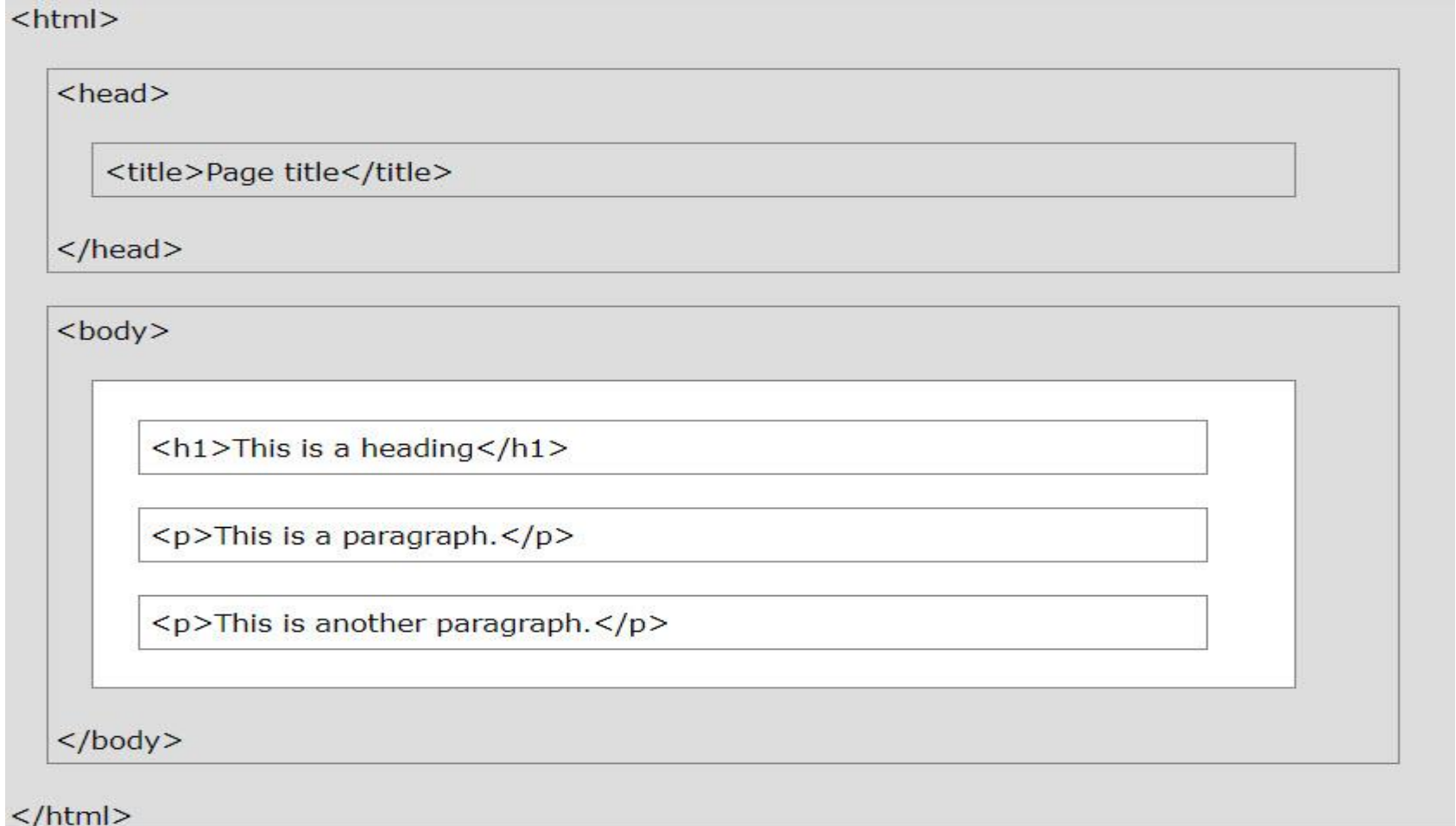# Objectives

◇ In this session, you will learn to:
- ◆ New elements in HTML 5
- ◆ Syntax of CSS

# Structure of HTML page

◇ Visualization of an HTML page structure:

```
<html>

    <head>

        <title>Page title</title>

    </head>

    <body>

            <h1>This is a heading</h1>

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

            <p>This is another paragraph.</p>

    </body>

</html>
```

# New elements in HTML5

◇ HTML 5 added the following form elements:
- ◆ <datalist>
- ◆ <output>
- ◆ Browsers do not display unknown elements.
- ◆ New elements that are not supported in older browsers will not "destroy" your web page.

- ◆ Datalist element
  - ◆ Specifies a list of pre-defined options for an <input> element.
  - ◆ Users will see a drop-down list of the pre-defined options as they input data.
  - ◆ The **list** attribute of the <input> element, must refer to the **id** attribute of the <datalist> element.

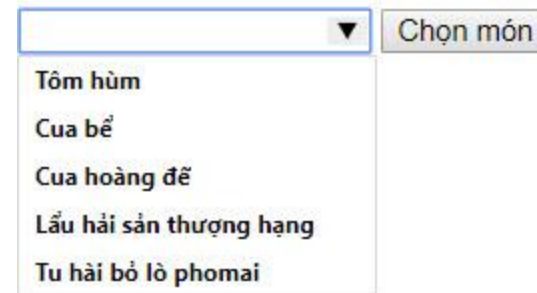# Datalist element

Example:

```html
<!DOCTYPE html>
<html>
<body>

<form action="/demo.html">
  <input list="listseafood" name="food">
  <datalist id="listseafood">
    <option value="Tôm hùm">
    <option value="Cua bể">
    <option value="Cua hoàng đế">
    <option value="Lẩu hải sản thượng hạng">
    <option value="Tu hài bỏ lò phomai">
  </datalist>
  <input type="submit" value="Chọn món">
</form>


</body>
</html>
```

Result:

# Output element

◇ <output> element

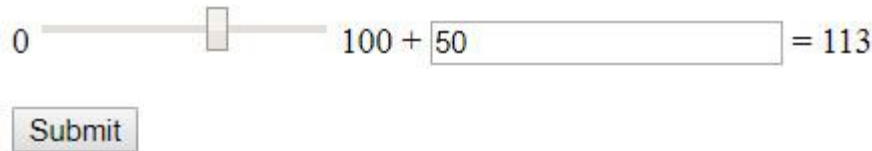   ◇ Represents the result of a calculation (like one performed by a script)

Example:

```html
<!DOCTYPE html>
<html>
<body>

<form action="/show.html"
oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>

</body>
</html>
```

Result:

0 ———————□———— 100 + [50          ] = 113

[ Submit ]

# HTML5 Input Types

- HTML 5 added several new input types:
    - Color
    - Date
    - Datetime-local
    - Email
    - Month
    - Number
    - Range
    - Search
    - Tel
    - Time
    - url
    - week

# Input Type Color

◇ Input Type Color
   ◆ Is used for input fields that should contain a color

Example:

Result:
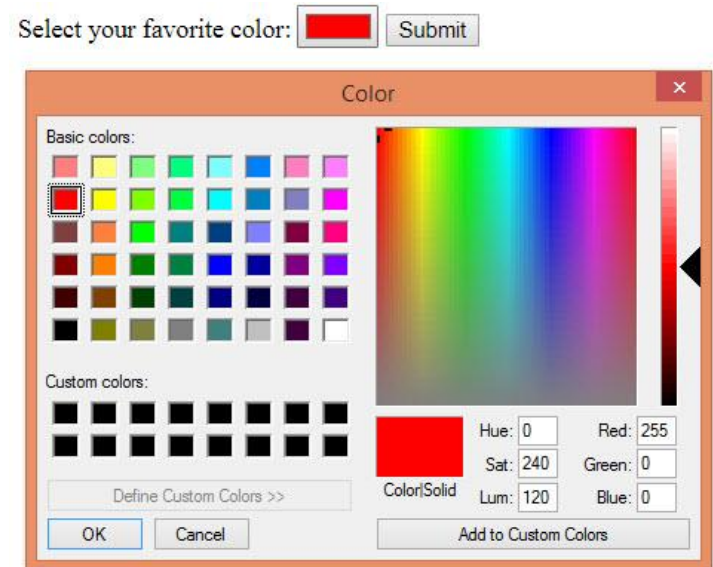
```
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  Select your favorite color:
  <input type="color" name="favcolor" value="#ff0000">
  <input type="submit">
</form>

</body>
</html>
```

# Input Type Date

◇ Input Type Date

◆ Is used for input fields that should contain a date.

Example:

Result:

```
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  Birthday:
  <input type="date" name="bday">
  <input type="submit">
</form>

</body>
</html>
```

Birthday: dd/mm/yyyy ▲▼ ▼ Submit

February 2018 ▼            ◄   ●   ►

| Mon | Tue | Wed | Thu | Fri | Sat | Sun |
|-----|-----|-----|-----|-----|-----|-----|
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 1 | 2 | 3 | 4 |

## ◇ Input Type Number

- ◇ Defines a **numeric** input field.
- ◇ You can also set restrictions on what numbers are accepted.

Example:                                                          Result:

```
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
  <input type="submit">
</form>

</body>
</html>
```

Quantity (between 1 and 5): 5      Submit

# ◇ Input Type Range

◇ Defines a control for entering a number whose exact value is not important (like a slider control).

◇ Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes

Example:                                          Result:

```
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php" method="get">
  Points:
  <input type="range" name="points" min="0" max="10">
  <input type="submit">
</form>

</body>
</html>
```

Points: [slider]  Submit

# Input restrictions

◇ List of some common input restrictions (some are new in HTML 5)

| Attribute | | Description |
|---|---|---|
| disabled | | Specifies that an input field should be disabled |
| max | 5 | Specifies the maximum value for an input field |
| maxlength | | Specifies the maximum number of character for an input field |
| min | 5 | Specifies the minimum value for an input field |
| pattern | 5 | Specifies a regular expression to check the input value against |
| readonly | | Specifies that an input field is read only (cannot be changed) |
| required | 5 | Specifies that an input field is required (must be filled out) |
| size | | Specifies the width (in characters) of an input field |
| step | 5 | Specifies the legal number intervals for an input field |
| value | | Specifies the default value for an input field |

# Validation HTML 5 form

◇ Teacher demo validation HTML5 form by using input restrictions associate with input types

# HTML5 Video/Audio

- Before HTML5, a video could only be played in a browser with a plug-in (like flash).
- The HTML5 <video> element specifies a standard way to embed a video in a web page.
- To show a video in HTML, use the **<video>** element
- To play an audio file in HTML, use the **<audio>** element
- The **controls** attribute adds video controls, like play, pause, and volume.
- It is a good idea to always include **width** and **height** attributes.
- The **<source>** element allows you to specify alternative video/audio files which the browser may choose from. The browser will use the first recognized format.

Example:

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```
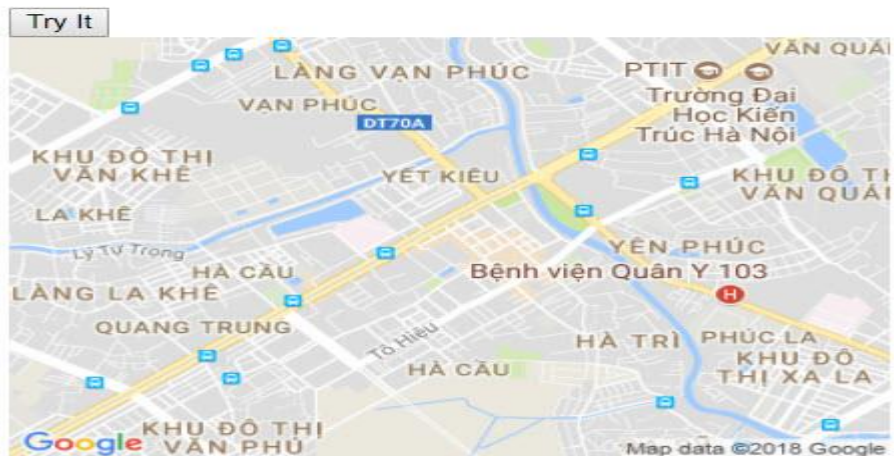
Result:

# HTML5 Geolocation

◇ The HTML Geolocation API is used to locate a user's position.

◇ The getCurrentPosition() method is used to return the user's position.

◆ To display the result in a map, you need access to a map service, like Google Maps.

Example:

```
function showPosition(position) {
    var latlon = position.coords.latitude + "," + position.coords.longitude;

    var img_url = "https://maps.googleapis.com/maps/api/staticmap?center=
    "+latlon+"&zoom=14&size=400x300&sensor=false&key=YOUR_:KEY";

    document.getElementById("mapholder").innerHTML = "<img src='"+img_url+"'>";
}
```

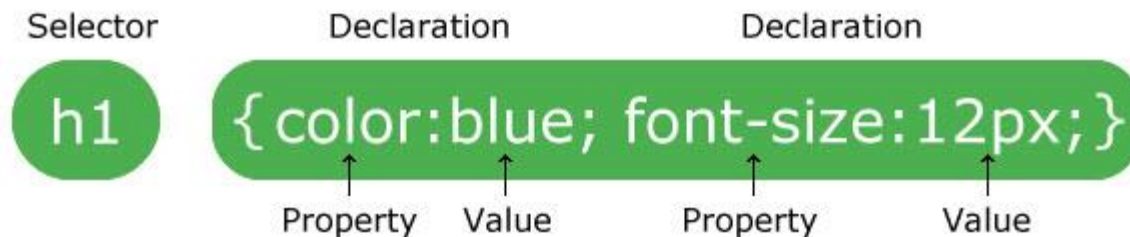Result:

# CSS

◇ Cascading Style Sheet (CSS):
- ◆ Is a collection of styles used to change the appearance of HTML elements on Web pages.
- ◆ Defines a set of standard rules that provide better control over the page layout and appearance.
- ◆ Is used to enforce consistent display standards across all the Web pages of the website.

# Identifying the Syntax of CSS

◇ CSS allows the creation of one or more rules for defining the style characteristics of HTML elements.

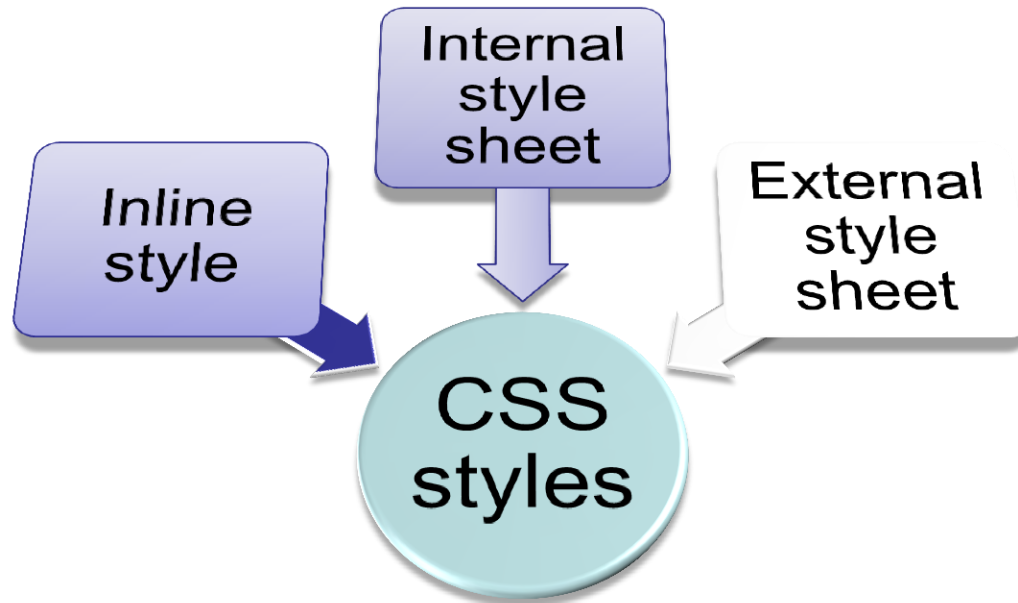◇ Each CSS rule consists of the following parts:

Selector

Declaration Block



Selector    Declaration    Declaration

h1  { color:blue; font-size:12px;}

Property  Value    Property    Value

# Identifying the Syntax of CSS (Contd.)

◇ You can use the following syntax to define a CSS rule:

```
selector
{
//Declaration Block
property: value;
}
```

◇ CSS styles can be broadly classified into the following types:

# Identifying CSS Selectors

- The user-defined CSS selectors can be classified into the following categories:
  - 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.
  - 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.
  - Type Selector
    - Selects elements based on the element name
  - Universal Selector
    - Define base style for all elements of web page

# Identifying CSS Selectors

- Teacher demo declaring CSS in HTML  page
  - Demo CSS types: Inline, internal, external
  - Demo CSS Selector: ID, class, type, universal

# Summary

- New elements in HTML 5
- Syntax of CSS