

CASCADING STYLE SHEETS (CSS)

css is a language used for describing the presentation of web pages, including colors, layouts and fonts.

css is independent of HTML and can be used with any XML - markup language.

HTML was created to describe the content of the webpage.

css is used for the presentation.

The style definitions are normally stored in external css file.

With external style sheet file, you can change the look of an entire website by changing just one file.

Levels of Style Sheets

The three levels of style sheets, in order from lowest level to highest level, are :-

- inline
- document level and
- external

Inline Style Sheet :-

Inline style sheets apply to the content of a single

element.

We add the style within our HTML tags. This gets highest priority than internal or external defined sheets.

Ex:-

```
<h1 style = "background-color: #f1f1f1;">
```

This is H1 tag </h1>

② Internal / Document Level / Embedded Style sheet:

~~External~~ Embedded style sheets apply to the whole body of a document. They are defined inside the head tag of the page.

This style gets priority over the external styles.

Ex:-

```
<html>
```

```
<head>
```

```
<title> Type a title </title>
```

```
<style type = "text/css">
```

```
    p { font-family: arial, verdana;
```

```
        background-color: #f1f1f1; }
```

3

</style>

</head>

<body>

<p> This is the content </p>

</body>

</html>

Document level is the way to impose a uniform style on the presentation of all of the content of a document.

3) External Style Sheet :-

External Style sheets can apply to the bodies of any number of documents.

External style sheets are not part of any of the documents to which they apply. They are stored separately and are referenced in all documents that use them.

The <link> tag is used to specify external style sheets.

Within <link>, the rel attribute is used to specify the relationship.

The href attribute of <link> is used to specify the URL of the style sheet document.

Ex:-

```
<link rel = "stylesheet" type = "text/css"  
      href = "http://www.cs.usc.edu/styles/  
      wbook.css"/>
```

Style Specification formats.

The format of a style specification depends on the level of style sheet.

- ① Inline style specification appear as values of the style attribute of a tag.

General form :-

```
style = "property_1: value_1; property_2: value_2;  
        ...; property_n: value_n;"
```

- ② Document style specification appears as the content of a style element within the header of a document.

General form :-

```
<style type = "text/css">  
    rule list  
</style>
```

Each style rule in a rule list has two parts.

① A selector : Indicate the tag or tags affected by the rule.

② A list of properties / value pairs.

The rule list is delimited by braces rather than double quotes.

```
selector { property_1: value_1;  
           property_2: value_2; ...;  
           property_n: value_n; }
```

If a property is given more than one value, then they can be ~~not~~ separated by spaces or commas.

External style sheets have a form similar to that of document style sheets.

The external file consists of a list of style rules.

Comments :-

Different type of comments are needed.

css comments are introduced with /* and */

terminated with a '/'.

Ex:-

```
<style type = "text/css">
```

```
/* styles for initial paragraph */
```

```
/* styles for other paragraphs */
```

```
</style>
```

Selector forms

The selector can have a variety of forms.

1. Simple selector form :-

Simple selector form is a single element name, such as h1.

In this case, the property values in the rule apply to all occurrences of h1.

The selector can be a list of element names separated by comma.

Ex:-

```
h1 { font-size : 24pt; }
```

```
h2, h3 { font-size : 30pt; }
```

<1.848e </head>

color: #ff0000;

padding: 10px;

border-radius: #222;

<div> <style> .text { font-size: 1em; } .bookoutline { border: 1px solid black; padding: 10px; } </style> <div>

element as our container. This is done by applying the border-radius to the outermost element of our selector, or a descendant selector.

In the body of the document, try it a different way that our descendants of the element that did not have a border-radius to the content of our

body { margin: 14pt; }

- :-)

Specifying the element name,

already in the selector, with only whitespace the document. This is done by shifting the element content of elements in certain positions in the selector can also specify that the style should

#2, #3 must define 30pt. The first specifys that the first content of

```
<body>  
  <p>Hello World</p>  
  <em>Again</em></p>
```

</body> and the end of the file with </html>
</html>

Class Selector

class selector is used to allow different occurrences of the same tag to use different style specifications.

Ex :- if you want two paragraph styles in a document —

- 1) normal
 - 2) warning

you could define these two classes in the content of a <style> tag as follows:-

```
<head>
<style type="text/css">
    p.normal { text-align: center;
                color: red; }
    p.warning { text-align: center;
                 color: blue; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
    <p class = "normal">
```

A paragraph of text we want to be presented in red color.

```
</p>
```

```
    <p class = "warning">
```

A paragraph of text that we want to be presented in the blue color

```
</p>
```

```
</body>
```

```
</html>
```

3. Generic selector

Sometimes it's consistent to have a class of style specifications that applies to the content of more than one kind of tag. In such cases, instead of defining separate classes for each type of tag, we can define a generic class, which is defined without a tag name, in place of tag name, we use the bname of generic class, which must begin with a period.

Ex:- .sale { property : value; }

```
<head>
<style type = "text/css">
    .sale { text-align: center;
            color: blue;
        }
</style>
</head>
<body>
    <h3 class = "sale"> Weekend </h3>
    <p class = "sale">
        - - - - -
        - - - - -
        - - - - -
    </p>
</body>
</html>
```

4. id selector

An id selector allow the application of a style to one specific element.

General form:

```
# specific-id { property: value; }
```

The style specified in the id selector applies to

the element with specific id.

Ex:-

```
<head>
<style type = "text/css">
# section 14 { font-size: 20 pt ;
}
</style>
</head>
<body>
<h2 id = "section 14"> 1.4 Calico Cats </h2>
</body>
</html>
```

Universal Selector

The universal selector, denoted by an asterisk (*), applies its style to all elements in the document.

Ex:-

```
<head>
<style type = "text/css">
* { color : red ; }
</style>
</head>
<body>
<p> ..... </p> </body></html>
```

This makes all elements in the document red.

6: Pseudo classes

Pseudo classes are used to define a special style of an element.

→ Two pseudo classes

1) hover

2) focus

are supported by FX2.

→ IE7 supports hover, but not focus.

→ Instead of a period, the names of pseudo classes begin with a colon (:).

→ hover pseudo class applies when its associated element has the mouse cursor over it.

→ focus pseudo class applies when its associated element has focus.

Ex:-

<html>

<head>

<style type = "text/css">

a:link {color: red;}

a:visited {color: green;}

a:hover {color: pink;}

a:focus {color: blue;}

</style> </head>

<body>

<p>

This is a link

<la>

<1p>

</body>

</html>

PROPERTY VALUE FORMS

css includes 60 different properties in seven categories :

fonts,

lists,

alignment of text,

margins,

colors,

backgrounds of

borders.

URL property values use a form that is slightly different from references to URLs in links.

The actual URL is placed in parentheses and preceded by url, for ex:-

url (tetons.jpg)

There can be no space between url and left parenthesis.

Color property values can be specified as color names, as six-digit hexadecinal numbers, or in RGB form.

Ex:

powerblue

or

#B0E0E6

or

rgb(176, 224, 230)

FONT PROPERTIES

The font properties are among the most commonly used of the style-sheet properties.

1) Font-families

The font-family property is used to specify a list of font names.

The browser will use the first ~~font~~ font in the list that it supports.

Ex:-

font-family : Arial, Helvetica, futura;

an alternative of its choosing.

Arial, Helvetica and Fettika are san-serif fonts, these can be used better as:-

Font-family : Arial, Helvetica, fettika, san-serif

Generic Name	Examples
script	Times New Roman, Garamond
sans-serif	MS Arial, Helvetica
cursive	Cafisch script, zapf-Chancery
fantasy	Clitter, cottonwood

If a font name has more than one word, the whole name should be delimited by single quotes.

Font-family : 'Times New Roman'

Font-size

The font-size property does ~~not~~ what if name implies.

font-size : 10 pt;

relative font-size values are defined :-

xx-small,

x-small,

small,

medium,

large,

x-large,

xx-large,

smaller f.

larger.

The disadvantages of the relative font sizes is the lack of strict font size control.

Different browsers use different values for them.

Ex:-

small might mean 10 pts on one browser &
8 pts on another.

3) font-variant

The default value of the font-variant property is normal.

This property can be set to small-caps to specify small capital characters.

Small cap characters are all uppercase, but the letters ^{that} are normally uppercase are somewhat larger than those that are normally lowercase.

4) font-styles

The font-style property is commonly used to specify italic.

font-style : italic

5) Font Weights

Used to specify the degree of boldness.

Ex:-

font-weight: bold

Values for font-weight are :-

1) bold

2) normal

3) bolder

4) lighter

Specific numbers also can be given in multiples of 100 from 100 to 900.

400 - normal

700 - bold.

6) Font Shorthand

If more than one font property must be specified, the values can be stated in a list as the value of the font property.

Ex:-

font: bold 14pt 'Times New Roman' Palatino.

This specifies that the font weight should be bold, the font size should be 14pt and either Times New Roman or palatino font should be used.

The order in which the property values are given in a font value list is important.

The order must be :-

1) The font names must be last

2) The font size must be second last

3) The font style, font variant and font

~~weight~~ weight, when they are included, can be in any order but must precede the font size and font names.

Example :-

<html>

<head> <title> Font Properties </title> ~~</head>~~

<style type = "text/css">

p.major { font-size: 14pt;

font-style: italic;

font-family: 'Times New Roman';

}

p.minor { font: 10pt bold 'Courier New'; }

h2 { font-family: 'Times New Roman';

font-size: 24pt; font-weight: bold; }

h3 { font-family: 'Courier New';

font-size: 18pt; }

1.1 The Basics of Computers

Chapter 1 Introduction

Too many don't make a profit, but they
certainly can get you in a lot of trouble.

If a job is worth doing, it's worth doing
right away.

<140>

<141>

<142> 1.1 The Basics of Computers <143>

<142> Chapter 1 Introduction <141>

<141>

Too many don't make a profit, but they
certainly can get you in a lot of trouble.

<141> $p_{down} = \frac{1}{2}$

<141>

right

If a job is worth doing, it's worth doing
right

<141> $p_{down} = \frac{1}{2}$

<140>

<141>

<140>

Scanned by CamScanner

Same Code Using External Style Sheets

fonts.html

```
<html>
```

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

```
<body>
```

```
  <p class = "major">
    If a job is worth doing, it's worth doing
                                right.
```

```
  <p>
```

```
  <p class = "minor">
```

Two wrongs don't make a right, but they
certainly can get you in a lot of trouble.

```
  </p>
```

```
  <h2> chapter 1 Introduction </h2>
```

```
  <h3> 1.1 Basics of Computers </h3>
```

```
</body>
```

```
</html>
```

The external style sheet is named as "styles.css".

styles.css

p. major { font-size: 14pt;

font-style: italic;

font-family: 'Times New Roman';

}

p. minor { font: 10pt bold 'Courier New'; }

h2 { font-family: 'Times New Roman';

font-size: 24pt;

font-weight: bold; }

h3 { font-family: 'Courier New';

font-size: 18pt; }



Text-Decoration

Text decoration property is used to specify some special features of text.

→ The available values are: line-through,
overline,
underline,
none (default).

Many browsers underline links. The none value can be used to avoid this.

Example:-

<html>

<head> <title> Text Decoration </title>

<style type = "text/css">

p.delete { text-decoration: line-through; }

p.cap { text-decoration: overline; }

p.attention { text-decoration: underline; }

<style>

</head>

<body>

<p class = "delete">

This illustrates line-through

<ip>

<p class = "cap">

This illustrates overline

<ip>

<p class = "attention">

This illustrates underline

<ip>

</body>

</html>

Output :-

This illustrates line-through

This illustrates overline

This illustrates underline

The letter-spacing property controls the amount of space between character in text.

value of letter-spacing are any length property value.

e.g. 2mm

LIST-PROPERTIES

Two presentation details of lists are specified in HTML documents:

- 1) the shape of bullet that precede the items in an unordered list &
- 2) the sequencing values that precede the items in ordered lists.

The list-style-type property is used to specify both of these.

The list-style-type property of an unordered list can be set to disc,
circle,
square or
none.

The default property value for bullet is disc.

Ex:-

```
<head> <title> Unordered List </title>
<style type = "text/css">
    ul { list-style-type : square }
</style>
</head>
<body>
    <h3> Some Common Single-Engine Aircraft
    </h3>
```


 Cessna Sky hawk

 Beechcraft Bonanza

 Piper Cherokee

<body>

Output :-

Some Common Single - Engine Aircraft

- Cessna Skyhawk
- Beechcraft Bonanza
- Piper Cherokee

~~Style~~ can be allowed to define

Styles can be defined to allow different list items
to have different bullet types :-

Ex:-

 - - - -

<style type = "text/css">

li. disc { list-style-type : disc; }

li. square { list-style-type : square; }

li. circle { list-style-type : circle; }

<style>

</head>

```
<body>
  <h3> Some common Aircrafts </h3>
```

```
  <ul>
```



```
    <li class = "disc"> Cessna </li>
```

```
    <li class = "square"> Beechcraft </li>
```

```
    <li class = "circle"> Piper </li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

Output :-

Some Common Aircrafts

- Cessna
- Beechcraft
- Piper

- Any image can be used in a list item bullet.
- Such a bullet is specified with the list-style-image property, whose value is specified with the url form.

Ex :- <head>

```
  <style type = "text/css">
```

```
    li.image { list-style-image: url('small.gif') }
```

}

```
  </style>
```

```
</head>
```

```
<body>
```

 <li class = "image"> Beechcraft

<body>

</html>

When ordered lists are nested, it's best to use different kinds of sequence values for the different levels of nesting.

The list-style-type property can be used to specify the type of sequencing values.

Possible sequencing values for Ordered lists

<u>Property</u>	<u>Values</u>	<u>Sequence Type</u>	<u>First four values</u>
decimal		Arabic Numerals	1, 2, 3, 4
upper-alpha		Uppercase letters	A, B, C, D
lower-alpha		Lowercase letters	a, b, c, d
upper-roman		Uppercase-Roman	I, II, III, IV
lower-roman		Numerals	i, ii, iii, iv
			Numerals

Example :-

<html>

<head> <title> Sequence type </title>

```
<style type = "text/css">
```

```
ol { list-style-type: upper-alpha; }
```

```
ol ol { list-style-type: upper-alpha; }
```

```
ol ol ol { list-style-type: decimal; }
```

```
<ol>
```

```
<li>
```

```
<li>
```

```
<li> General Aviation
```

```
<ol>
```

```
<li> Single - Engine Aircraft
```

```
<ol>
```

```
<li> Tail wheel </li>
```

```
<li> Tricycle </li>
```

```
<ol>
```

```
<li> Dual - Engine Aircraft
```

```
<ol>
```

```
<li> Wing - mounted engines </li>
```

```
<li> Fuselage - mounted engines </li>
```

```
</ol>
```

```
</ol>
```

```
</li>
```

```
<li> Commercial Aviation
```

<obj>
 Dual - engine

<obj>

 wing - mounted engine

 fuselage - mounted engines

<obj>

 Tri - Engine

<obj>

 third engine in stabilizer

 third engine in fuselage

<obj>

<obj>

<obj>

Aircraft Types

I. General Aviation

A. Single - Engine Aircraft

1. Tail wheel

2. Tri cycle

B. Dual - Engine Aircraft

1. Wing - mounted engines
2. Fuselage - mounted engines.

II. Commercial Aviation

A. Dual Engine

1. Wing-mounted engines
2. Fuselage-mounted engines

B. Tri-engine

1. Third engine in stabilizer
2. Third engine in fuselage

COLOR

Color Groups

Three levels of collection of colors might be used by an HTML document.

Colors in CSS are specified as :-

- a valid color name : "red"
- an RGB value : "rgb(255,0,0)"
- a hexadecimal value : "#ff0000"

The smallest useful set of colors includes only those that have standard names. This collection of sixteen colors is called the named colors.

The names and hexadecimal codes for the named color are given below :-

Name	Hexadecimal Code	Name	Hexadecimal Code
black	000000	green	008000
silver	C0C0C0	lime	00FF00
gray	808080	olive	808000
white	FFFFFF	yellow	FFFF00
magenta	FF0000	navy	000080
red	FF0000	blue	0000FF
purple	800080	teal	008080
fuchsia	FF00FF	aqua	00FFFF

A larger set of colors, called the web palette, include 216 colors. These colors are called web safe colors.

Elements of this set of colors have hexadecimal values for red, green and blue that are restricted to 00, 33, 66, 99, CC and FF.

Color Properties :-

The color property is used to specify the foreground color of HTML elements.

Ex:- <html> <head> <title> Color Properties </title>

<style type = "text/css">

th { color : red }

```
th { color: orange }
```

```
<style>
```

```
</head>
```

```
<body>
```

```
<table border = "5px">
```

```
<tr>
```

```
<th class = "red"> Apple </th>
```

```
<th class = "orange"> Orange </th>
```

```
<th class = "orange"> Screw </th>
```

```
<tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

The background-color property is used to set the background color of an element, where the element could be the whole body of the document -

```
Ex:- <html>
```

```
<head>
```

```
<style type = "text/css">
```

```
p { stand-out: front-size: 24 pt;
```

```
color: blue;
```

```
background-color: red;"}
```

```
<style> </head>
```

```
<body>
```

```
<p class = "standout">
```

To really make it stand out, we can add a red background.

< p >

< body >

</body>

Output :- To really make it stand out,
we a red background

ALIGNMENT OF TEXT



text-indent

- The first line of a paragraph can be indented using the text-indent property.
- This property takes either a length or a % value.

Ex:-

<html>

<head> <title> Text-indent </title>

<style type = "text/css">

p {
 text-indent : 20px
}

text-indent : 0.5in }

</style>

</head>

<body>

<p class = "indent">

Now is the time for all good web

presentation details in their documents.

<ip>
<body>
<html>

Output :-

Now is the time for all good web programmers to begin using details in their documents.

text-align

The text-align property, for which the possible keyword values are : left, center,

right & justify is used

* to arrange text horizontally

Ex:-

```
p { text-align: right }
```

The default value for text-align is left.

float property

The float property which is set for images and tables, is used to specify that text should flow around some element. The possible

values for float are : left,
right, f
none (default).

Example :-

```
<html>
<head> <title> The float property </title>
<style type = "text/css">
img { float: right }
</style>
<head>
<body>
<p>
<img src = "c210news.jpg" alt = "Picture" />
<p>
```

This is a picture of a Cessna 210. The 210 is the flagship single - engine Cessna aircraft.

Although the 210 began as a four - place aircraft.

```
<hp>
</body>
</html>
```

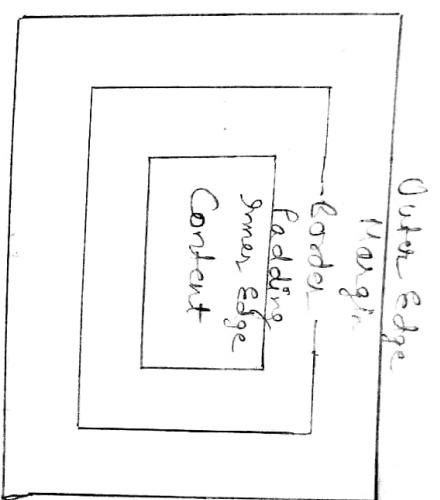
Output :-



This is a picture of a Cessna 210. The 210 is the flagship single - engine Cessna aircraft. Although the 210 began as a four - place aircraft.

THE BOX MODEL

Diagram



Box model is used when talking about design and layout.

CSS box model is a box that wraps around every HTML element.

It consists of :-

- i) Content : the content is the part of box where image and text appears.
- ii) Padding : the amount of space between the content and the border.
- iii) Border : Border is the one that goes around padding and content.
- iv) Margin : clearly our area outside the border.

Borders :-

Every element has a property - border-style, that controls the element's content has a border, as well as the style of the border.

The border-style has different values :- solid, dotted, double & dashed.

The default value is none, which is why the content of elements do not normally have borders.

The styles of one of the four sides of an element can be set with :-

- v) border-top-style
- ii) border-bottom-style
- w) border-left-style
- v) border-right-style

→ The border-width property is used to specify the thickness of a border.

Possible values are :- thin, medium (default), thick or

a length value in pixels.

Setting border-width sets the thickness of the four

sides of an element.

The width of one of the four sides can be set using :-

1) border - bottom - width

2) border - left - width

3) border - right - width.

The colors of border is controlled by the border-color property.

The individual borders of an element can be colored differently through :-

i) border - top - color,

ii) border - bottom - color,

iii) border - left - color,

iv) border - right - color.

NOTE :- If a table has a border that was specified

with the border attribute, the border property override the original border.

Example :-

```
<html>
<head> <title> Table borders </title>
```

```
<style type = "text/css">
table { border - top - width : medium;
```

border - bottom - width : thick ;
border - top - color : red ;
border - bottom - color : blue ;
border - top - style : dotted ;
border - bottom - style : dashed ;

{}

P { border - style : dashed ;
border - width : thin ;
border - color : green ;

}

< / style >

< / head >

< body >

< table border = "5" >

< caption > fruit Juice Drinks < / caption >

< tr >

< th > < / th >

< th > Apple < / th >

< th > Orange < / th >

< th > Grapes < / th >

< / tr >

< tr >

< th > Breakfast < / th >

< td > 0 < / td >

< td > 1 < / td >

< td > 0 < / td >

<tr>

<th> Lunch </th>

<td> 1 </td>

<td> 0 </td>

<td> 0 </td>

<tr>

<tr>

<th> Dinner </th>

<td> 0 </td>

<td> 0 </td>

<td> 1 </td>

<tr>

<table>

<p>

Now is the time for all good men to

Leave CSS.

<p>

</body>
</html>

Output :-

Fruit Juice Drinks		
	Apple	Orange
Breakfast	0	0
Lunch	0	0
Dinner	0	1

Margins and Padding :-

padding :- the space between the content of an element and its border.

margin :- space between the border of an element and the element's neighbor.

When there is no border, the margin plus the padding is the space between the content of an element and its neighbor.

The margin properties are named margin, which applies to four sides of an element :-

margin - left,
margin - right,
margin - top +
margin - bottom.

The ~~margin~~ padding properties are named padding, which applies to all four sides :-

padding - left,
padding - right,
padding - top,
padding - bottom.

Program Code 1-

```
<!-- marpads.html -->
<?xml version = "1.0" encoding = "utf-8"?>
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Margins and Padding </title>
<style type = "text/css">
```

```
p.one {margin: 0 .2in;
padding: 0 .2in;
background-color: #c0c0c0;
border-style: solid;}
```

```
p.two {margin: 0 .1in;
padding: 0 .3in;
background-color: #c0c0c0;
border-style: solid;}
```

```
p.three {margin: 0 .3in;
padding: 0 .1in;
background-color: #c0c0c0;
border-style: solid;}
```

```
p.four {margin: 0 .4in;
background-color: #c0c0c0;}
```

```
p.five {padding: 0 .4in;
background-color: #c0c0c0;}
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p> Here is the first line.
```

```
</p>
```

```
<p class = "one">
```

```
Now is the time for all good Web programmers to
```

```
learn to use style sheets. <br /> [margin = 0 .2in,
```

```
padding = 0 .2in]
```

```
</p>
```

```
<p class = "two">
```

```
Now is the time for all good Web programmers to
```

```
learn to use style sheets. <br /> [margin = 0 .1in,
```

```
padding = 0 .3in]
```

```
</p>
```

```
<p class = "three">
```

```
Now is the time for all good Web programmers to
```

```
learn to use style sheets. <br /> [margin = 0 .4in,
```

```
padding = 0 .1in]
```

```
</p>
```

```
<p class = "four">
```

```
Now is the time for all good Web programmers to
```

```
learn to use style sheets. <br /> [margin = 0 .4in,
```

```
no margin, no border]
```

```
</p>
```

```
<p class = "five">
```

```
Now is the time for all good Web programmers to
```

```
learn to use style sheets. <br /> [padding = 0 .4in,
```

```
no margin, no border]
```

```
</p>
```

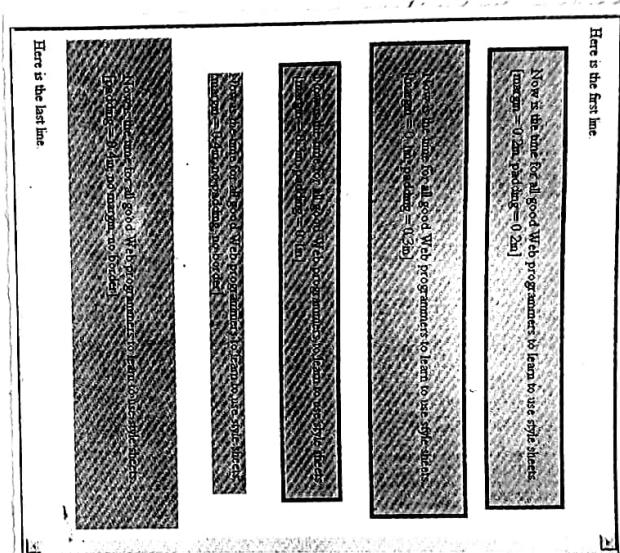
```
<p> Here is the last line.
```

```
</p>
```

```
</body>
```

```
</html>
```

Output :-



BACKGROUND IMAGES

The background-image property is used to place an image in the background of an element.

Example :-

```
<html>
<head> <title> Background Images </title>
<style type = "text/css">
body { background-image: url (cif2.gif);}
p { margin-left: 30px; margin-right: 30px;
```

margin-top: 5px; font-size: 14pt; }

卷之三

卷之三

卷之三

↳ *Die Wissenschaften der Erde*

an^other airplane

卷之三

The climb rate of the B-25 at sea level is 720 feet per minute.

Output

<1p>
<body>
</body>

Tiling :- the background image is replicated as necessary to fill the area of the element. The replication is called tiling.

Tiling can be controlled with background-repeat property.

background-repeat values :-
in : -A (100% 80%)

- iii) no-repeat
iv) repeat-x
v) repeat-y

no-repeat value specifies that just one copy of the image is to be displayed.
repeat-x value means that the image is to be repeated horizontally.

repeat-y value means to repeat vertically.

The position of nonrepeated background image can be specified with background-position property, which can take a large number of diff. values. The keyword values are :- top,
center,
bottom,
left,
right.

These can be used as combinations, as :-
~~top left~~, top left,
bottom right,
top center.

If only one keyword is given, the other is assumed to be center.
So, top is equivalent to top center.

The and <div> tag sometimes, we want to apply special font properties to less than a whole paragraph of

text.

for ex:- It is often useful to have a word or phrase in a line appear in a different font size or color.

`` tag is used for this purpose.

→ There is no default layout for the content of ``.

Ex:-

`<html>`

`<head> <title> span tag </title>`

`<head>`

`<body>`

`<p>`

It is seen that today `` is

displayed as `holiday `

`<p>`

`</body>`

`</html>`

Here, the word "is holiday" is not displayed differently, as we have not specified any style for the `` tag.

→ The purpose of `` tag is to change property values of `para` of a line of content.

Ex:-

<html>

<head> <title> Span tag Example </title>
<style type = "text/css">

• Spanned

{ font-size : 24pt ;
font-family : Arial ;
color : red }

< / style > < head >

< body >

< p > It is sure that today is

< span class = " spanned " > holiday

< / span >

< / p >

< / body >

< / html >

Output :-

It is sure that today is holiday.

<div> tag

<div> tag is used to apply a style to a section of a document rather than each paragraph,

→ The primary use of `<div>` tag is to specify presentation details for a section or division of a document.

Ex:

```
<!DOCTYPE html>
<html> <head> <title> My First Web Page </title>
<body>
<h1> Welcome to my first web page! </h1>
<h2> This is a heading </h2>
<p> This is a normal paragraph. </p>
</body>
</html>
```

```
div { color: #0000FF }
```

```
<div class = "primary">
<h1> Welcome to my first web page! </h1>
<h2> This is a heading </h2>
<p> This is a normal paragraph. </p>
</div>
```

This is a paragraph in div element.

```
<div>
<p> This is a normal paragraph </p>
</div>
```

`</body>`

Output:

This is a normal paragraph.