COMP S380F Web Applications: Design and Development
Lab 1: HTML5, CSS & JavaScript

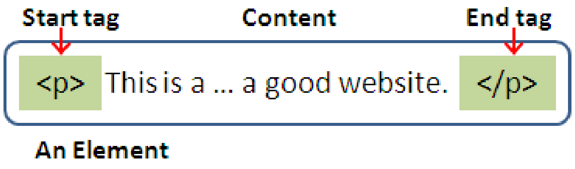
In this lab, we have some hands-on exercises on HTML5, CSS and JavaScript. We will build a ticket booking page for the Open Park of Hong Kong. The lab materials are provided in the course homepage.

Task 1: HTML5

HTML is a markup language used for structuring and presenting content on the World Wide Web (WWW). It is developed by Tim Berners Lee at CERN in 1991. The latest version is ***HTML5***, which is now supported in most web browsers. A HTML tutorial can be found in https://www.w3schools.com.

The following is a simple HTML5 code, where the content is marked up by tags.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Page Title</title>
</head>
<body>
  <h1>This is a Heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```



The table below shows some common HTML tags. A tag may have attributes, e.g., href in <a href = "URL">. The HTML ***attribute values*** must be enclosed in ***double quotes***.

Tag	Marked-up Content
<h1>heading</h1>	Heading (**<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>**, **<h6>** are available for different heading levels).
<p>paragraph</p>	Paragraph
<br />	Line break
<hr />	Horizontal rule

<ol>
  <li>1st list item</li>
  <li>2nd list item</li>
</ol>
	Ordered list: **<ol>**
	(Unordered list: **<ul>**)
	List item: **<li>**

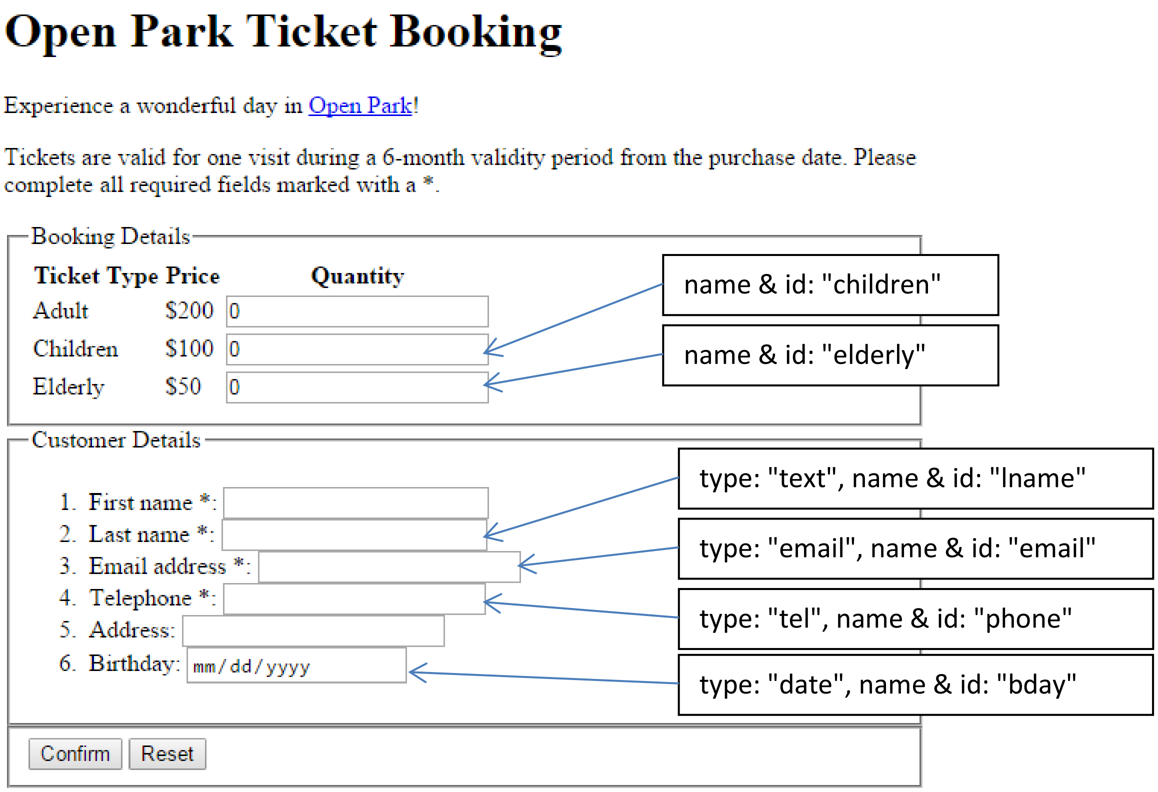
<img src="logo.url" />	Image; the attribute **src** specifies the image file location.
<a href="about_us.html">About us </a>	Anchor for hyperlink; **href** specifies the location of the resource, which can be a ***relative URL*** (e.g., about_us.html) or ***full URL*** (e.g., http://www.abc.com/about_us.html).
<form action="purchase" method="post"> … components of the form </form>	• **HTML form** allows users to send information to a web application. • **action** is the program to be executed by the server when the form data is submitted; it can be in full or relative URL. • The **method** attribute specifies the HTTP method of the request from the client to the server.

<input type=... name=... id=... value=... />	A form component for getting user input. • **type**: specifies what type of control is used, e.g., “text”, “email”, “date”, “submit”, “reset”. • **name**: used by the server to process the data. • **id**: assigned for a particular HTML element and can also be used in conjunction with a **<label>** tag. • **value**: default value for the input control.

<fieldset>
  <legend>Booking Details</legend>
   ... form elements
</fieldset>
<fieldset>
  <legend>Delivery Address</legend>
  ... form elemenets
</fieldset>
	• **<fieldset>** is used inside a form to group form elements into different sections.
	• **<legend>** gives the section a name.

<table>
  <tr>
    <th>Header1</th>
    <th>Header2</th>
  </tr>
  <tr>
    <td>Data1</td>
    <td>Data2</td>
 </tr>
</table>
	Table: **<table>**
	Table row: **<tr>**
	Table header: **<th>**
	Table data: **<td>**

Your task: Understand the HTML code of ticket.html and update it to create the following page.



Open Park Ticket Booking

Experience a wonderful day in Open Park!

Tickets are valid for one visit during a 6-month validity period from the purchase date. Please complete all required fields marked with a *.

Booking Details

Ticket Type	Price	Quantity
Adult	$200	0
Children	$100	0
Elderly	$50	0

Customer Details

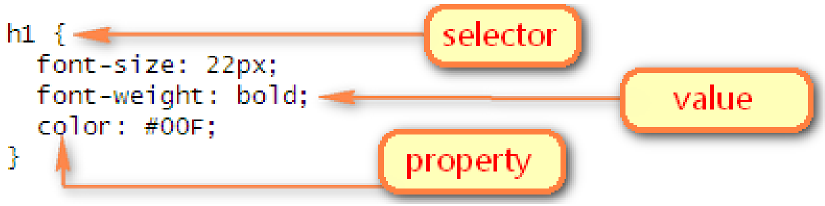
1. First name *:
2. Last name *:
3. Email address *:
4. Telephone *:
5. Address:
6. Birthday: mm/dd/yyyy

name & id: "children"
name & id: "elderly"
type: "text", name & id: "lname"
type: "email", name & id: "email"
type: "tel", name & id: "phone"
type: "date", name & id: "bday"
Confirm
Reset

Task 2: CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. A CSS tutorial can be found in https://www.w3schools.com.

Style sheets contain ***rules***. Each rule is a formatting instruction that applies to a part of your web page. A rule contains a ***selector*** and a set of ***property-value pairs***.



```
h1 {
  font-size: 22px;
  font-weight: bold;
  color: #00F;
}
```

The selector declares which part of the markup a style applies to. It can be

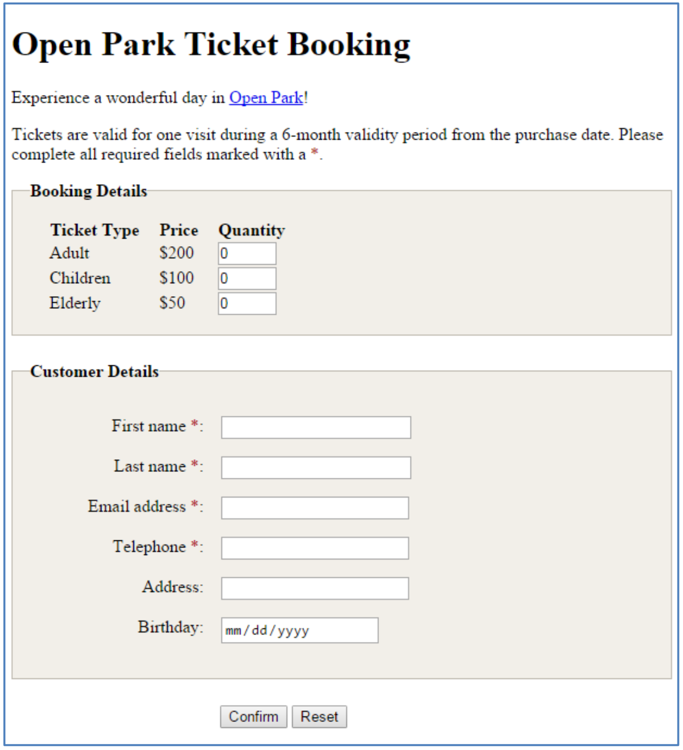
- an **HTML tag**
- **#id_name** (the ID of a HTML element, which cannot be reused for another HTML element)
- **.class_name** (the class of HTML elements, which can be used for a set of different HTML elements)

There are three ways to apply CSS to HTML:

- ***In-line***: inserts style sheet directly inside an HTML element.
- ***Embedded***: embeds an internal style sheet inside an HTML document using the “style” tag.
- ***External***: stores an external style sheet in a separate file.

Your task: Understand the CSS file style.css. Then, apply it to ticket.html by modifying the HTML code:

1. Add the CSS file style.css to the working directory.
2. Add the following tag inside the head tag:
   ```
   <link rel="stylesheet" type="text/css" href="style.css"/>
   ```
3. Replace all asterisks (i.e., *) with the following code:
   ```
   <span class="star">*</span>
   ```



Open Park Ticket Booking

Experience a wonderful day in Open Park!

Tickets are valid for one visit during a 6-month validity period from the purchase date. Please complete all required fields marked with a *.

Booking Details

Ticket Type	Price	Quantity
Adult	$200	0
Children	$100	0
Elderly	$50	0

Customer Details

First name *:

Last name *:

Email address *:

Telephone *:

Address:

Birthday: mm/dd/yyyy

Task 3: HTML5 Form Validation

Before HTML5 appears, form validation relies on JavaScript, which will be covered in Task 4. HTML5 introduces a number of new attributes, input types and other elements to HTML, and now some input validation can be done purely in HTML5 and CSS.

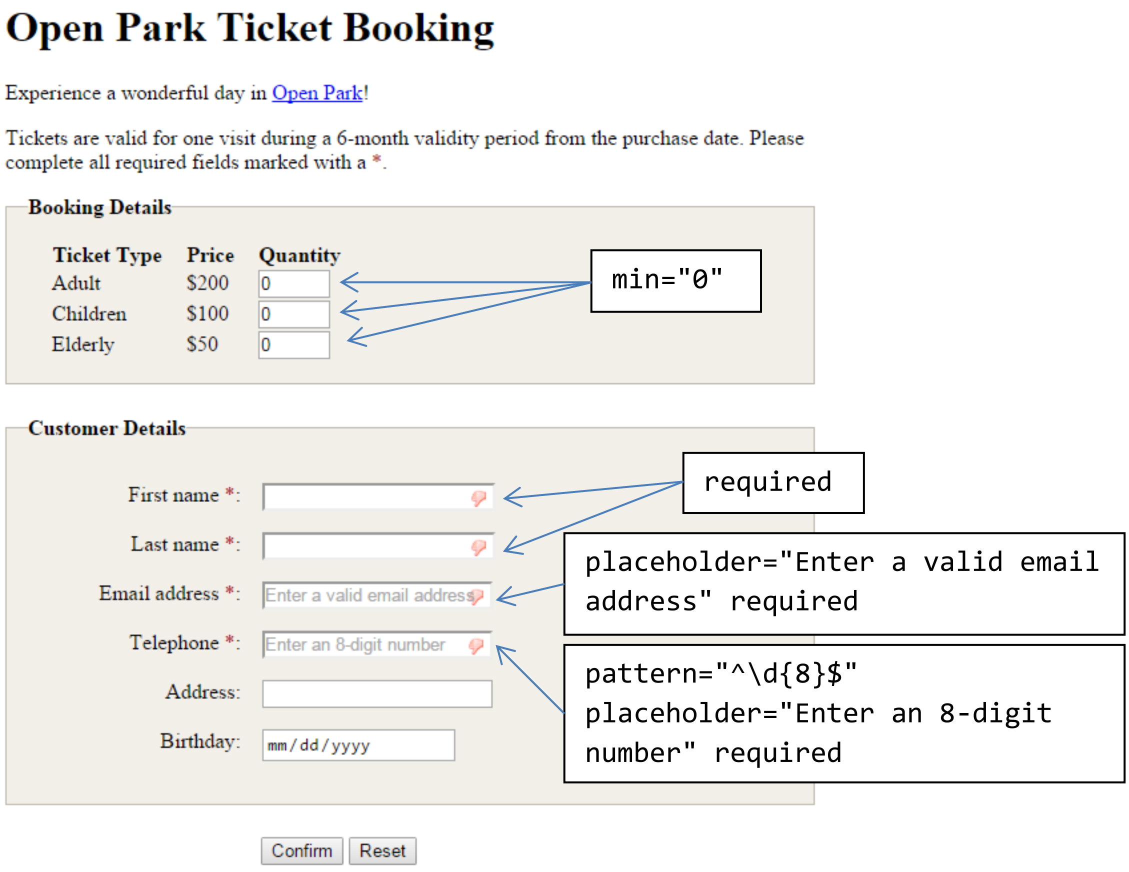
Your task: We want to make sure that the customer has filled in all the required fields and, more importantly, in the correct format. Follow the following steps:

1. Add the folder “images” (which contains the two images “valid.png” and “invalid.png”) to your working directory.
2. Add the following CSS code to the bottom of style.css:

```
input:required:invalid, input:focus:invalid {
  background-image: url(images/invalid.png);
  background-position: right top;
  background-repeat: no-repeat;
}

input:required:valid {
  background-image: url(images/valid.png);
  background-position: right top;
  background-repeat: no-repeat;
}
```

3. Add the following attributes to the input elements:



Open Park Ticket Booking

Experience a wonderful day in Open Park!

Tickets are valid for one visit during a 6-month validity period from the purchase date. Please complete all required fields marked with a *.

Booking Details

Ticket Type	Price	Quantity
Adult	$200	0
Children	$100	0
Elderly	$50	0

Customer Details

First name *:

Last name *:

Email address *: Enter a valid email address

Telephone *: Enter an 8-digit number

Address:

Birthday: mm / dd / yyyy

min="0"

required

placeholder="Enter a valid email address" required

pattern="^\d{8}$" placeholder="Enter an 8-digit number" required

Task 4: JavaScript

JavaScript is a scripting language that can run inside web browsers. It can make web pages more dynamic and interactive, e.g., creating pop-up windows, interactive menus and mouse events. It can make use of the ***Document Object Model (DOM)*** to access an HTML document as a hierarchy of nodes or objects.

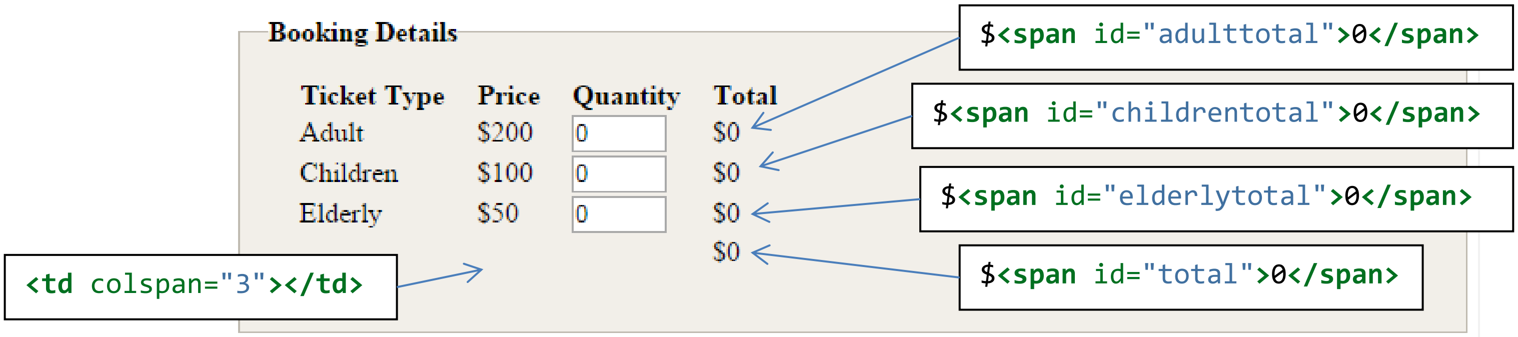
JavaScript has the following characteristics:

- ***Loosely-typed***: When we declare a variable, we only have to declare its name without a type.
- ***Object-based***: It has different built-in objects, and some DOM objects, e.g., window, document.
- ***Event-driven***: Programs can respond to user interface actions (e.g., mouse movement, click, keystroke).

Your task: We want to display the total price for each ticket type and the overall total price. We also want to make sure that the customer has booked at least 1 ticket before submitting the form.

Follow the following steps:

1. Add the JavaScript file myscript.js to the working directory.
2. Add a new column of booking details in the HTML code, as follows:



Booking Details

Ticket Type	Price	Quantity	Total
Adult	$200	0	$0
Children	$100	0	$0
Elderly	$50	0	$0
			$0

```
$<span id="adulttotal">0</span>
```

```
$<span id="childrentotal">0</span>
```

```
$<span id="elderlytotal">0</span>
```

```
$<span id="total">0</span>
```

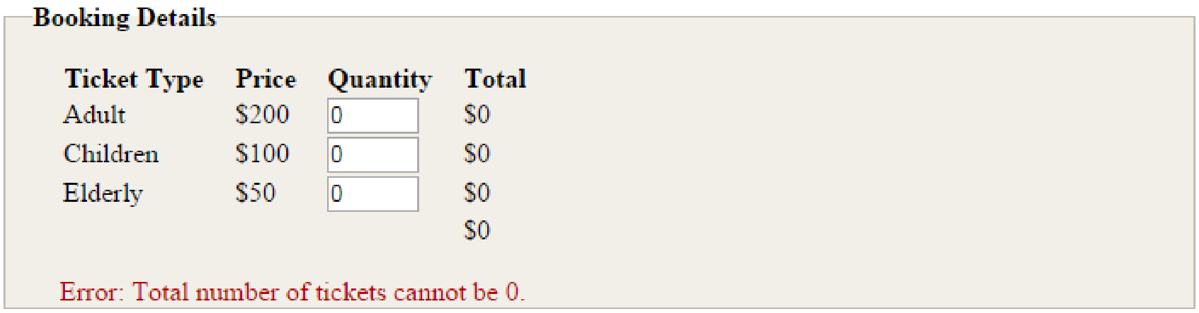
```
<td colspan="3"></td>
```

3. After the table's end tag, add the following HTML code for error message display:

```
<span id="error_msg"></span>
```

4. Add the CSS rule for the error message, which is shown when we click the submit button but no ticket is booked:

```
#error_msg {
    margin-left: 2em;
    color: #b30000;
}
```



Booking Details

Ticket Type	Price	Quantity	Total
Adult	$200	0	$0
Children	$100	0	$0
Elderly	$50	0	$0
			$0

Error: Total number of tickets cannot be 0.

5. In the start tag of “form”, add the following attribute for validating the number of booked tickets: `onsubmit="return validator()"`
6. After the body’s end tag, add **`<script src="myscript.js"></script>`**.
   - In the JavaScript DOM, we can use `document.getElementById("adult")` to access an HTML element with ID **adult**.
   - As there is only one form in the HTML page, we can access the form using `document.forms[0]`.
   - We can use **`addEventListener`** to associate a function with an event of the HTML element such that when the event appears, the function is executed.
   - The JavaScript code does not work if we add the **`<script>`** tag before the body tag, as the HTML element are not yet loaded when the code is run.

Appendix: Source code of myscript.js

```
var numType = 3;
var types = ["adult", "children", "elderly"];
var prices = [200, 100, 50];
var totals = [0, 0, 0];

var total = document.getElementById("total");
var errMsg = document.getElementById("error_msg");

function fn(t) {
    totals[t] = parseInt(document.getElementById(types[t]).value) * prices[t];
    document.getElementById(types[t] + "total").innerHTML = totals[t];
    total.innerHTML = totals[0] + totals[1] + totals[2];
    errMsg.innerHTML = "";
}

document.getElementById("adult").addEventListener("input", function() {fn(0);});
document.getElementById("children").addEventListener("input", function() {fn(1);});
document.getElementById("elderly").addEventListener("input", function() {fn(2);});

document.forms[0].addEventListener("reset", resetHandler);

function resetHandler(evt) {
    for (t = 0; t < numType; t++) {
        totals[t] = 0;
        document.getElementById(types[t] + "total").innerHTML = 0;
    }
    total.innerHTML = 0;
    errMsg.innerHTML = "";
}

function validator(){
    if (totals[0] + totals[1] + totals[2] == 0) {
        errMsg.innerHTML = "Error: Total number of tickets cannot be 0.";
        return false;
    } else
        return true;
}
```