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

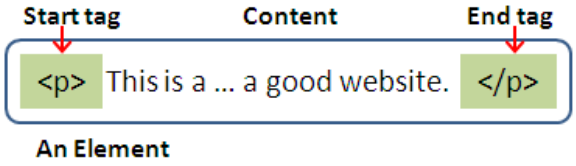
In this lab session, we will build up a ticket booking page for the Open Park of Hong Kong. Throughout the process, we will have some hands-on exercises on HTML5, CSS and Javascript. The lab materials are provided in the Wikicourse page.

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 of the web browsers.

This is a simple example of HTML5 code. 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>
```

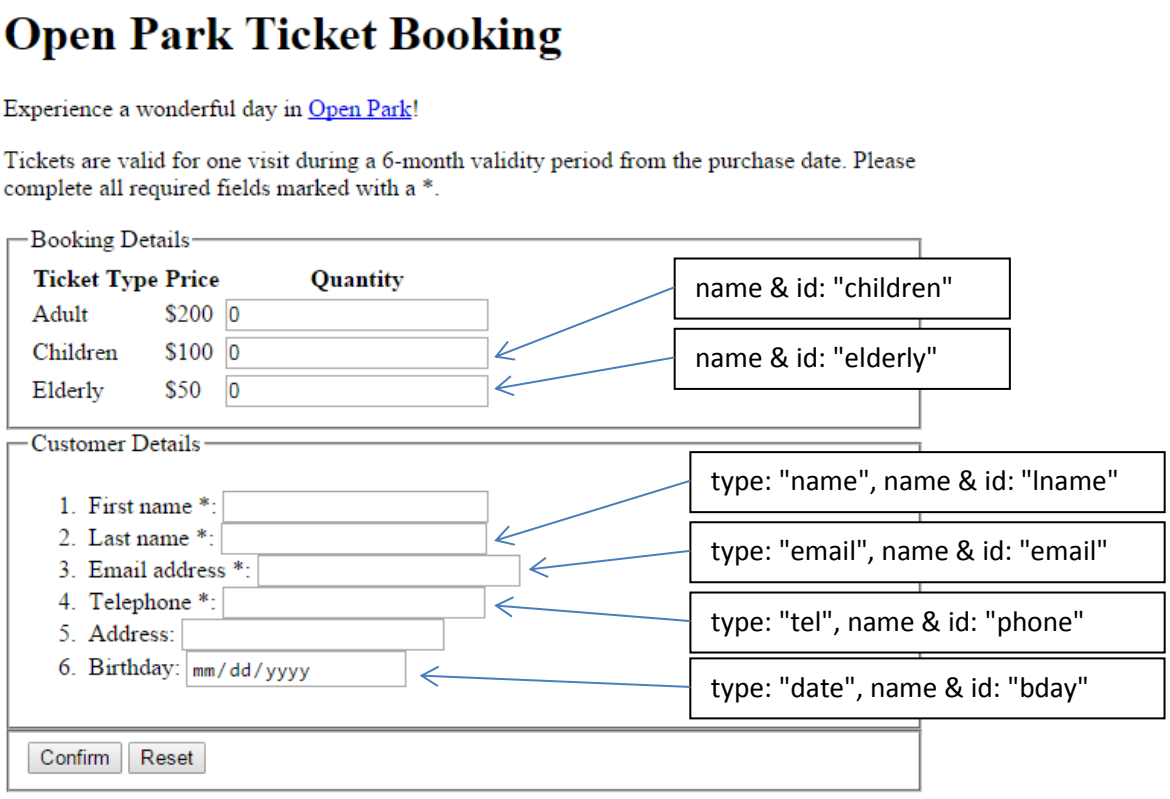


Below is a list of some common HTML tags. Some tag may have attributes, e.g., href in <a href = "URL">.

Tag	Marked-up Content
heading	Heading (h1, h2, …, h6 are available for different heading levels).
paragraph	Paragraph
 	Line break
<hr />	Horizontal rule
   1st list item 2nd list item 	Ordered list: ol (Unordered list: ul) List item: li
	Image; the attribute src specifies the image file location.
About us 	Anchor for link; href specifies the location of the resource to link, which can be in 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>	• Form, which allows users to send information to a web application. • "action" is the program to be executed by the server when the form's data is submitted; it can be in full or relative URL. • The specified "method" attribute enable an HTTP "POST" request to be made 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., “name”, “email”, “text”, “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

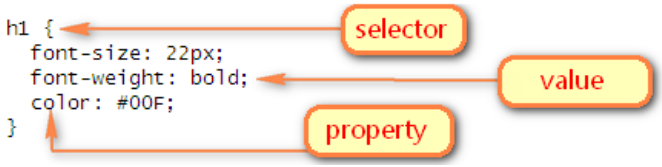
Your task: Understand the HTML code of ticket.html and update it to create the following page.



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.

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.



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

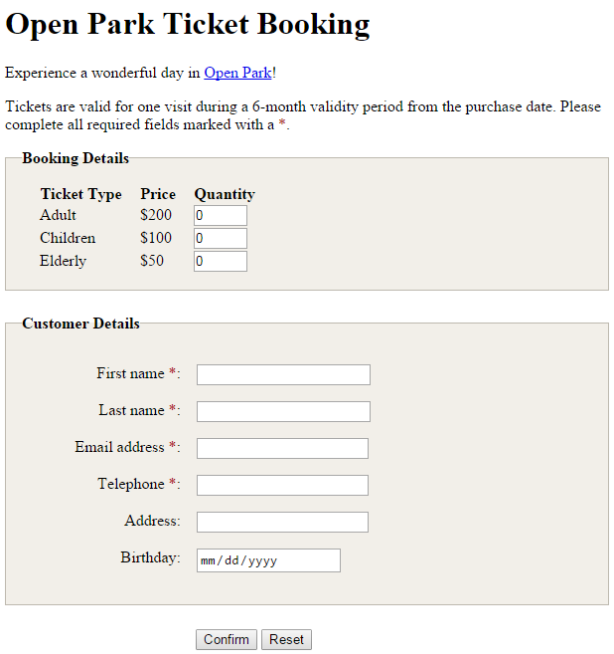
- an HTML tag
- .id_name (the ID of a particular HTML element and cannot be reused for another HTML element)
- #class_name (the class of HTML elements and can be used for a set of different HTML elements).

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:
   *



Task 3: HTML5 Form Validation

Before HTML5 appears, form validation relies on javascript, which we will encounter 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.

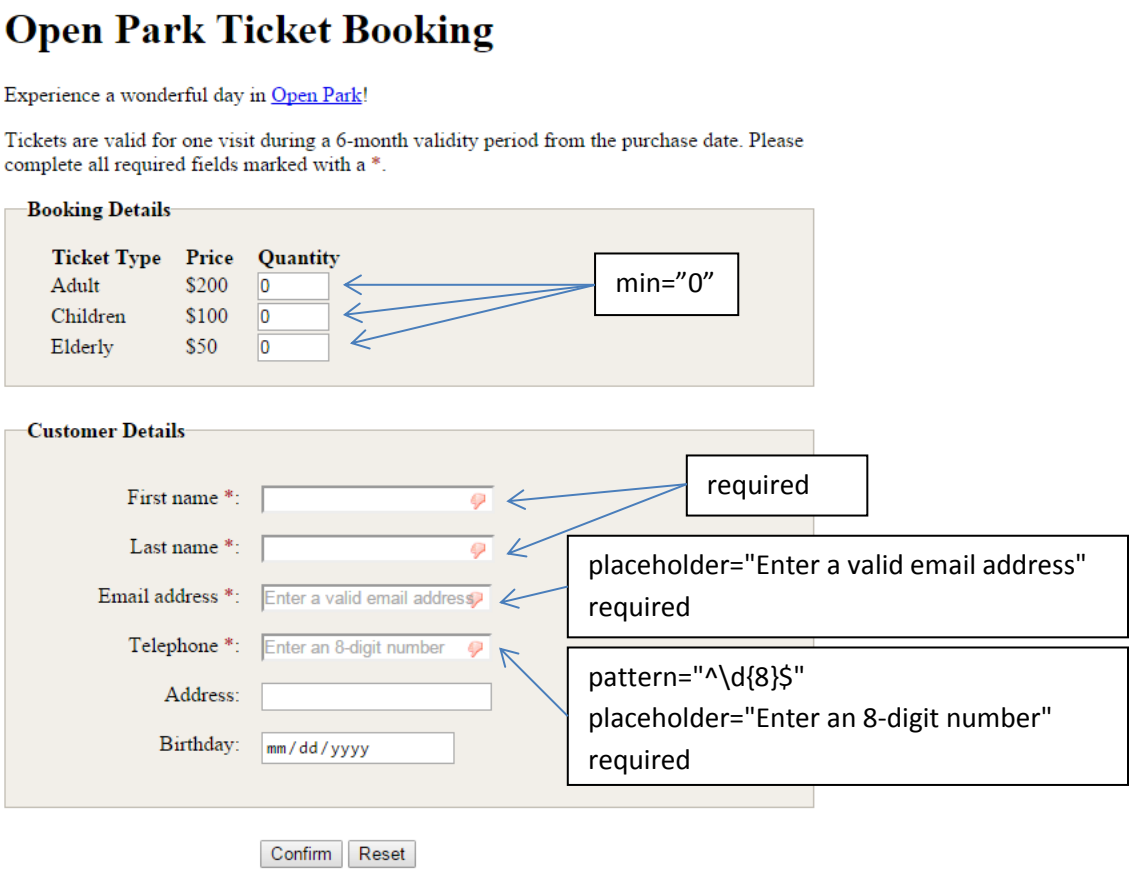
Your task: We want to make sure that the customer has filled in all the required fields and, more importantly, in the correct format. Follow the following steps:

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:



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.

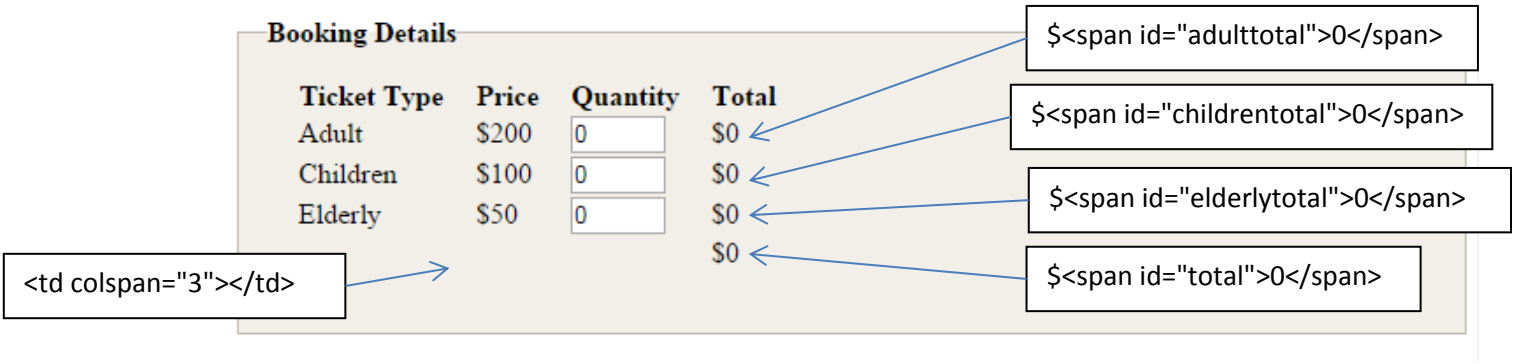
Javascript has the following characteristics:

- Loosely-typed: When we declare a variable, we only have to declare its name, not its type.
- Object-based: It has different built-in objects, and the some DOM object, 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 tickets before submitting the form.

Follow the following steps:

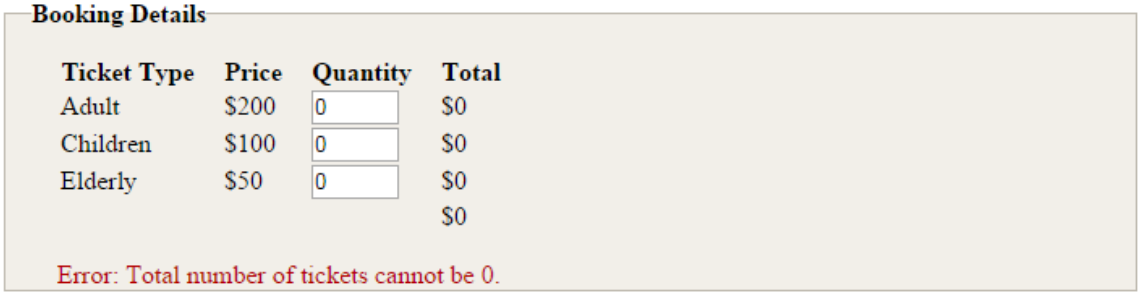
1. Add the Javascript file myscript.js to the working directory.
2. Add a new column of booking details in the HTML code, as follows:



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

4. Add the CSS rule such that the error message is shown if we click the submit button when no ticket is booked:

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



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 it before the body tag, as the HTML element does not exist yet when the code is loaded.

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) {
    errMsg.innerHTML = "";
    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;
}
```