Tutorial 4: HTML, CSS, JS

[Software used]

Any text editor you like (e.g. Notepad++, Visual Studio Code, Sublime, etc.), or some IDEs (e.g. Jetbrains WebStorm)

[Experimental Objective]

By building a web frontend of a flight management system from scratch, you'll learn some basic usage of HTML, CSS and Javascript.

HTML `table` Tag

Using HTML `table` tag, we can create a table on web page. A HTML table includes a `<table>` element, the `<table>` element includes one or more `<tr>`, `<th>` and `<td>` element. `<tr>` defines the row of a table, `<th>` defines the header of table, and `<td>` defines the cell of table (i.e. A single column in a row).

Let's create a new file named `table.html`, and input the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Flight list</title>
</head>
<body>
<table>
    <caption><h2>Flight List</h2></caption>
    <tr>
        <th>Flight No.</th>
        <th>Airline Company</th>
        <th>From</th>
        <th>To</th>
        <th>Date</th>
        <th>Departure Time</th>
        <th>Arrival Time</th>
    </tr>
    <tr>
        <td>ZH1858</td>
        <td>Shenzhen Airlines</td>
        <td>SHA</td>
        <td>PEK</td>
        <td>2019/09/25</td>
        <td>07:55</td>
        <td>10:10</td>
    </tr>
```


zhuym@sustech.edu.cn

```
    <tr>
        <td>MU5479</td>
        <td>Eastern Airlines</td>
        <td>TAO</td>
        <td>WUH</td>
        <td>2019/09/25</td>
        <td>13:40</td>
        <td>15:55</td>
    </tr>
    <tr>
        <td>CA5795</td>
        <td>Air China</td>
        <td>KMG</td>
        <td>CKG</td>
        <td>2019/09/20</td>
        <td>06:50</td>
        <td>08:25</td>
    </tr>
    <tr>
        <td>HU7707</td>
        <td>Hainan Airlines</td>
        <td>PEK</td>
        <td>SZX</td>
        <td>2019/09/20</td>
        <td>22:00</td>
        <td>01:30+1</td>
    </tr>
    <tr>
        <td>SC4837</td>
        <td>Shandong Airlines</td>
        <td>CKG</td>
        <td>TAO</td>
        <td>2019/09/20</td>
        <td>07:00</td>
        <td>09:25</td>
    </tr>
</table>
</body>
</html>
```

Then open this file in your browser, and you'll see the following:

Flight List

Flight No.	Airline Company	From	To	Date	Departure Time	Arrival Time
ZH1858	Shenzhen Airlines	SHA	PEK	2018/09/25	07:55	10:10
MU5479	Eastern Airlines	TAO	WUH	2018/09/25	13:40	15:55
CA5795	Air China	KMG	CKG	2018/09/20	06:50	08:25
HU7707	Hainan Airlines	PEK	SZX	2018/09/20	22:00	01:30+1
SC4837	Shandong Airlines	CKG	TAO	2018/09/20	07:00	09:25

That's a fairly simple and crude page, but we can add some styles on it and make it have a nicer look. Create a new file named `table2.html`, and input the following:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Flight list</title>
</head>
<body>
<table border="1" cellpadding="10" cellspacing="5">
    <caption><h2>Flight List</h2></caption>
    <tr>
        <th>Flight No.</th>
        <th>Airline Company</th>
        <th>From</th>
        <th>To</th>
        <th>Date</th>
        <th>Departure Time</th>
        <th>Arrival Time</th>
    </tr>
    <tr bgcolor="#F9E3AF">
        <td>ZH1858</td>
        <td>Shenzhen Airlines</td>
        <td>SHA</td>
        <td>PEK</td>
        <td>2019/09/25</td>
        <td>07:55</td>
        <td>10:10</td>
    </tr>
    <tr bgcolor="#F9E3AF">
        <td>MU5479</td>
        <td>Eastern Airlines</td>
        <td>TAO</td>
        <td>WUH</td>
        <td>2019/09/25</td>
```

```
        <td>13:40</td>
        <td>15:55</td>
    </tr>
    <tr bgcolor="#F9E3AF">
        <td>CA5795</td>
        <td>Air China</td>
        <td>KMG</td>
        <td>CKG</td>
        <td>2019/09/20</td>
        <td>06:50</td>
        <td>08:25</td>
    </tr>
    <tr bgcolor="#F9E3AF">
        <td>HU7707</td>
        <td>Hainan Airlines</td>
        <td>PEK</td>
        <td>SZX</td>
        <td>2019/09/20</td>
        <td>22:00</td>
        <td>01:30+1</td>
    </tr>
    <tr bgcolor="#F9E3AF">
        <td>SC4837</td>
        <td>Shandong Airlines</td>
        <td>CKG</td>
        <td>TAO</td>
        <td>2019/09/20</td>
        <td>07:00</td>
        <td>09:25</td>
    </tr>
</table>
</body>
</html>
```

And here are the explanations about the attributes we've added:
`border`: The `border` attribute specifies if a border should be displayed around the table cells or not.
`cellpadding`: The `cellpadding` attribute specifies the space, in pixels, between the cell wall and the cell content.
`cellspacing`: The `cellspacing` attribute specifies the space, in pixels, between cells.
`bgcolor`: The `bgcolor` attribute specifies a background color of a table.
(Note: The attributes above are not supported in HTML5 standard. This is for demonstration only, do not use them on production.)

zhuym@sustech.edu.cn

The page will then look like this:

Flight List

Flight No.	Airline Company	From	To	Date	Departure Time	Arrival Time
ZH1858	Shenzhen Airlines	SHA	PEK	2018/09/25	07:55	10:10
MU5479	Eastern Airlines	TAO	WUH	2018/09/25	13:40	15:55
CA5795	Air China	KMG	CKG	2018/09/20	06:50	08:25
HU7707	Hainan Airlines	PEK	SZX	2018/09/20	22:00	01:30+1
SC4837	Shandong Airlines	CKG	TAO	2018/09/20	07:00	09:25

CSS (Cascading Style Sheets)

We can achieve the same effect by using CSS (**C**ascading **S**tyle **S**heet).
Create a new file named `table.css`, and copy the following code:

```
@CHARSET "UTF-8";

#a {
    font-size: 20px;
    font-style: italic;
}

table {
    border: 2px solid black;
    border-collapse: separate;
    border-spacing: 5px;
    margin: 0 auto;
}

th {
    border: 1px solid black;
    padding: 15px;
    text-align: center;
    font-size: 15px;
}

td {
    border: 1px solid black;
    padding: 15px;
    background-color: #F9E3AF;
    text-align: center;
```

```
    font-size: 15px;
}
```

Don't forget to add the following line to your `table.html`, inside `<head>` tag:

```
<link rel="stylesheet" type="text/css" href="table.css"/>
```

In this stylesheet, there are several CSS Selectors:

Each of them took effect on `<table>`, `<th>`, `<td>` elements, respectively. `#a` is an **ID selector**, which set the style for a tag with given `id` attribute. The `#a` selector in CSS corresponds with the `id="a" attribute.`
`In` table.html, we edit the code of table caption as following:

```
<caption id="a"><h2>Flight List</h2></caption>
```

The page will then look like this:

Flight List

Flight No.	Airline Company	From	To	Date	Departure Time	Arrival Time
ZH1858	Shenzhen Airlines	SHA	PEK	2018/09/25	07:55	10:10
MU5479	Eastern Airlines	TAO	WUH	2018/09/25	13:40	15:55
CA5795	Air China	KMG	CKG	2018/09/20	06:50	08:25
HU7707	Hainan Airlines	PEK	SZX	2018/09/20	22:00	01:30+1
SC4837	Shandong Airlines	CKG	TAO	2018/09/20	07:00	09:25

HTML Forms

If we want to add some new flight information to this table, we need to create a user interface for users to input flight information manually, and create an item in the table. The HTML `<form>` element is used to create a HTML form.

Form can contain the `<input>` element. E.g., the text box, the check box, the radio button, and the submit button, etc.

Then, let's create a form for submitting flight information.

Add the following code in your `table.html`, between the tag `</table>` and `</body>`:

```
<div>
    <form name="form1">
        Flight No.: <input type="text" name="flight-no"/><br>

        Airline Company: <input type="text" name="airline-company"/><br>

        From: <input type="text" name="from"/><br>

        To: <input type="text" name="to"/><br>

        Date:
        <select id="year">
            <option>2020</option>
            <option>2019</option>
            <option>2018</option>
        </select> year
        <select id="month"></select> month
        <select id="day"></select> day <br>

        Departure Time:<select id="dhour"></select>:
        <select id="dminute"></select><br>

        Arrival Time:<select id="ahour"></select>:
        <select id="aminute"></select><br>

        <input type="button" value="Add Flight"><br>
    </form>
</div>
```

In the `input` elements in the form, different `type` values stands for different input boxes. `text` are the text boxes, `select` is the dropdown menu, `option` are the options under the dropdown menu. `button` is the submit button.

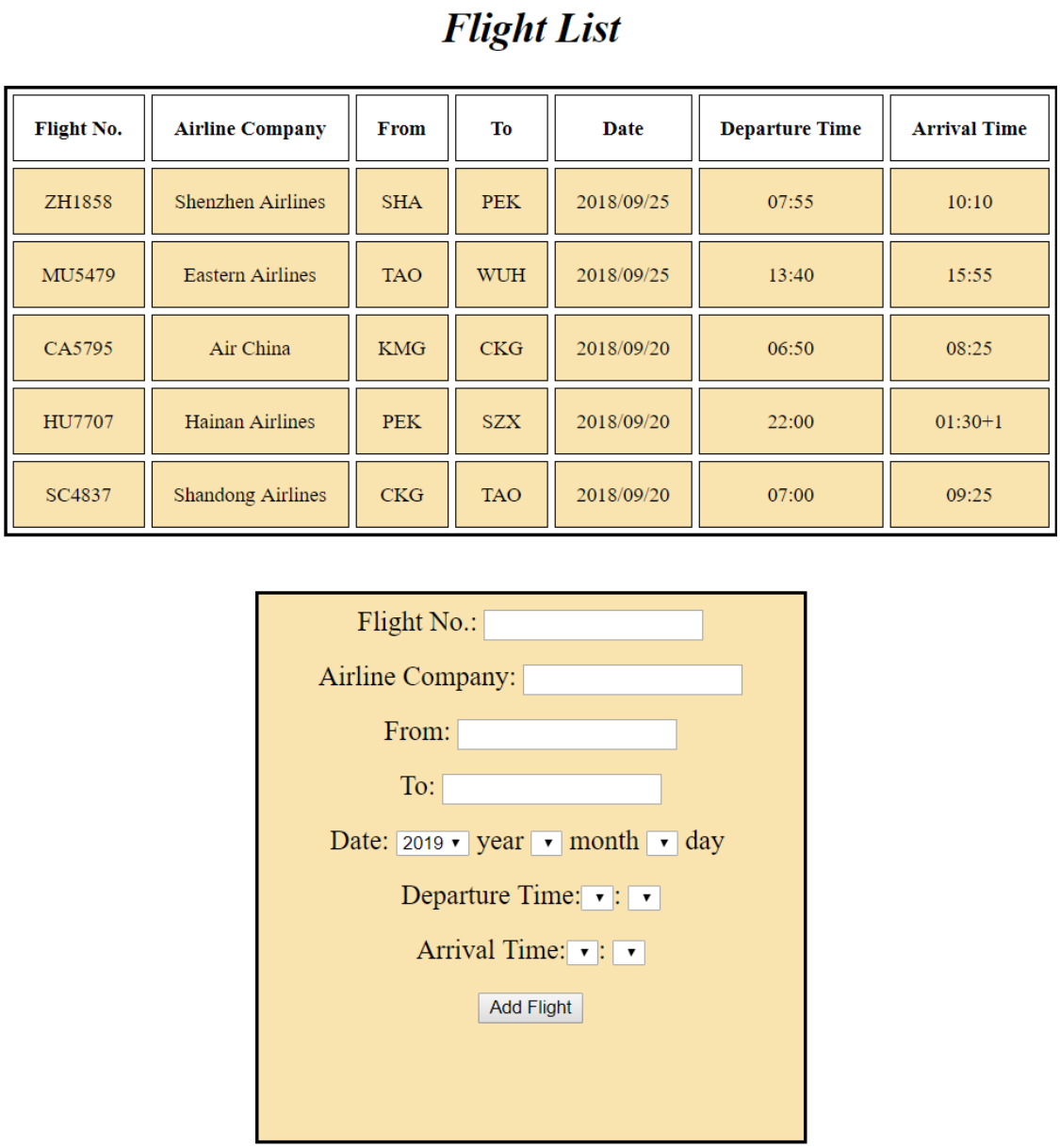
The aim of `<div>` element is to create divisions in the whole HTML document. They can separate the whole document to independent, different parts. It can be used for layout management, and do not need any special styles attached to it.

At the same time, we should add some styles to the `div` element. We'll also achieve this using CSS selectors in stylesheet:

```
div {
    line-height: 40px;
    width: 400px;
    height: 300px;
    text-align: center;
    background-color: #F9E3AF;
    font-size: 20px;
    margin: 40px auto 0;
```

```
    border: 2px solid black;
}
```

The page will then look like this:



Using JavaScript to validate user inputs

Adding JavaScript to your page

It is the first try that we connect the html file to the javaScript file.

First, create a file named `table.js` under the same directory of `table.html`:

zhuym@sustech.edu.cn

```
function onClickAddFlight() {
    let flightNo = document.querySelector('form input[name="flight-no"]').val
ue;
    alert(flightNo);
}
```

This function means to display the flight Number that being entered by user when clicking the "Add Flight" button, through which we can check whether our script is correctly linked to the HTML document.

The way we get the information entered by user is to use **querySelector()** method. The **QuerySelector()** method can return the first sub-element that is matched to the specific group of selectors. **Document** can be regarded as the root node of the html page, from which, we can acquire all sub-elements from the html page.

After that, in the `<head>` section of HTML document, adding the following line:

```
<script type="text/javascript" src="table.js"></script>
```

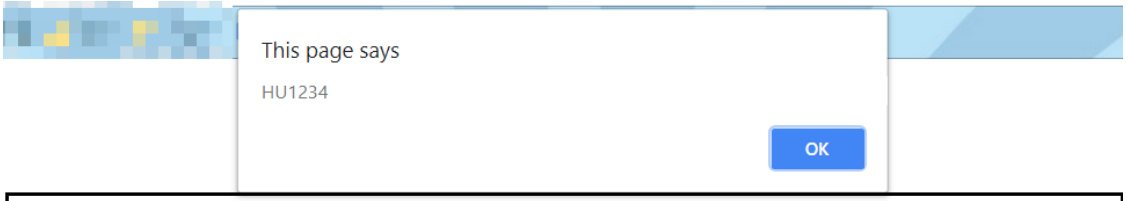
Then, modify the submit button, add a `onClick` attribute:

```
<input type="button" value="Add Flight" onclick="onClickAddFlight()"><br>
```

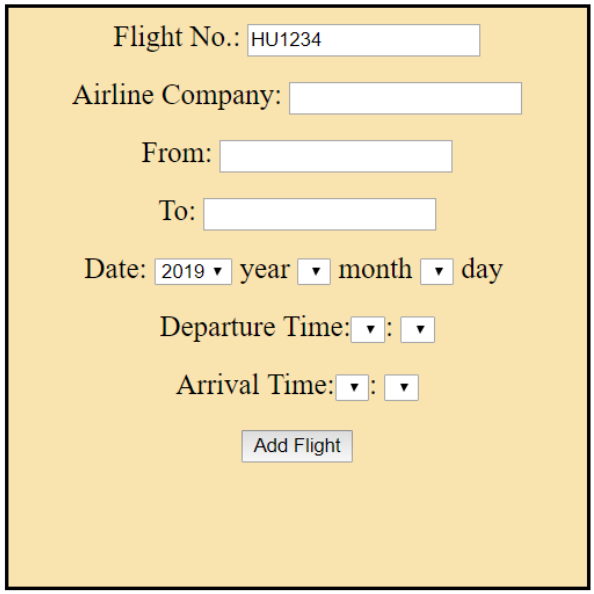
Note: This approach to add event listener for click is to test whether the html file is connected to .js file only. Do NOT hardcode event handlers in HTML attributes in real cases.

zhuym@sustech.edu.cn

Then we can open `table.html` in browser, input a flight number, and click "Add Flight" button to see the effect:



Flight No.	Airline Company	From	To	Date	Departure Time	Arrival Time
ZH1858	Shenzhen Airlines	SHA	PEK	2018/09/25	07:55	10:10
MU5479	Eastern Airlines	TAO	WUH	2018/09/25	13:40	15:55
CA5795	Air China	KMG	CKG	2018/09/20	06:50	08:25
HU7707	Hainan Airlines	PEK	SZX	2018/09/20	22:00	01:30+1
SC4837	Shandong Airlines	CKG	TAO	2018/09/20	07:00	09:25



Using regular expression to validate user input

Regular Expressions are patterns that used to match character combinations in strings. You can use regular expressions to match the patterns you want, e.g., e-mail addresses, IP addresses, phone numbers, etc.

Note: Regular expression is not a required part in this lecture. But it is better for you to understand some basic concepts and how to use them.

Here are some examples of using regular expressions in JavaScript:

```
/[A-Z0-9]+/ // matches one or more single upper case letter from A to Z, or a
 digit from 0-9
/[A-Z]{3}/ // matches exactly three uper case letters, each letter in the ran
ge of A to Z
/^Flight[A-Z]+\d+$/ // matches a line which starts with "Flight", then follow
ing one or more letters from A to Z, then one or more digits from 0-9, then t
he end of line.
```

Here are some explanations to the regular expressions above:

\d: Stands for digits (0-9)

[]: Stands for a character in given range

{N}: Repeat the pattern for N times

+: Appears no less than once

^: Start of a line

$: End of a line

If you want to learn more about regular expression, the following websites may be of help:

RegExr

Regexper

We have the following requirements for user input:

- Flight No. only contains upper case letters and digits. The first part is the code of airline companies, which is 2 characters long and can only contain digits and upper case letters. The second part is the sequence number, which is 3-4 digits long and can only contain numbers.

```
/^[A-Z0-9]{2}\d{3,4}$/
```

- The origin and destination airport code ("From" and "To" rows in the table) can only be 3 characters long, and can only contain upper case letters.

```
/^[A-Z]{3}$/
```

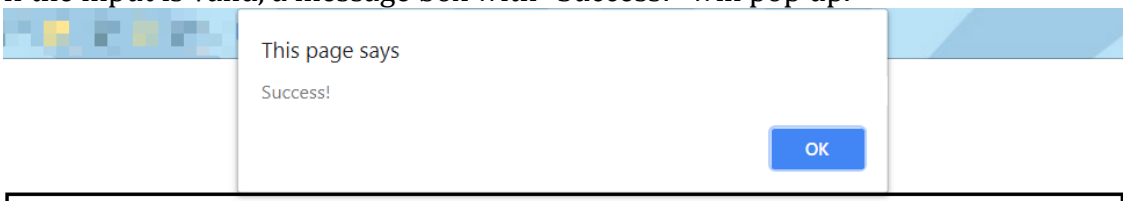
We modify the `table.js` as follows:

```
function onClickAddFlight() {
    let flightNo = document.querySelector('form input[name="flight-no"]').val
ue;
    let origin = document.querySelector('form input[name="from"]').value;
    let destination = document.querySelector('form input[name="to"]').value;
    if (validateInput(flightNo, origin, destination)) {
        alert("Success!");
```

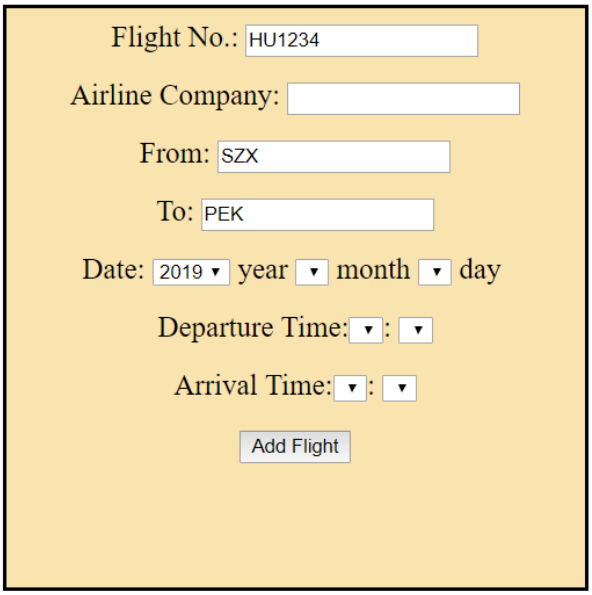
```
    }
}

function validateInput(flightNo, origin, destination) {
    let flightNoRegex = new RegExp(/^[A-Z0-9]{2}\d{3,4}$/);
    let airportCodeRegex = new RegExp(/^[A-Z]{3}$/);
    if (!flightNoRegex.test(flightNo)) {
        alert("Invalid Flight No.");
        return false;
    }
    if (!airportCodeRegex.test(origin)) {
        alert("Invalid origin airport code.");
        return false;
    }
    if (!airportCodeRegex.test(destination)) {
        alert("Invalid destination airport code.");
        return false;
    }
    return true;
}
```

If the input is valid, a message box with "Success!" will pop up.



Flight No.	Airline Company	From	To	Date	Departure Time	Arrival Time
ZH1858	Shenzhen Airlines	SHA	PEK	2018/09/25	07:55	10:10
MU5479	Eastern Airlines	TAO	WUH	2018/09/25	13:40	15:55
CA5795	Air China	KMG	CKG	2018/09/20	06:50	08:25
HU7707	Hainan Airlines	PEK	SZX	2018/09/20	22:00	01:30+1
SC4837	Shandong Airlines	CKG	TAO	2018/09/20	07:00	09:25



More on `<select>` dropdown menu

Next, we'll add some items into the date and the time menu.

First, setting the year for a reasonable range (from 2000 to 2020).

Adding an `onload` attribute to the `<body>` tag, while delete the old `<option>` items:

zhuym@sustech.edu.cn

```
<body onload="initial()">

...
        Date:
        <select id="year"></select> year
        <select id="month"></select> month
        <select id="day"></select> day <br>
...
```

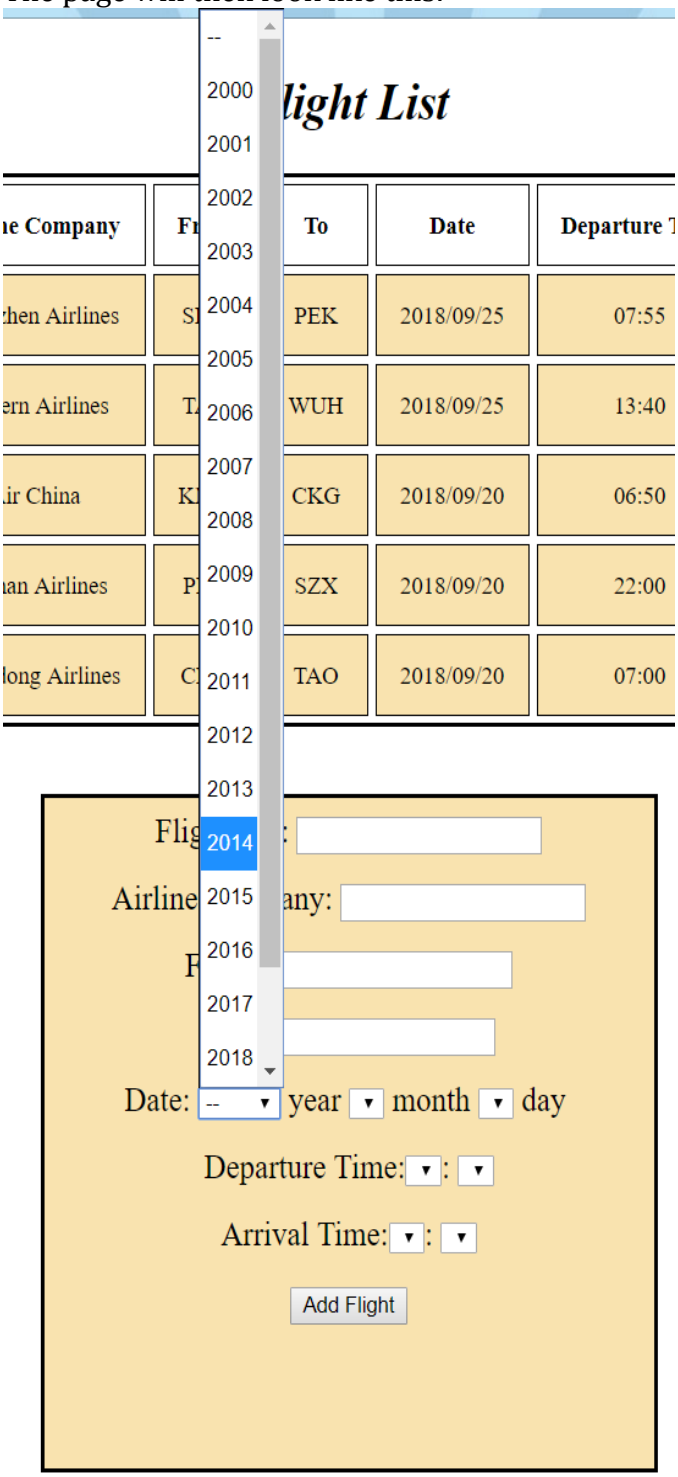
onload attribute stands for the code that will be executed when loading the element.

Note: This approach to add event listener is for demonstration only. Do NOT hardcode event handlers in HTML attributes in real cases.

And define `initial()` function in `table.js`:

```
function initial() {
    let year = document.getElementById("year");
    year.innerHTML = "";
    year.options.add(new Option("--", null));
    for (let i = 2000; i <= 2020; i++) {
        year.options.add(new Option(i, i));
    }
}
```

The page will then look like this:



zhuym@sustech.edu.cn

Next, we use the same approach to modify month, day, and two times.

First, add **onchange** attribute to the dropdown menu of year and month. The event will be triggered when the selected item changes.

```
<select id="year" onchange="setMonth()"></select> year
<select id="month" onchange="setDay()"></select> month
```

Note: This approach to add event listener is for demonstration only. Do NOT hardcode event handlers in HTML attributes in real cases.

Add the following code to `table.js`:

```
function setMonth() {
    let month = document.getElementById("month");
    month.innerHTML = "";

    month.options.add(new Option("--", null));
    for (let i = 1; i <= 12; i++) {
        month.options.add(new Option(i, i));
    }
}

function setDay() {
    let year = document.getElementById("year").value;
    let month = document.getElementById("month").value;
    let day = document.getElementById("day");
    let data = new Array(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
    // clear the items
    day.innerHTML = "";

    // add new items
    day.options.add(new Option("--", null));
    for (let i = 1; i <= data[month - 1]; i++) {
        day.options.add(new Option(i, i));
    }
    if (((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) && month == 2
) {
        day.options.add(new Option(29, 29));
    }
}
```

To ensure that date and time are correctly entered, `onClickAddFlight` function should be modified. When user fills all the selections for date and time, the form will pop-up a "success", otherwise it may inform user to input date and time. **Please complete this part by yourself.**

Adding <tr> elements dynamically

After receiving and validating user's input, we need to implement the function to add new flight into the table. Here we'll use a <tbody> tag, which represents the body part of a table. A <tbody> element must include one or more <tr> tags.

In this exercise, we use the following method: Add a table line <tr> element to <tbody>, as a template, and hide it by setting the style to display:none. Every time we add a new line, we display the newly added table line.

Delete the old data, and add a <tbody> tag, set its style attribute to dispaly: none:

```
<table>
    <caption id="a"><h2>Flight List</h2></caption>
    <tr>
        <th>Flight No.</th>
        <th>Airline Company</th>
        <th>From</th>
        <th>To</th>
        <th>Date</th>
        <th>Departure Time</th>
        <th>Arrival Time</th>
        <th></th>
    </tr>
    <tbody id="tbody" style="display: none">
    <tr>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td>
            <input type="button" value="Delete">
        </td>
    </tr>
    </tbody>
</table>
```

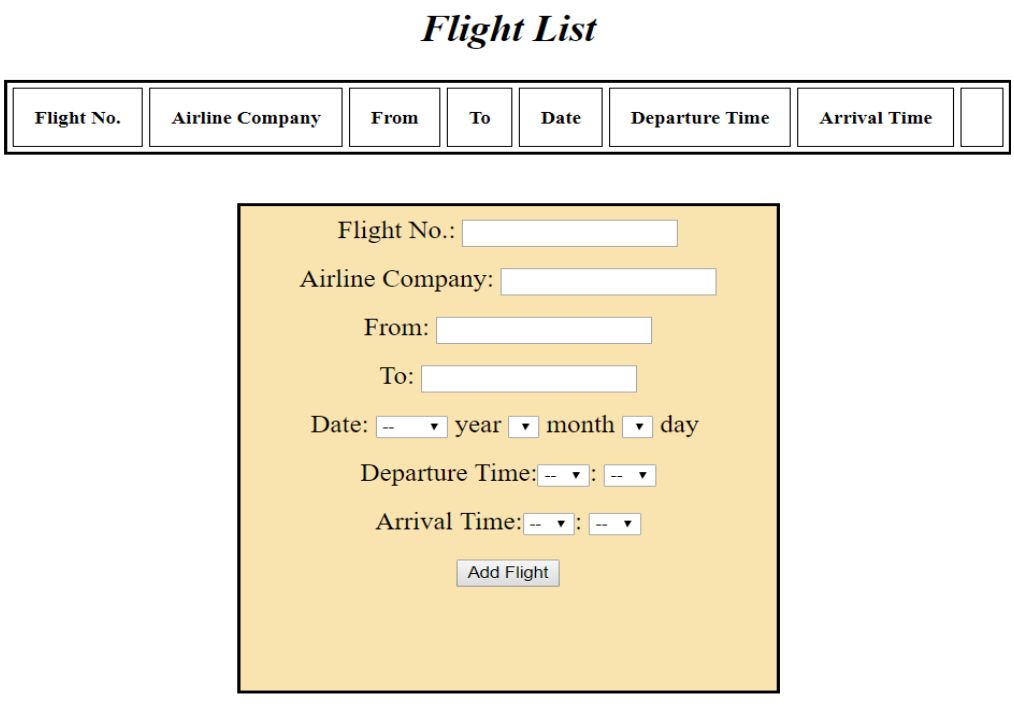
Modify table.js, add a function with dynamically add row to table.

```
function addRow() {
    let bodyObj = document.getElementById("tbody");
    if (bodyObj === null) {
        alert("Body of Table not Exist!");
        return;
    }
    let year = document.getElementById("year").value;
    let month = document.getElementById("month").value;
```

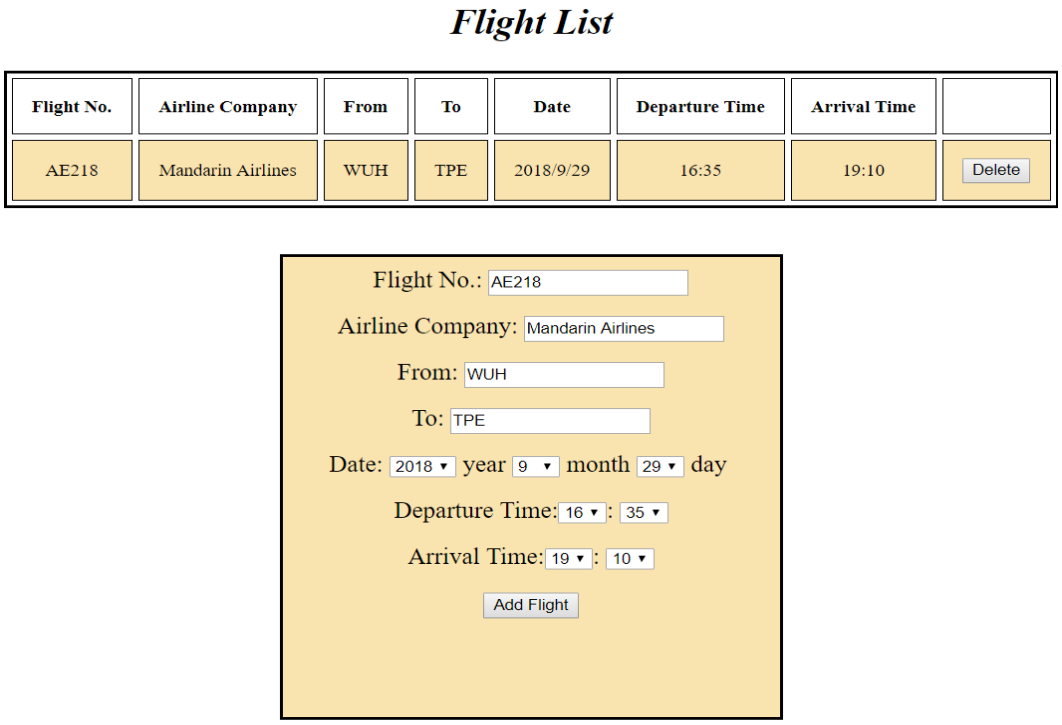
```
    let day = document.getElementById("day").value;
    let dhour = document.getElementById("dhour").value;
    let dminute = document.getElementById("dminute").value;
    let ahour = document.getElementById("dhour").value;
    let aminute = document.getElementById("dminute").value;
    let rowCount = bodyObj.rows.length;
    let cellCount = bodyObj.rows[0].cells.length;
    bodyObj.style.display = ""; // display the tbody
    let newRow = bodyObj.insertRow(rowCount++);
    newRow.insertCell(0).innerHTML = document.forms[0]["flight-no"].value;
    newRow.insertCell(1).innerHTML = document.forms[0]["airline-company"].val
ue;
    newRow.insertCell(2).innerHTML = document.forms[0].from.value;
    newRow.insertCell(3).innerHTML = document.forms[0].to.value;
    newRow.insertCell(4).innerHTML = year + "/" + month + "/" + day;
    newRow.insertCell(5).innerHTML = dhour + ":" + dminute;
    newRow.insertCell(6).innerHTML = ahour + ":" + aminute;
    newRow.insertCell(7).innerHTML = bodyObj.rows[0].cells[cellCount - 1].inn
erHTML; // copy the "delete" button
    bodyObj.rows[0].style.display = "none"; // hide first row
}
```

Then delete the `alert("Success!")` in function `OnClickAddFlight`, and add the call to `addRow()` function.

The page will look like the following:



Before adding new row



After adding new row

Deleting `<tr>` elements dynamically

Add `removeRow()` function to `table.js`:

```
function removeRow(inputobj) {
    if (inputobj == null) return;
    let parentTD = inputobj.parentNode;
    let parentTR = parentTD.parentNode;
    let parentTBODY = parentTR.parentNode;
    parentTBODY.removeChild(parentTR);
}
```

We use the button itself to find the parent `<tr>` element (which is the row to be removed). Then delete it in `<tbody>`.

Add the function to event listener of "delete" button via `onclick` attribute:

```
<input type="button" value="Delete" onclick="removeRow(this)">
```

Note: This approach to add event listener is for demonstration only. Do NOT hardcode event handlers in HTML attributes in real cases.

Use hyperlinks to jump across the pages

Create a new HTML document named `myPage.html`, input the following:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>myPage</title>
</head>
<body>
<a href="table.html">
    <h1>click here to enter the flight list</h1>
</a>
</body>
</html>
```

The page will look like this:

That's a simple and dumb page. If you click the hyperlink, you'll be redirected to the flight list page (table.html)

Designed by Yueming ZHU (in 2017) and Zhengchang HUA (modified in 2018).