

# WEEK08 – HTML FORM WITH SIMPLE JAVASCRIPT

---

WEN-BIN JIAN

DEPARTMENT OF ELECTROPHYSICS, NATIONAL CHIAO TUNG UNIVERSITY

# OUTLINE

---

- 1. HTML Form**
- 2. Basic JavaScript Concept**
- 3. Basic Object-Oriented Programming**
- 4. JavaScript in HTML**
5. Input Elements in Form: text, password, button, radio box, check box
6. An Array Variable in HTML
7. New HTML5 Input Elements
8. Put Codes in `<script></script>`

# HTML FORMS

- `<form>` ... `</form>`: a form used to collect user input
- `<input>`: a user input element, many different types like text, radio, ...
- `<fieldset>`: an container used to group input elements
- `<legend>`: an element to define a caption for `<fieldset>`
- `<input type="submit">`: a submit button, pressed after the form is filled – for a formal usage, data in the form have to be submitted to a **server side language processing**.

one form – one submit button  
inside the same fieldset of the form, use ids to use elements

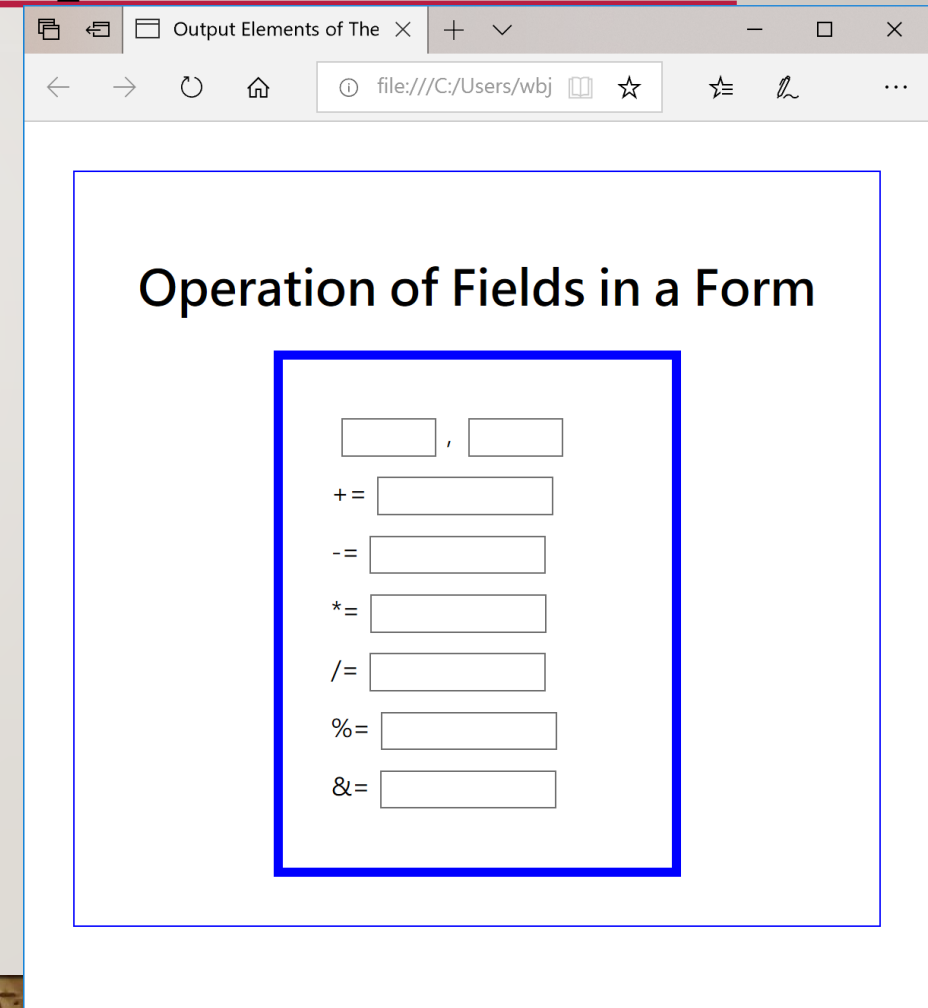
IC\_W801.html

The screenshot shows a web browser window with the title 'Form Example 1'. The address bar shows a file path: 'file:///C:/Users/wbjia/Documents/Lecture%20Notes/2018%20(9-12)?'. The main content area contains a form with a text area containing Chinese text, a section titled '與作者對話' (Communicate with the author) with a text input field, and a '發送郵件' (Send email) button. The footer of the browser window displays 'Copyright Owned By Mr. Wen-Bin Jinn'.

# HTML FORMS

- `<form action="file or mailto:abc@mail.com">`: submit data to a server or to a command
  - The action is usually referred to a server side scripting file.
- `<form method="get or post">`: the submitted data will be visible or not
- `<output>`: an element displaced either in or out of the form while it belongs to the form
  - The action always requires codes for processing. Here the JavaScript codes have to be implemented.

IC\_W802.html



The screenshot shows a web browser window with the title "IC\_W802.html". The address bar shows the file path "file:///C:/Users/wbj". The browser content displays a form titled "Operation of Fields in a Form". The form contains several input fields and operators:

- Two input fields separated by a comma:  ,
- Arithmetic operators with input fields:  $+=$  ,  $-=$  ,  $*=$  ,  $/=$  ,  $\%=$  , and  $\&=$

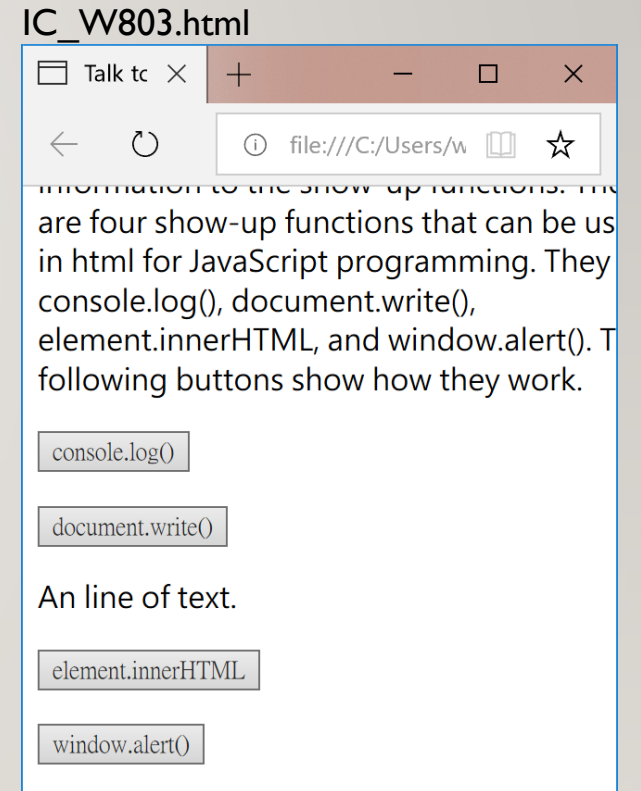
# BASIC CONCEPTS OF JAVASCRIPT & OBJECT-ORIENTED PROGRAMMING

---

- The `oninput` of the table element is used to call a JavaScript function.
- You can write a JavaScript function in the document like `func()` and arrange the event handling as `oninput="func();"`.
- Another commonly used event function is the `onclick` event of the button element.
- You can alternatively put one line code in it like `window.alert("my information")` to handle the event.
- You can also put several lines of codes in it like `{line 1; line 2; line 3;...}`

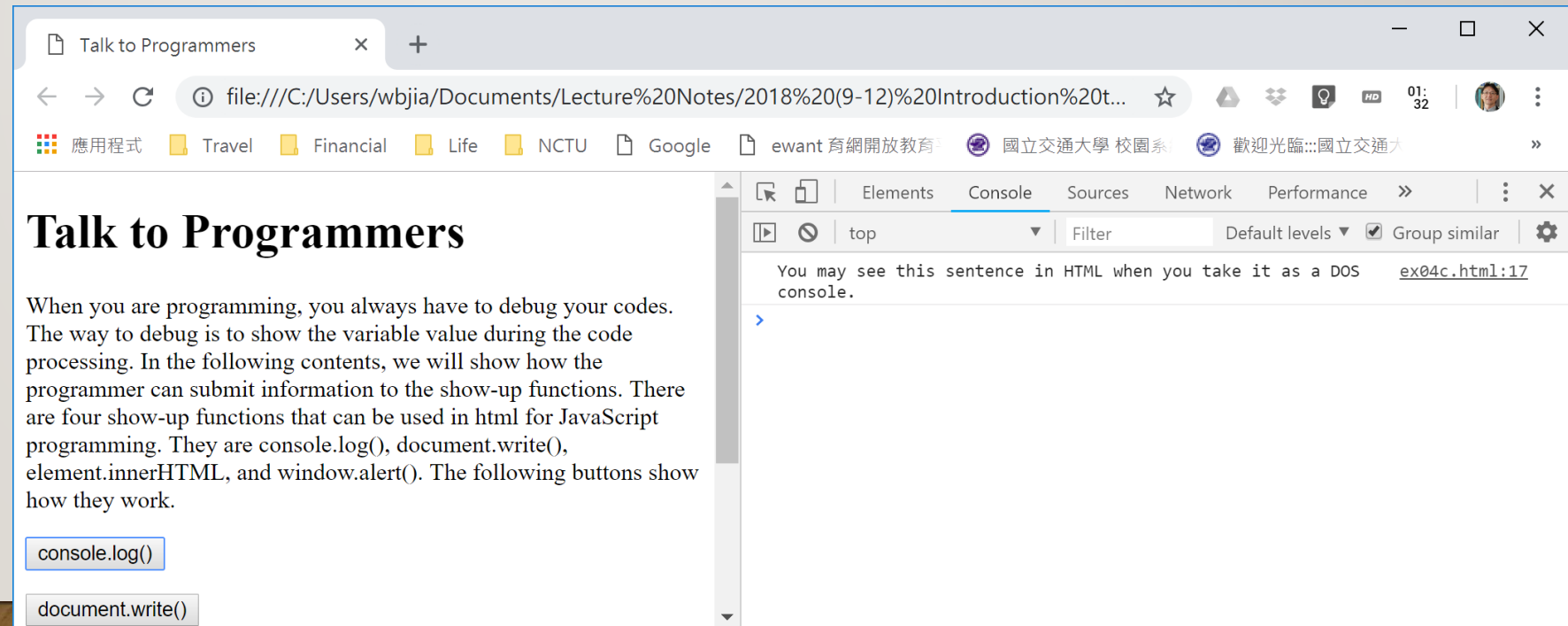
# BASIC CONCEPTS – debug

- Simple talking functions are: `element.innerHTML`, `document.write()`, `window.alert()`, `console.log()` (開發人員工具 → 顯示主控台).
- Data objects are `element`, `document`, `window`, `console`.
- Here `innerHTML` is one data of the element. Recall that value is also data of some other elements.
- The `write()` function is one function of the document object and the `alert()` function is one function of the window object. The window object is different from the document.



# BASIC CONCEPTS – debug

- Debug your HTML code in Chrome.
- Ctrl+Shift+I (滑鼠右鍵 → 檢查 → Console)
- Internet Explore: 滑鼠右鍵 → 檢查元素 → 主控台



# BASIC CONCEPTS – var, type, conversion

---

- **The id or name of an element is the variable name used in JavaScript (used in the scope of the same fieldset in the same form). You may use `document.form_name.elem_name` to get your element objects.**
- In JavaScript, variables are declared using “`var name_of_variable;`”.
- Note that there are no different types for variables.
- In C or C++, you have variable types of (unsigned) short, long, int, char, float, double, byte, bool.
- In JavaScript, the `var` command declares a new variable without indication of its type.
- The type can be converted using functions like `parseInt()`.





# BASIC CONCEPTS – `document.getElementById()`

---

- The scope of the variables are in the document.
- When you trigger any 'refresh' of your html document, all the contents of your variables disappear.
- The variable is one object data so you can change the data attributes by using `variable_name.attribute` like `name.value`.
- For elements with ids, you access the object using `document.getElementById("elem_id")`. Then you use its data (attributes) & functions.



# BASIC CONCEPTS – operators, conversion functions

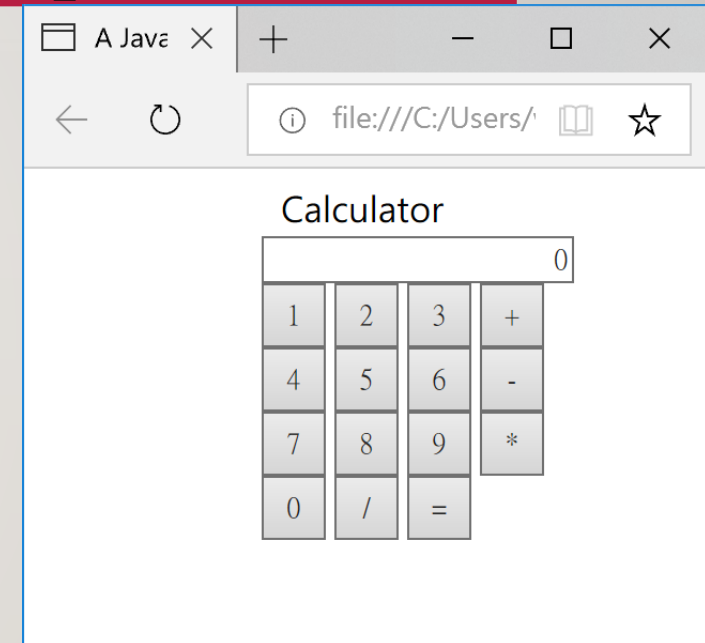
---

- The JavaScript provides basic operation for numbers: +, -, \*, /, %, >>, <<, &, | ...
- The JavaScript provides basic operation for strings: +
- The JavaScript provides functions to transform a string to an integer or a float number: parseInt(), parseFloat().
- Note that the parseInt() and parseFloat() are standalone functions. Sometimes you need to include libraries for calling some functions.
- In JavaScript, you have object data, child objects, object functions, child object functions, standalone functions.

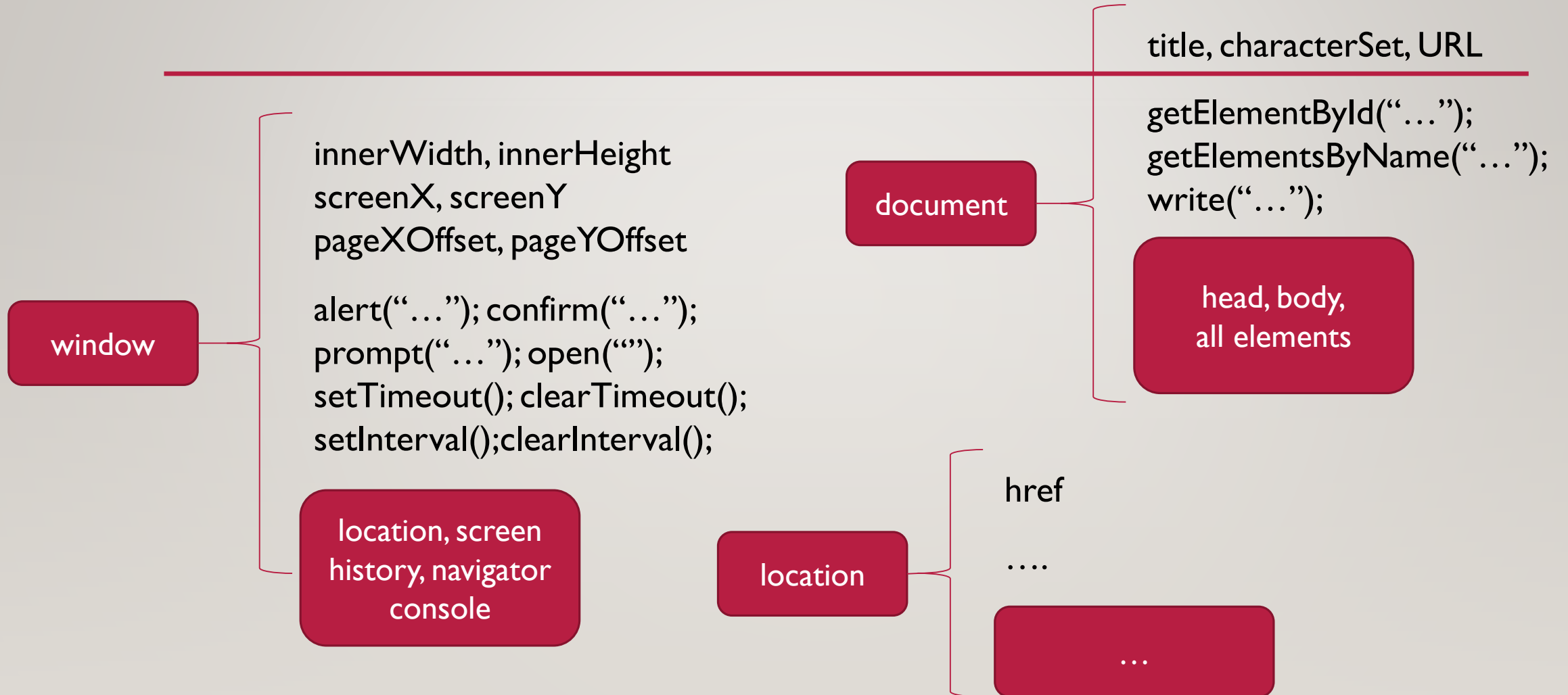
# BASIC CONCEPTS – function

- Any events can be processed by a series of commands embraced by curly braces.
- You can collect that series of commands into a function.
- In JavaScript, we define a function as “**function func\_name(input parameters) {lines of commands;}**”.
- Put the Javascript codes in any places embraced by `<script>` and `</script>` in your html document.

IC\_W806.html



# BASIC CONCEPTS – objects



# OUTLINE

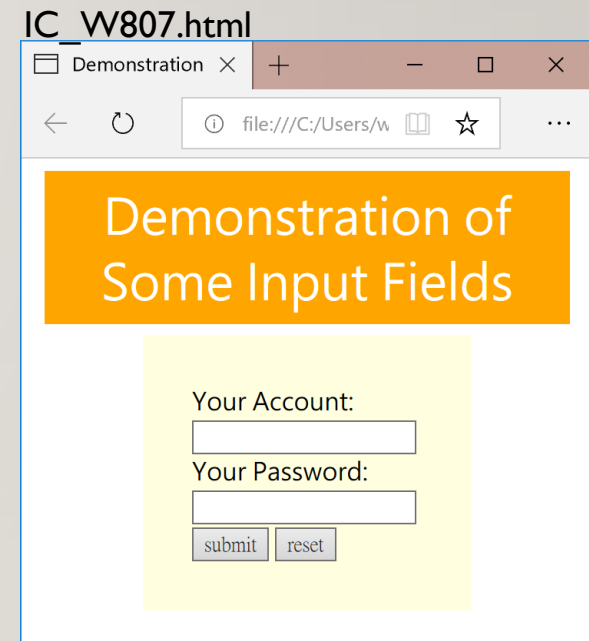
---

1. HTML Form
2. Basic JavaScript Concept
3. Basic Object-Oriented Programming
4. JavaScript in HTML
- 5. Input Elements in Form: text, password, button, radio box, check box**
- 6. An Array Variable in HTML**
- 7. New HTML5 Input Elements**
- 8. Put Codes in `<script></script>`**

# HTML INPUT & INPUT TYPES

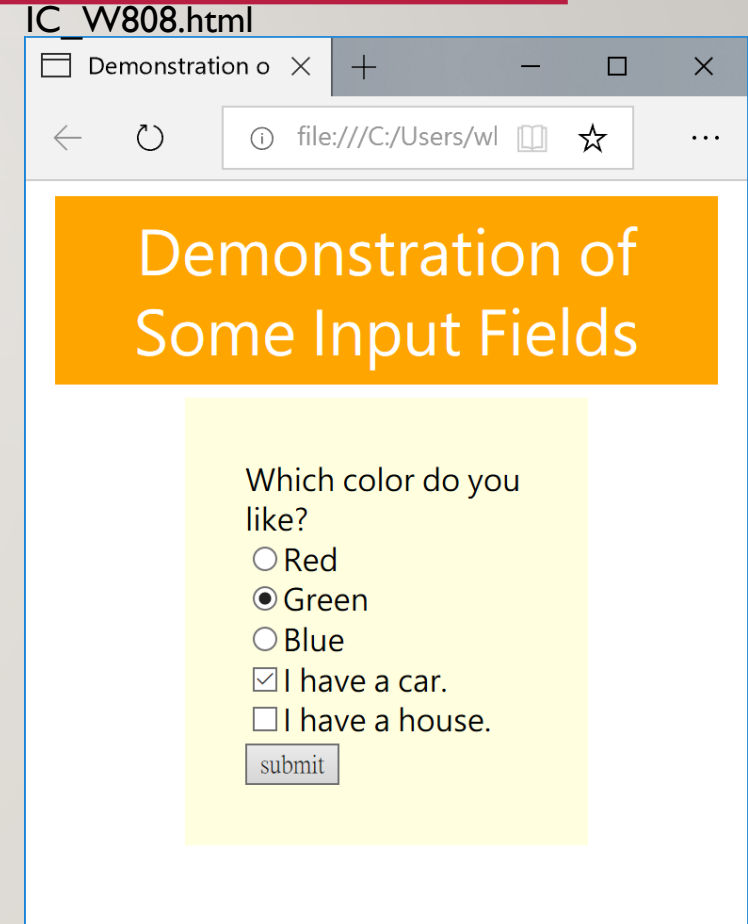
---

- Form elements can be displaced outside the form as elements of the document.
- The `<input>` element is a **void (empty) element** and it can be displayed in several ways according to the attribute of type.
- The `<input type="text">` defines a one-line text input field.
- The `<input type="password">` defines a one-line password field.
- The `<input type="submit">` defines a button for submitting form data.



# HTML INPUT & INPUT TYPES

- The `<input type="reset">` defines a reset button that will reset the form values.
- The `<input type="radio" name="g_name">` defines a radio button. Radio buttons are grouped by setting the same name. The attribute `checked` defines that this radio button is selected.
- The `<input type="checkbox" name="g_name">` defines a checkbox.
- The `<input type="button">` defines a button.



# THE ARRAY VARIABLE FOR THE INPUT TYPE OF RADIO BUTTON

---

- Some elements using the same variable for their attribute of 'name'. That will result in the declaration of an array variable.
- The array variable in JavaScript is declared by
  - `var emta = [];` // empty array
  - `var anum = [1, 5, 20];` // an array of number, now `anum[0]` is 1, `anum[1]` is 5, and `anum[2]` is 20; `anum.length` is 3
- The number of elements in an array is obtained by using `array_var_name.length`.

IC\_W809.html

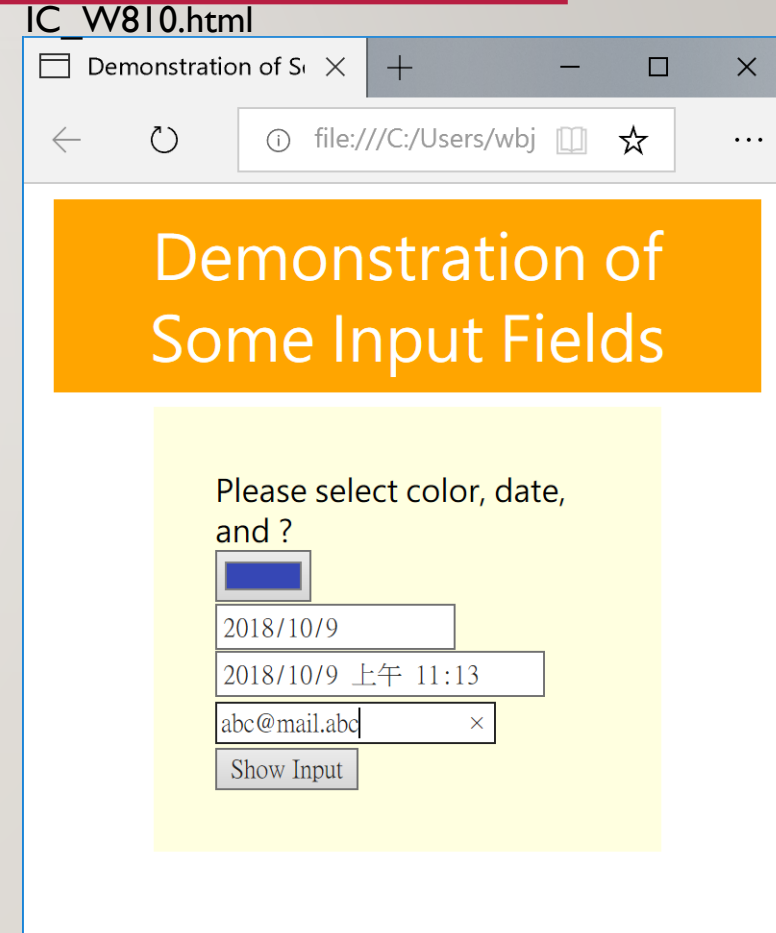
```
<input type="checkbox" name="chkbx">I have a car.<br>
<input type="checkbox" name="chkbx">I have a house.<br>
<input type="checkbox" name="chkbx">I got married.<br>

<input type="button" value="check all" onclick="{
    for(var i=0; i<chkbx.length; i++) chkbx[i].checked = true;
}">
<input type="button" value="submit" onclick="{
    var outtxt = "";
    if (chkbx[0].checked) outtxt+='You got a car.\n';
    if (chkbx[1].checked) outtxt+='You got a house.\n';
    if (chkbx[2].checked) outtxt+='You got married.';
    window.alert(outtxt);
}">
```



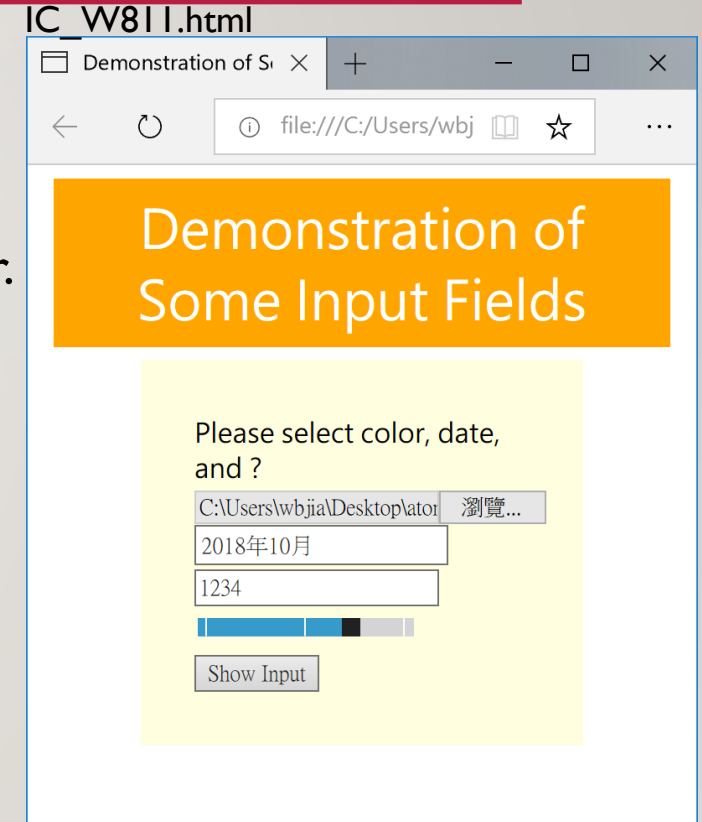
# HTML INPUT & INPUT TYPES – HTML5

- The `<input type="color">` defines a color.
- The `<input type="date">` defines a date. The attributes of `min` and `max` set the boundary.
- The `<input type="datetime-local">` defines a date and time input field.
- The `<input type="email">` defines an input field for an email address.



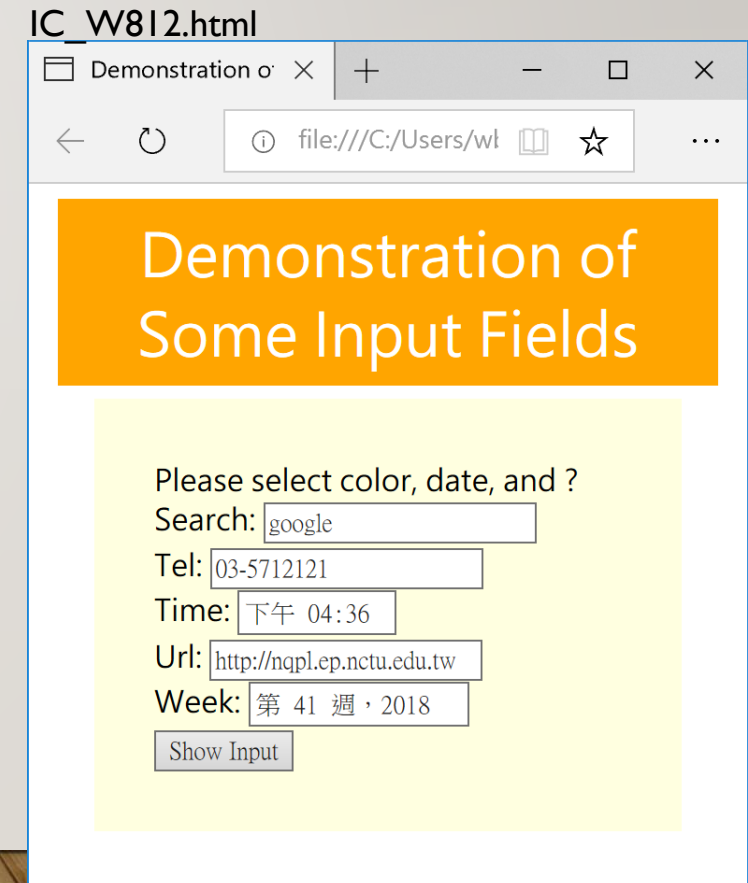
# HTML INPUT & INPUT TYPES – HTML5

- The `<input type="file">` defines an input of a file with a “browse” button for file uploads.
- The `<input type="month">` defines a selector of a month and year.
- The `<input type="number">` defines a numeric input field.
- The `<input type="range">` defines a control for entering a number in a range.



# HTML INPUT & INPUT TYPES – HTML5

- The `<input type="search">` defines a field of words used for searching.
- The `<input type="tel">` defines a field for a telephone number.
- The `<input type="time">` defines a selector for a time.
- The `<input type="url">` defines a input field for a URL.
- The `<input type="week">` defines a selector for a week.



# THE WORKING SCOPE OF JAVASCRIPT VARIABLES IN HTML

---

- You can put your codes in `<script></script>`
- Use the browser object model, the document, to get the element
- Use the function `'document.getElementsByName("name");'` to get the element
- Then you can do it the same way
- You can put `<script>` in the `<head>` or in the `<body>`

IC\_W8I3.html

```
<script>
    var chkbx = document.getElementsByName("chkbx");
    function ckaclk(){
        for(var i=0; i<chkbx.length; i++) chkbx[i].checked = true;
    }
    function smtclk(){
        var outtxt = "";
        if (chkbx[0].checked) outtxt+="You got a car.\n";
        if (chkbx[1].checked) outtxt+="You got a house.\n";
        if (chkbx[2].checked) outtxt+="You got married.";
        window.alert(outtxt);
    }
</script>
```

# EXERCISE

---

1. Please prepare a form for password input (a password input field and an OK button). In the form, please give buttons of digital numbers, 0, ..., 9, for the user to input the password. Please arrange the buttons of digital numbers not in order. When the user press the OK button, please use `window.alert()` to print out the password.
2. Please design a simple calculator with plus, minus, time, divide operations for binary numbers. The maximum number of digits is 12.
3. Please design a XO game by using buttons. When the first player clicks, the button shows O. When the second player clicks, the button shows X. You don't need to determine who wins the game.