENTER;
```

```
  MARGIN: 0;
```

```
  PADDING: 20PX; }
```

```
H1 {
```

```
  COLOR: #3498DB;
```

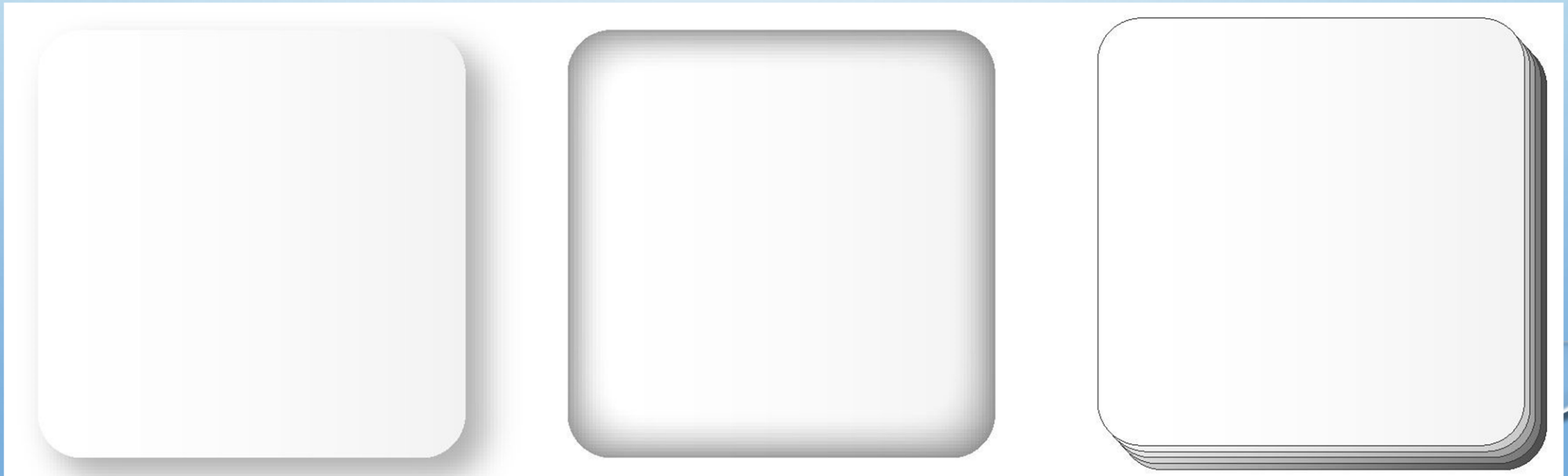
```
}
```



BOX SHADOWS

In CSS enable the creation of a shadow effect around an element, enhancing its visual presentation and providing depth.

The box-shadow property is used to define the characteristics of the shadow, including its color, blur radius, spread distance, and position.



BOX-SHADOW

SYNTAX:

BOX-SHADOW: [HORIZONTAL-OFFSET] [VERTICAL-OFFSET] [BLUR-RADIUS] [SPREAD-DISTANCE] [COLOR];

We already know about horizontal-offset, vertical-offset, blur-radius and color while we learnt about text-shadow. So, lets see what's **spread-distance**.

SPREAD DISTANCE: The **fourth value** represents the Spread distance of the shadow. It determines whether the shadow expands or contracts. Positive values expand the shadow, while negative values contract it.

```
div {  
  width: 300px;  
  height: 100px;  
  padding: 15px;  
  background-color: coral;  
  box-shadow: 4px 4px 8px 2px #333;  
}
```

This is a div element with a box-shadow

TEXT SHADOWS:

Text shadows in CSS allow you to create a shadow effect behind text elements, providing a visual enhancement to the typography.

The text-shadow property is used to define the characteristics of the shadow, such as its color, blur radius, and offset.



CSS
TEXT SHADOW
EFFECTS

TEXT-SHADOW

SYNTAX:

TEXT-SHADOW: [HORIZONTAL-OFFSET] [VERTICAL-OFFSET] [BLUR-RADIUS] [COLOR];

HORIZONTAL AND VERTICAL OFFSETS: The **first two values** represent the horizontal and vertical offsets of the shadow. Positive values move the shadow to the right (horizontal) or down (vertical), while negative values move the shadow to the left or up.

BLUR RADIUS: The **third value** represents the blur radius of the shadow. It determines how much the shadow is softened or blurred. Larger values result in a more diffuse and softer shadow.

COLOR: The **fourth value** represents the color of the shadow. It can be specified using a color name, hexadecimal code, RGB, or RGBA values.

TEXT-SHADOW: 3PX 5PX 4PX ORANGE;

Text-shadow effect!

TEXT SHADOW CSS

MULTIPLE SHADOWS:

YOU CAN APPLY MULTIPLE SHADOWS TO A SINGLE TEXT ELEMENT BY SEPARATING EACH SHADOW DEFINITION WITH A COMMA.

TEXT-SHADOW: 3PX 5PX 4PX RED, -2PX -3PX 5PX BLUE;



Text-shadow effect!