# Web Programming

## Lecture 5

*Cascading Style Sheets (CSS)*

***Part 2***

By

***Asst. Lect. Ali Al-khawaja***

# CSS Fonts

## Font Selection is Important

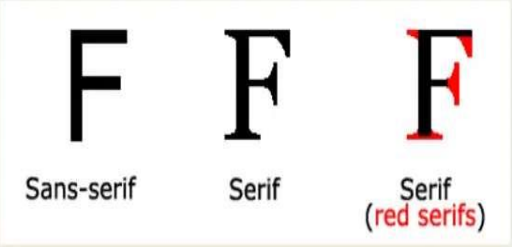
- Choosing the right font has a huge impact on how the readers experience a website.
- The right font can create a strong identity for your brand.
- Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

# Generic Font Families

- In CSS there are five generic font families:

1. **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
2. **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
3. **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
4. **Cursive** fonts imitate human handwriting.
5. **Fantasy** fonts are decorative/playful fonts.

- All the different font names belong to one of the generic font families.
- Difference Between Serif and Sans-serif Fonts is in the image



# Some Font Examples

| Generic Font Family | Examples of Font Names |
|---|---|
| Serif | Times New Roman<br>Georgia<br>Garamond |
| Sans-serif | Arial<br>Verdana<br>Helvetica |
| Monospace | Courier New<br>Lucida Console<br>Monaco |
| Cursive | Brush Script MT<br>Lucida Handwriting |
| Fantasy | Copperplate<br>Papyrus |

# The CSS font-family Property

- In CSS, we use the font-family property to specify the font of a text.
-
```
.p1 {
  font-family: "Times New Roman", Times, serif;
}
```

Ex: This is a paragraph, shown in the Times New Roman font.

```
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
```

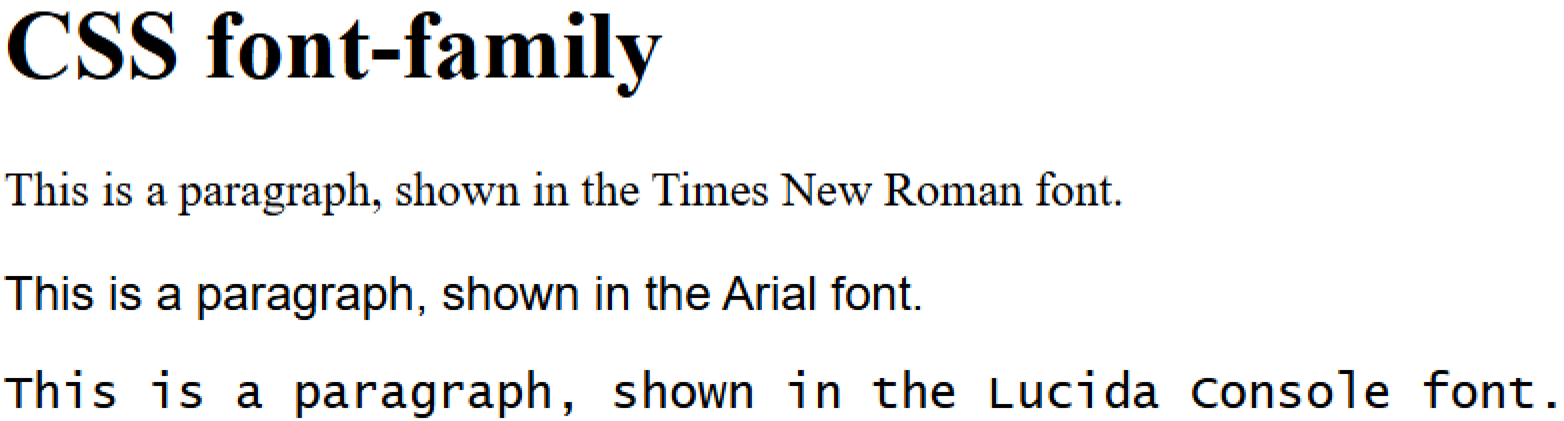
Ex: This is a paragraph, shown in the Arial font.

```
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
```

Ex: This is a paragraph, shown in the Lucida Console font.

```
<!DOCTYPE html>
<html>
<head>
<style>
.p1 {
  font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
</style>
</head>
<body>
<h1>CSS font-family</h1>
<p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
<p class="p2">This is a paragraph, shown in the Arial font.</p>
<p class="p3">This is a paragraph, shown in the Lucida Console font.</p>
</body>
</html>
```

# Output



## CSS font-family

This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.

This is a paragraph, shown in the Lucida Console font.

# What are Web Safe Fonts?

- Web safe fonts are fonts that are universally installed across all browsers and devices.

- **Fallback Fonts**

- However, there are no 100% completely web safe fonts. There is always a chance that a font is not found or is not installed properly.

- Therefore, it is very important to always use fallback fonts.

- This means that you should add a list of similar "backup fonts" in the font-family property. If the first font does not work, the browser will try the next one, and the next one, and so on. Always end the list with a generic font family name.

# Best Web Safe Fonts for HTML and CSS

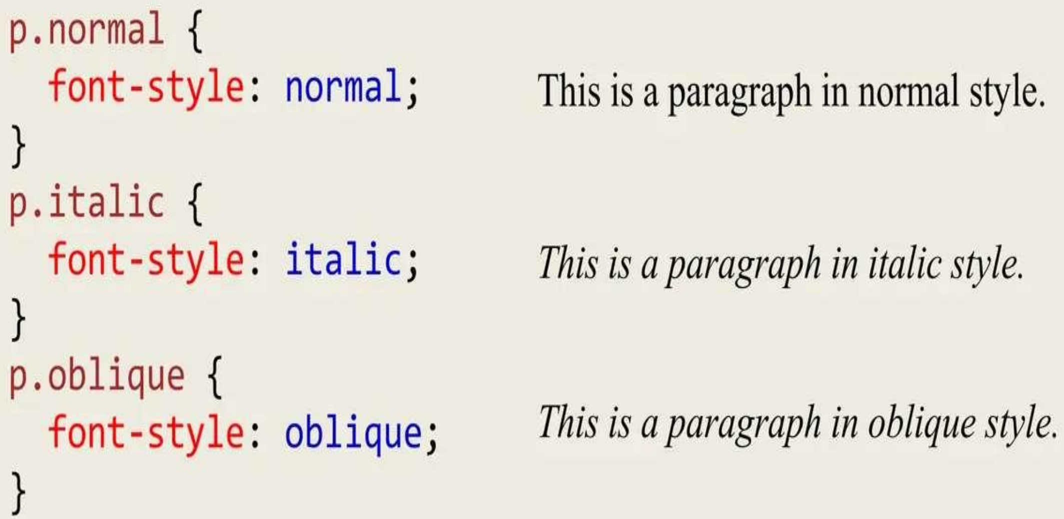
- The following list are the best web safe fonts for HTML and CSS:

- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

# CSS Font Style

- The font-style property is mostly used to specify italic text.

- This property has three values:
  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)



```
p.normal {
  font-style: normal;
}
p.italic {
  font-style: italic;
}
p.oblique {
  font-style: oblique;
}
```

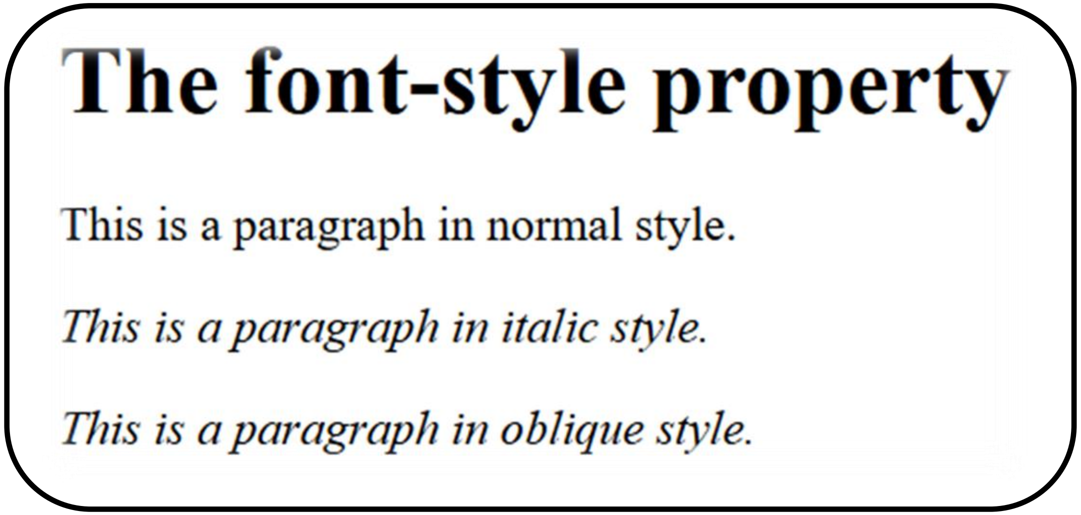
This is a paragraph in normal style.

*This is a paragraph in italic style.*

*This is a paragraph in oblique style.*

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-style: normal;
}
p.italic {
  font-style: italic;
}
p.oblique {
  font-style: oblique;
}
</style>
</head>
<body>
<h1>The font-style property</h1>
<p class="normal">This is a paragraph in normal style.</p>
<p class="italic">This is a paragraph in italic style.</p>
<p class="oblique">This is a paragraph in oblique style.</p>
</body>
</html>
```

# Output



**The font-style property**

This is a paragraph in normal style.

*This is a paragraph in italic style.*

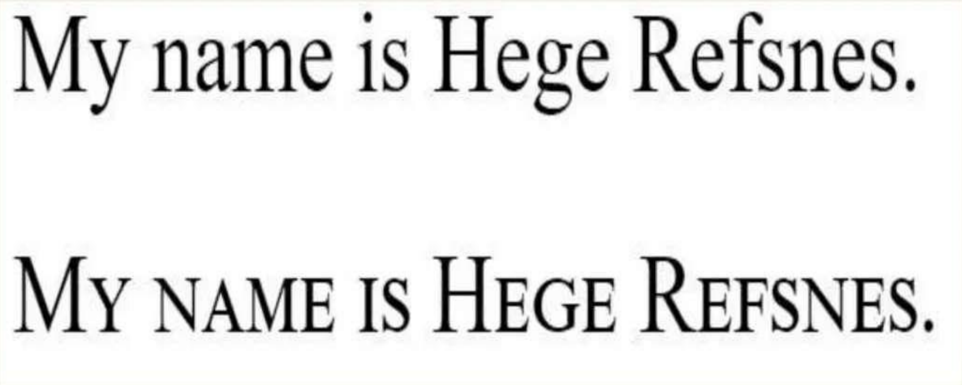
*This is a paragraph in oblique style.*

# Font Variant

- The font-variant property specifies whether or not a text should be displayed in a small-caps font.

- In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

```
p.normal {
  font-variant: normal;
}

p.small {
  font-variant: small-caps;
}
```



My name is Hege Refsnes.

MY NAME IS HEGE REFSNES.

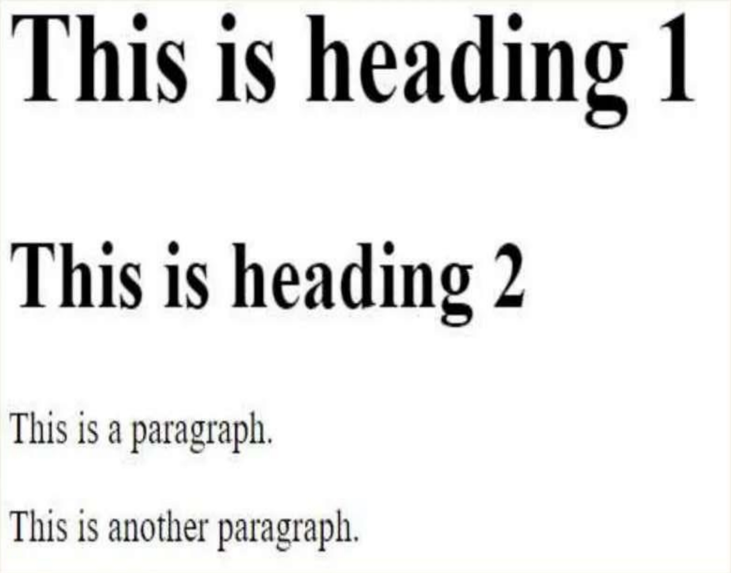
# Font Size

- The font-size property sets the size of the text.
- Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.
- Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.
- The font-size value can be an absolute, or relative size.
- **Absolute size:**
- Sets the text to a specified size
- Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
- Absolute size is useful when the physical size of the output is known
- **Relative size:**
- Sets the size relative to surrounding elements
- Allows a user to change the text size in browsers

# Set Font Size With Pixels

- Setting the text size with pixels gives you full control over the text size:

```
h1 {
  font-size: 40px;
}

h2 {
  font-size: 30px;
}

p {
  font-size: 14px;
}
```



# This is heading 1

## This is heading 2

This is a paragraph.

This is another paragraph.

# CSS Google Fonts

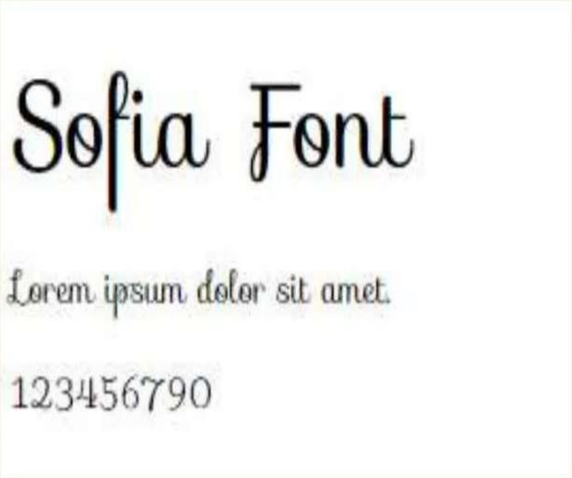
## Google Fonts

- If you do not want to use any of the standard fonts in HTML, you can use Google Fonts.
- Google Fonts are free to use, and have more than 1000 fonts to choose from.

## How To Use Google Fonts

- Just add a special style sheet link in the <head> section and then refer to the font in the CSS.

```
<head>
<link rel="stylesheet" href="https://fonts.goog
leapis.com/css?family=Sofia">
<style>
body {
  font-family: "Sofia", sans-serif;
}
</style>
</head>
```



Sofia Font

Lorem ipsum dolor sit amet.

123456790

# The CSS Font Property

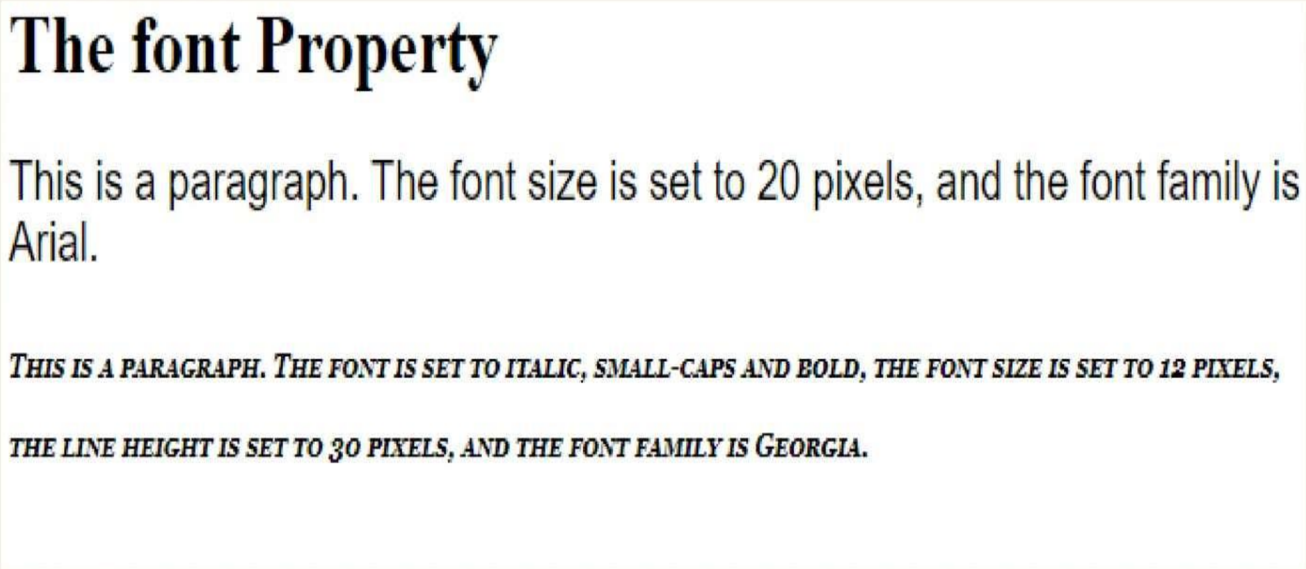
- To shorten the code, it is also possible to specify all the individual font properties in one property.

The font property is a shorthand property for:

- font-style
- font-variant
- font-weight
- font-size/line-height
- font-family

# Example of font shorthand property

-
  ```
  p.a {
    font: 20px Arial, sans-serif;
  }

  p.b {
    font: italic small-caps bold 12px/30px Georgia, serif;
  }
  ```



## The font Property

This is a paragraph. The font size is set to 20 pixels, and the font family is Arial.

*This is a paragraph. The font is set to italic, small-caps and bold, the font size is set to 12 pixels, the line height is set to 30 pixels, and the font family is Georgia.*

# CSS Margins

- The CSS margin properties are used to create space around elements, outside of any defined borders.
- With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).
- Margin - Individual Sides

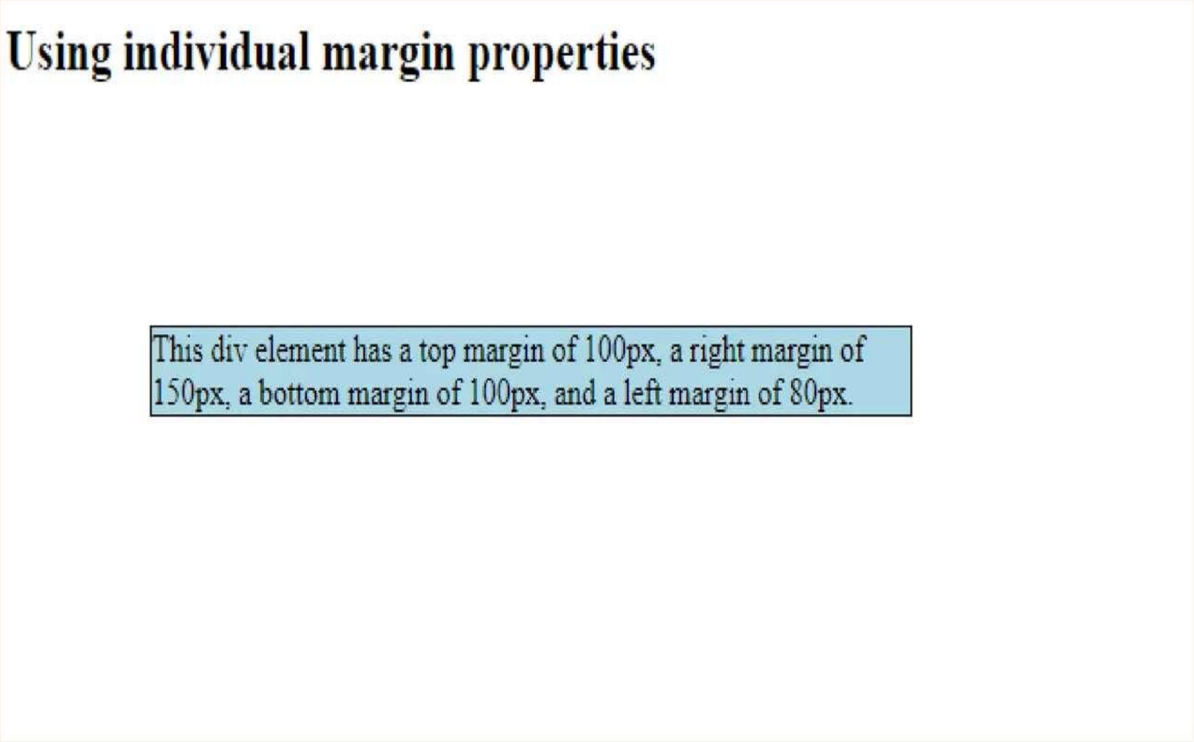
- CSS has properties for specifying the margin for each side of an element:
- margin-top
- margin-right
- margin-bottom
- margin-left

# margin properties

- All the margin properties can have the following values:

- auto - the browser calculates the margin
- length - specifies a margin in px, pt, cm, etc.
- % - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

```css
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```



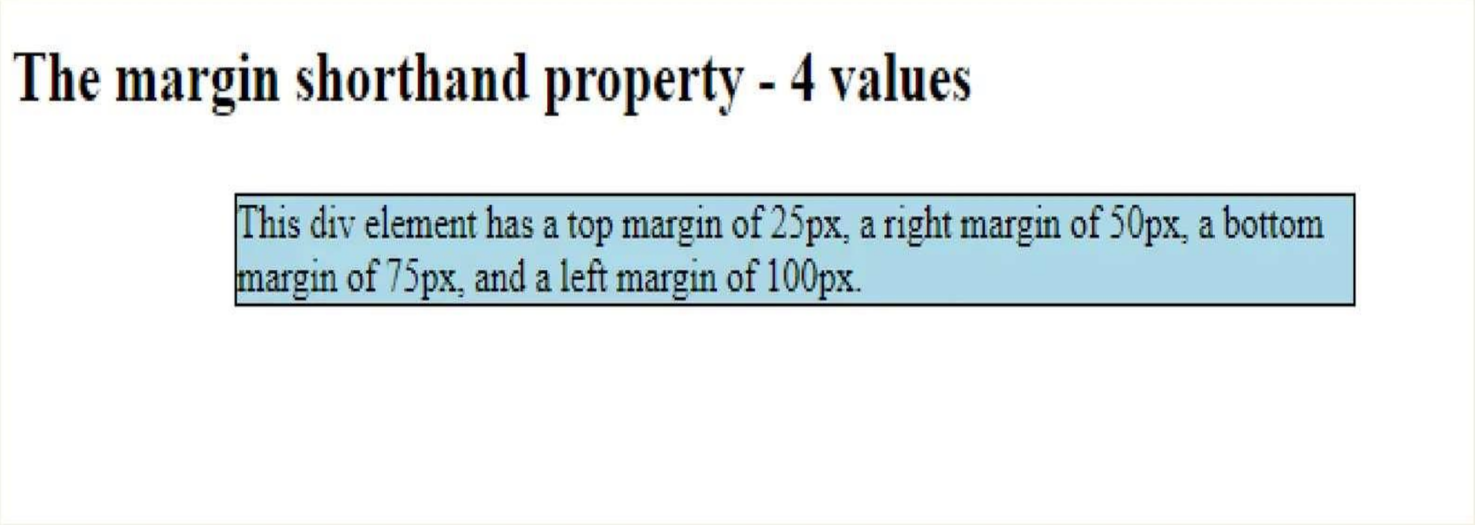
# Using individual margin properties

This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

# Margin - Shorthand Property

- To shorten the code, it is possible to specify all the margin properties in one property.

```
p {
  margin: 25px 50px 75px 100px;
}
```

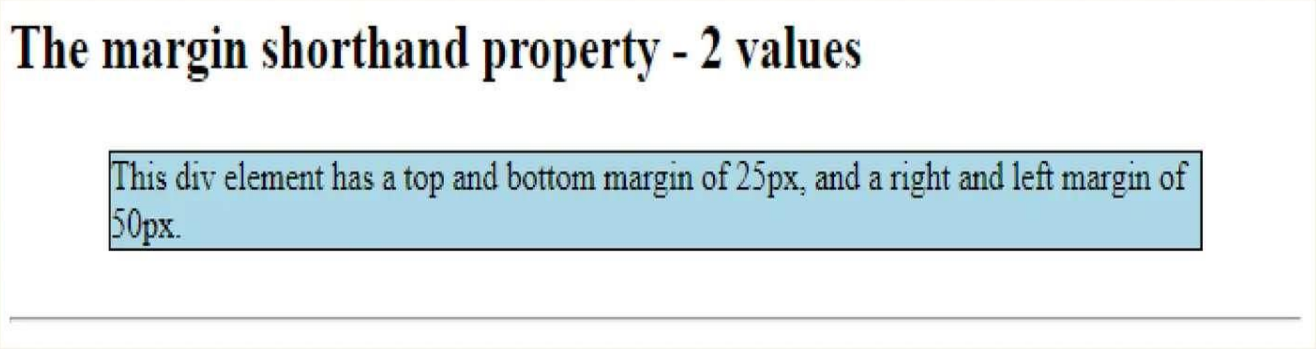


## The margin shorthand property - 4 values

This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.

- If the margin property has two values:
- margin: 25px 50px;
- top and bottom margins are 25px
- right and left margins are 50px

```
p {
  margin: 25px 50px;
}
```



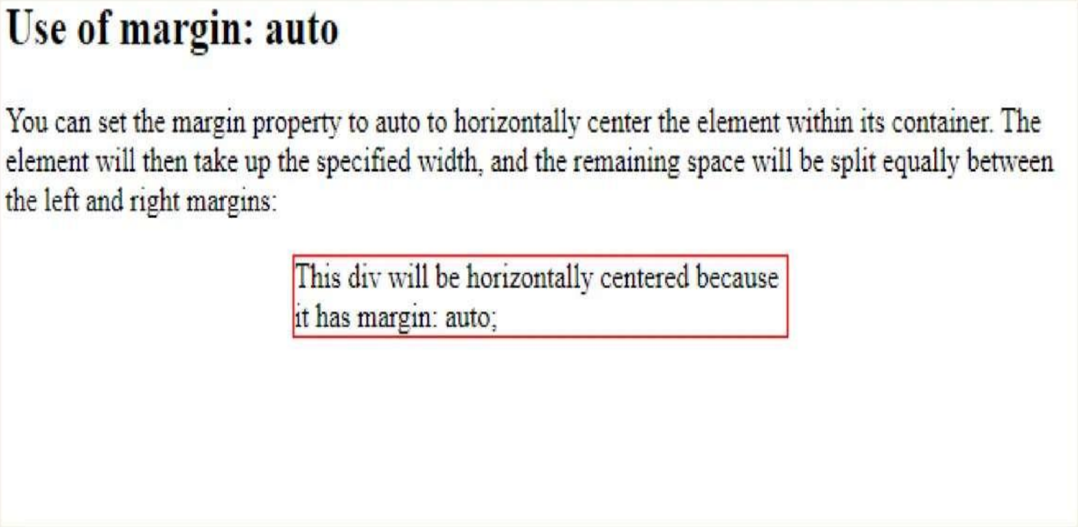
## The margin shorthand property - 2 values

This div element has a top and bottom margin of 25px, and a right and left margin of 50px.

# The auto Value of margin

- You can set the margin property to auto to horizontally center the element within its container.
- The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

```
div {
  width: 300px;
  margin: auto;
  border: 1px solid red;
}
```



**Use of margin: auto**

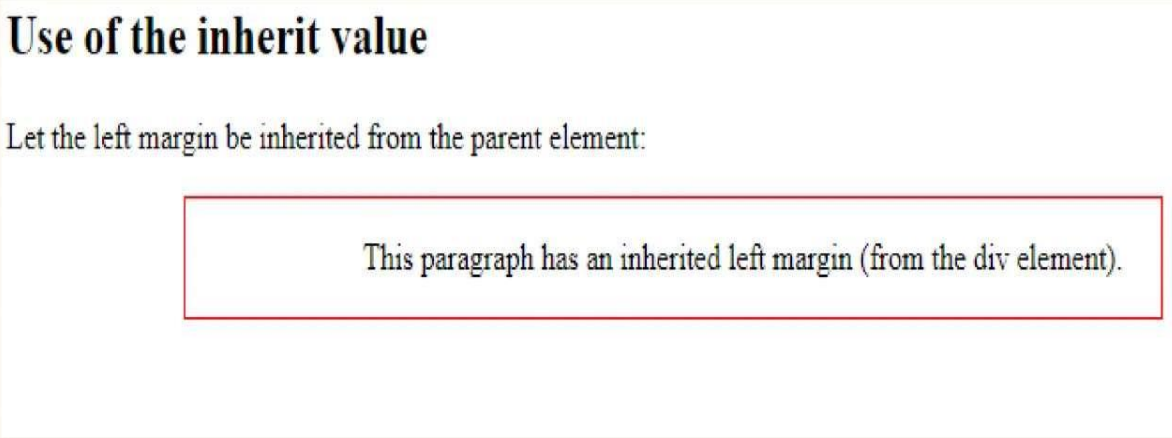
You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:

This div will be horizontally centered because it has margin: auto;

# The inherit Value

- This example lets the left margin of the <p class="ex1"> element be inherited from the parent element (<div>):

```
div {
  border: 1px solid red;
  margin-left: 100px;
}

p.ex1 {
  margin-left: inherit;
}
```

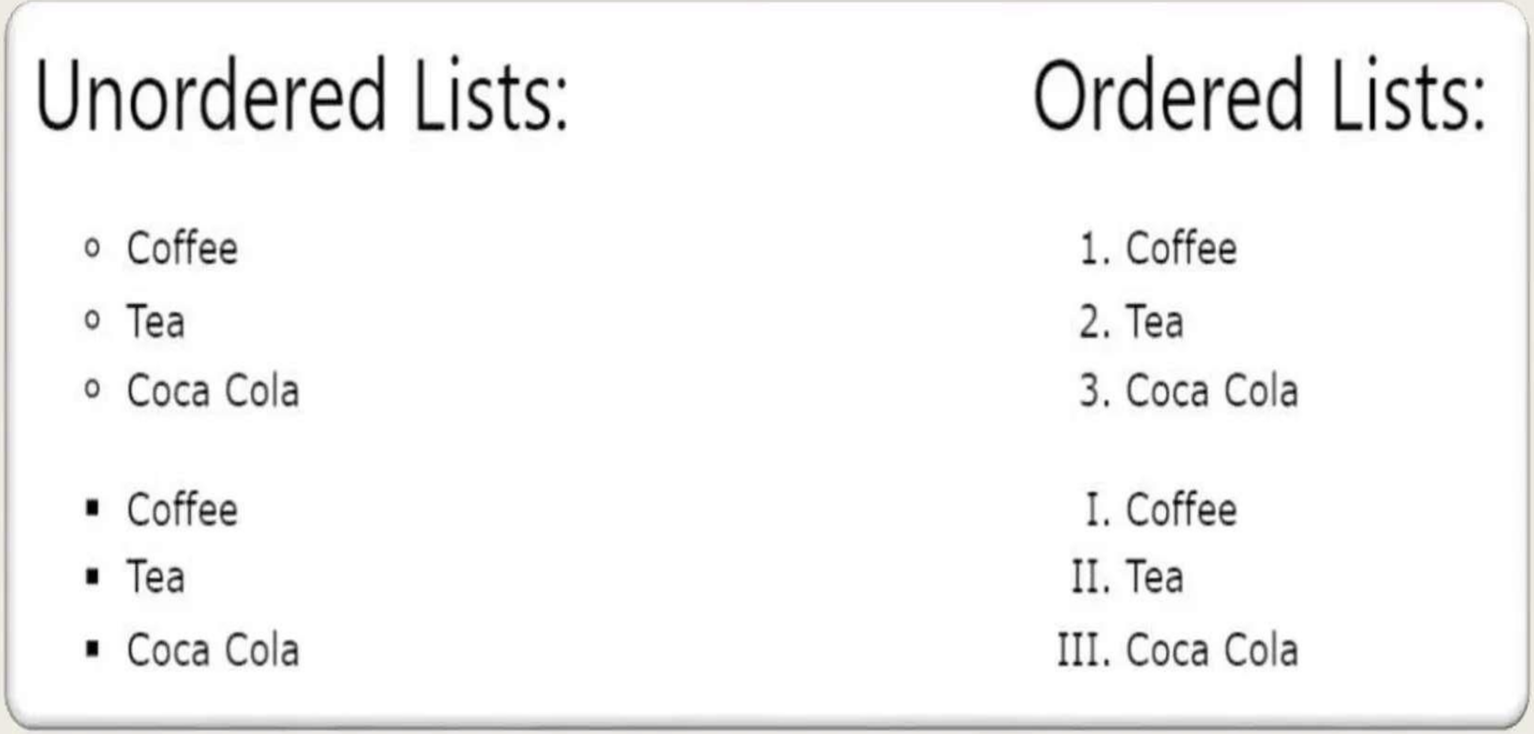


## Use of the inherit value

Let the left margin be inherited from the parent element:

This paragraph has an inherited left margin (from the div element).

# CSS Lists



Unordered Lists:

○ Coffee
○ Tea
○ Coca Cola

▪ Coffee
▪ Tea
▪ Coca Cola

Ordered Lists:

1. Coffee
2. Tea
3. Coca Cola

I. Coffee
II. Tea
III. Coca Cola

# CSS Lists

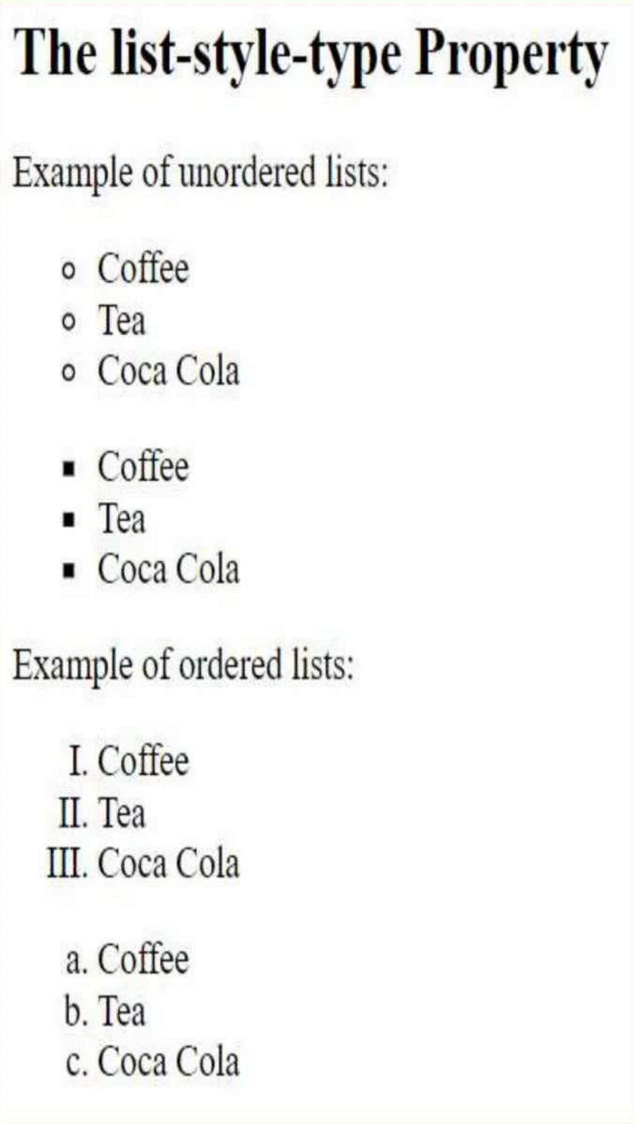
- HTML Lists and CSS List Properties
- In HTML, there are two main types of lists:
- unordered lists (<ul>) - the list items are marked with bullets
- ordered lists (<ol>) - the list items are marked with numbers or letters
- The CSS list properties allow you to:
- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

# Html Code of unordered lists and ordered lists

```
<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```
<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
```

```
ul.a {
  list-style-type: circle;
}

ul.b {
  list-style-type: square;
}

ol.c {
  list-style-type: upper-roman;
}

ol.d {
  list-style-type: lower-alpha;
}
```



# The list-style-type Property

Example of unordered lists:

○ Coffee
○ Tea
○ Coca Cola

▪ Coffee
▪ Tea
▪ Coca Cola

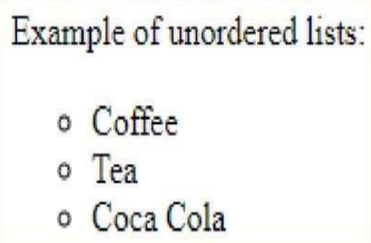
Example of ordered lists:

I. Coffee
II. Tea
III. Coca Cola

a. Coffee
b. Tea
c. Coca Cola

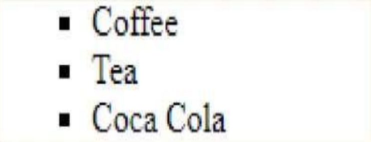
# Different List Item Markers

```
ul.a {
  list-style-type: circle;
}

ul.b {
  list-style-type: square;
}

ol.c {
  list-style-type: upper-roman;
}

ol.d {
  list-style-type: lower-alpha;
}
```

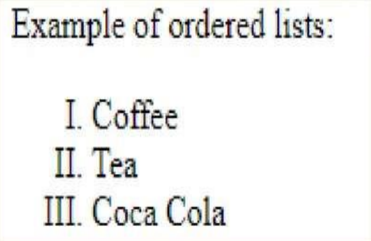


Example of unordered lists:

○ Coffee
○ Tea
○ Coca Cola

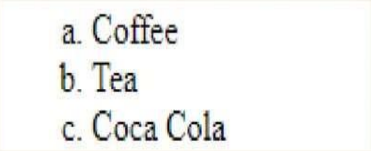


▪ Coffee
▪ Tea
▪ Coca Cola



Example of ordered lists:

I. Coffee
II. Tea
III. Coca Cola

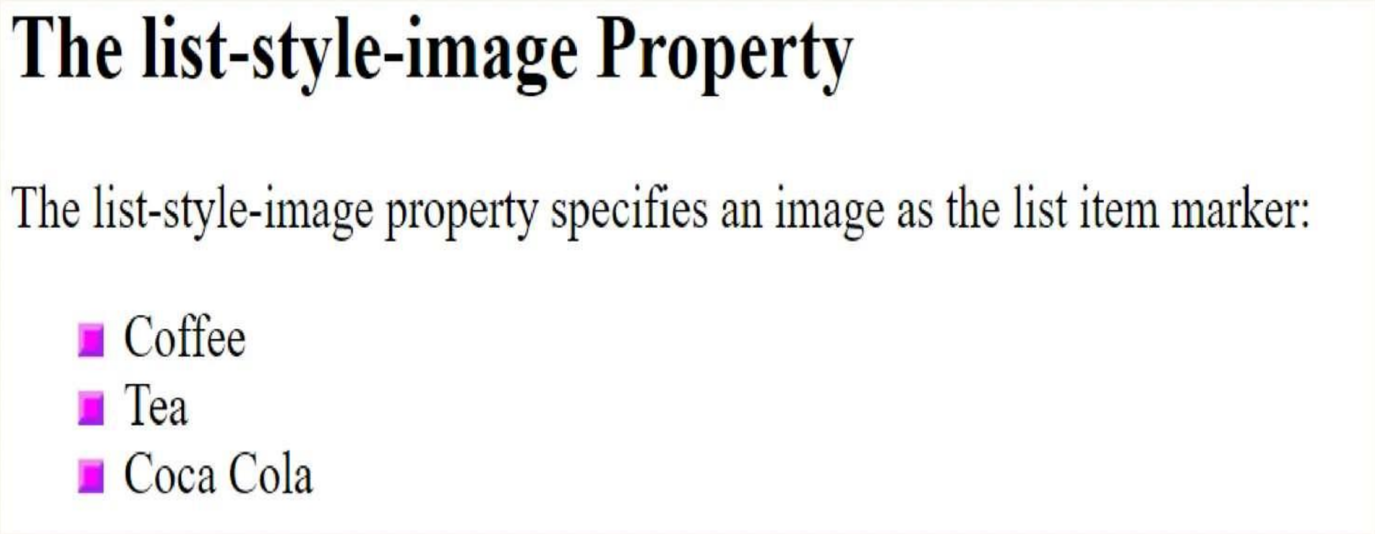


a. Coffee
b. Tea
c. Coca Cola

# An Image as The List Item Marker

- The list-style-image property specifies an image as the list item marker

-
  ```
  ul {
    list-style-image: url('sqpurple.gif');
  }
  ```



## The list-style-image Property

The list-style-image property specifies an image as the list item marker:

■ Coffee
■ Tea
■ Coca Cola

# Styling List With Colors

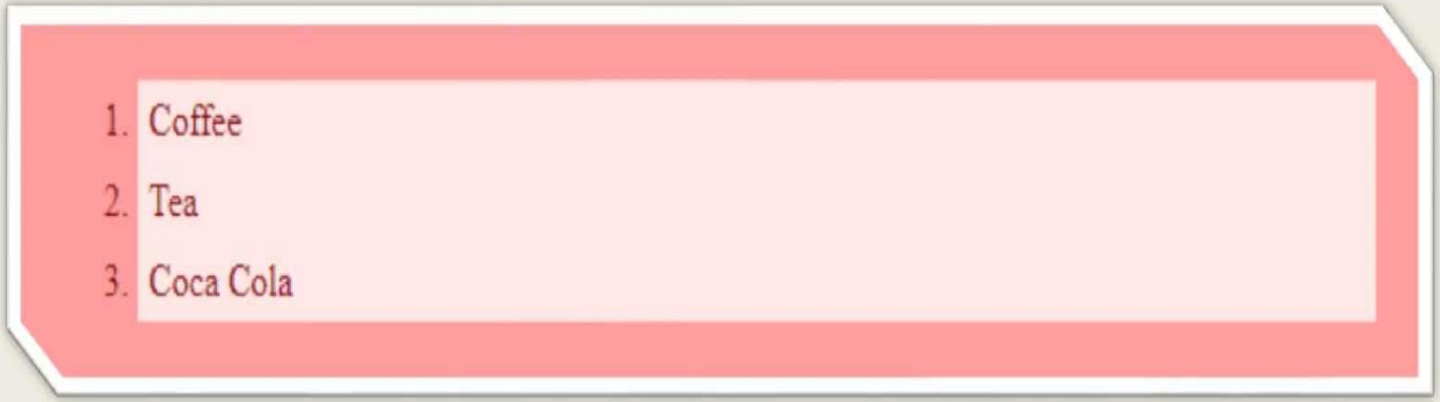
## CSS

-
  ```
  ol {
    background: #ff9999;
    padding: 20px;
  }
  ```

-
  ```
  ol li {
    background: #ffe5e5;
    color: darkred;
    padding: 5px;
    margin-left: 35px;
  }
  ```

## HTML

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
```



1. Coffee
2. Tea
3. Coca Cola

# Styling List With Colors

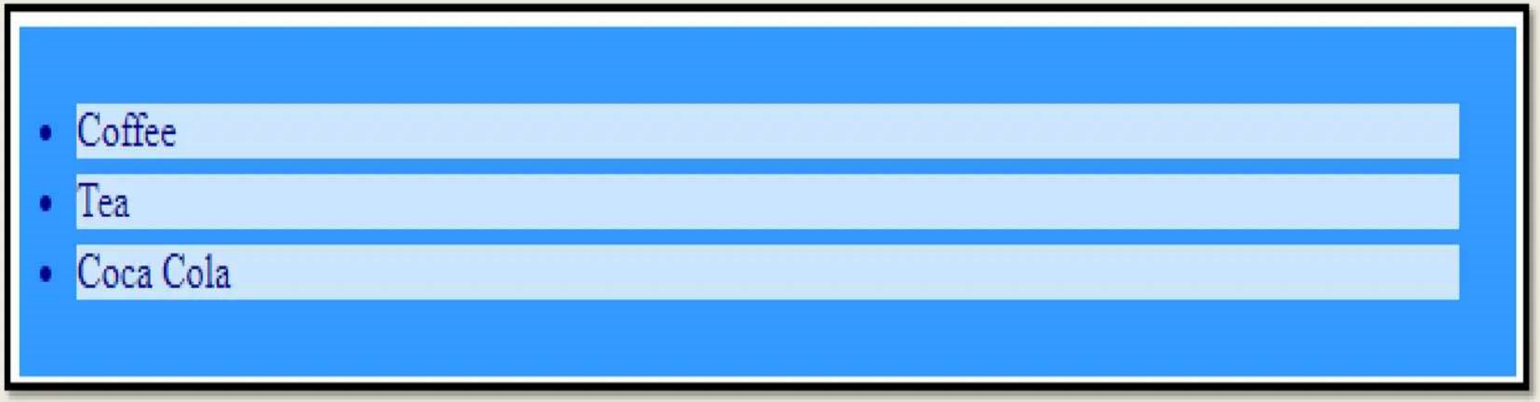
## CSS

-
  ```
  ul {
    background: #3399ff;
    padding: 20px;
  }
  ```

-
  ```
  ul li {
    background: #cce5ff;
    color: darkblue;
    margin: 5px;
  }
  ```

## HTML

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```



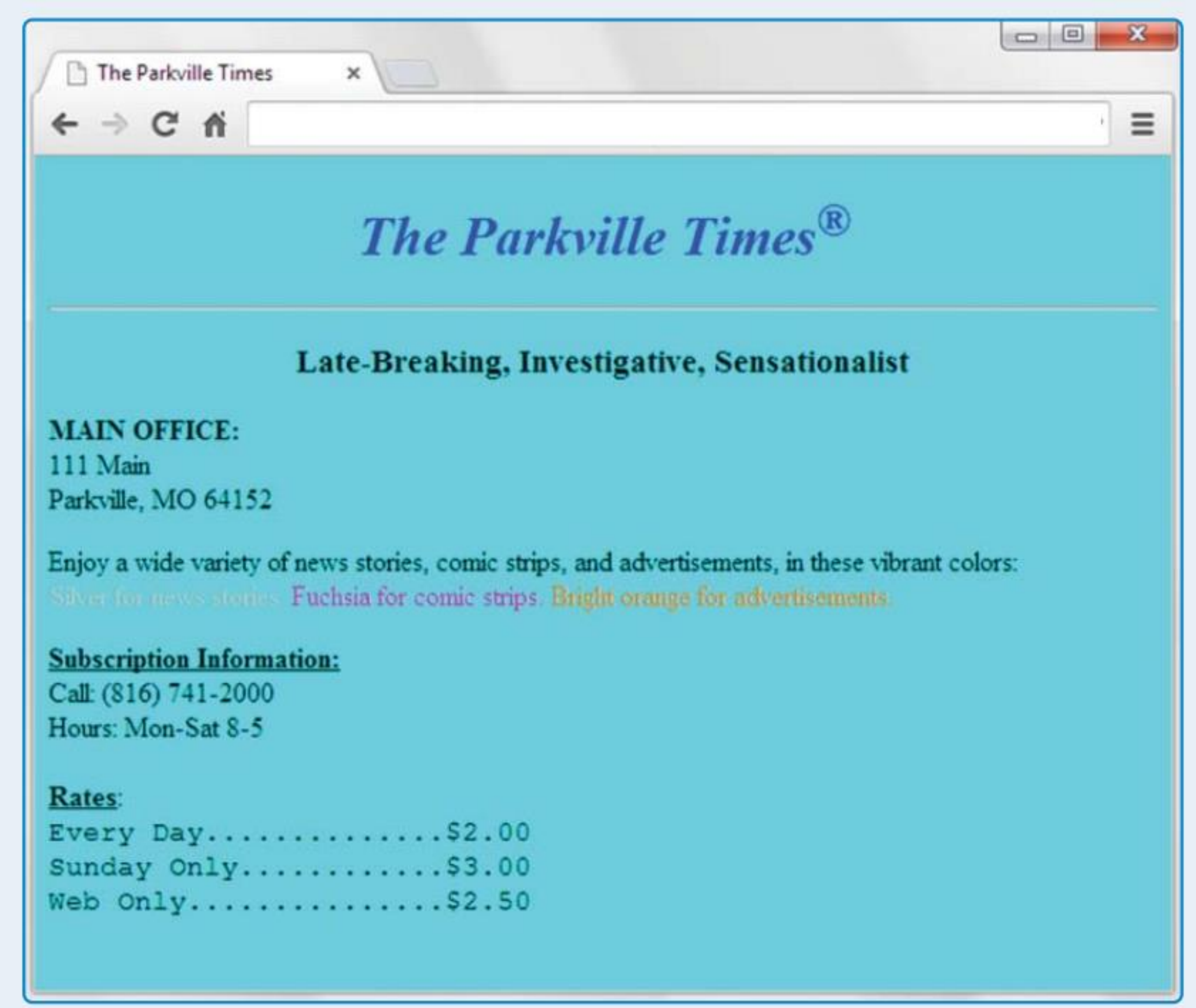
- Coffee
- Tea
- Coca Cola

# Think you

## Any questions ??

# Homework

Create the following web page, and name the file newspaper.html



## *The Parkville Times*®

### Late-Breaking, Investigative, Sensationalist

**MAIN OFFICE:**
111 Main
Parkville, MO 64152

Enjoy a wide variety of news stories, comic strips, and advertisements, in these vibrant colors:
Silver for news stories. Fuchsia for comic strips. Bright orange for advertisements.

**Subscription Information:**
Call: (816) 741-2000
Hours: Mon-Sat 8-5

Rates:

```
Every Day..............$2.00
Sunday Only............$3.00
Web Only...............$2.50
```