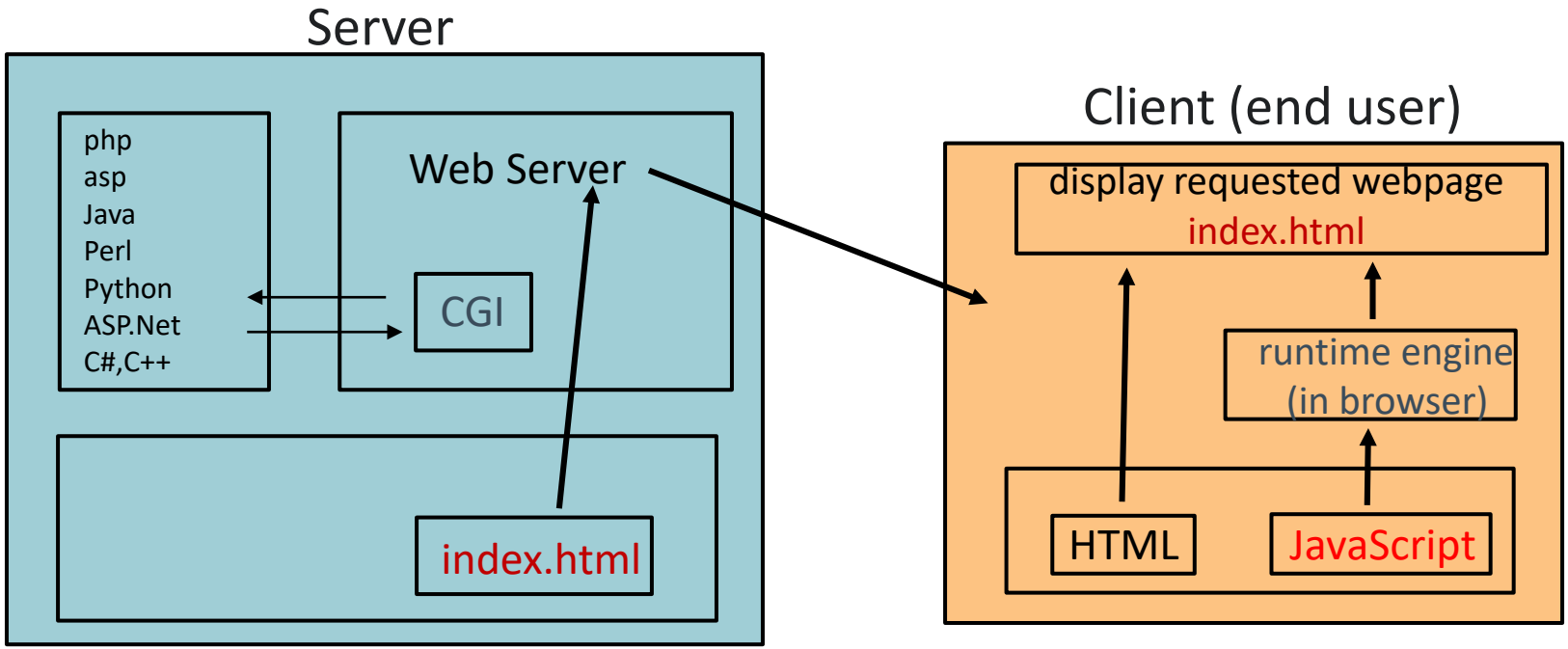
JavaScript: Introduction to Scripting

Shih-Yi (James) Chien
National Chengchi University
sychien@nccu.edu.tw

Introduction

- JavaScript
  - Scripting language
  - Enhance the functionality and appearance of web pages
- Major web browsers contain JavaScript **interpreters**
  - Process the commands written in JavaScript
- Object-oriented programming (OOP) language
  - document, images, button, window…

Server-side programs vs. Client-side JavaScript



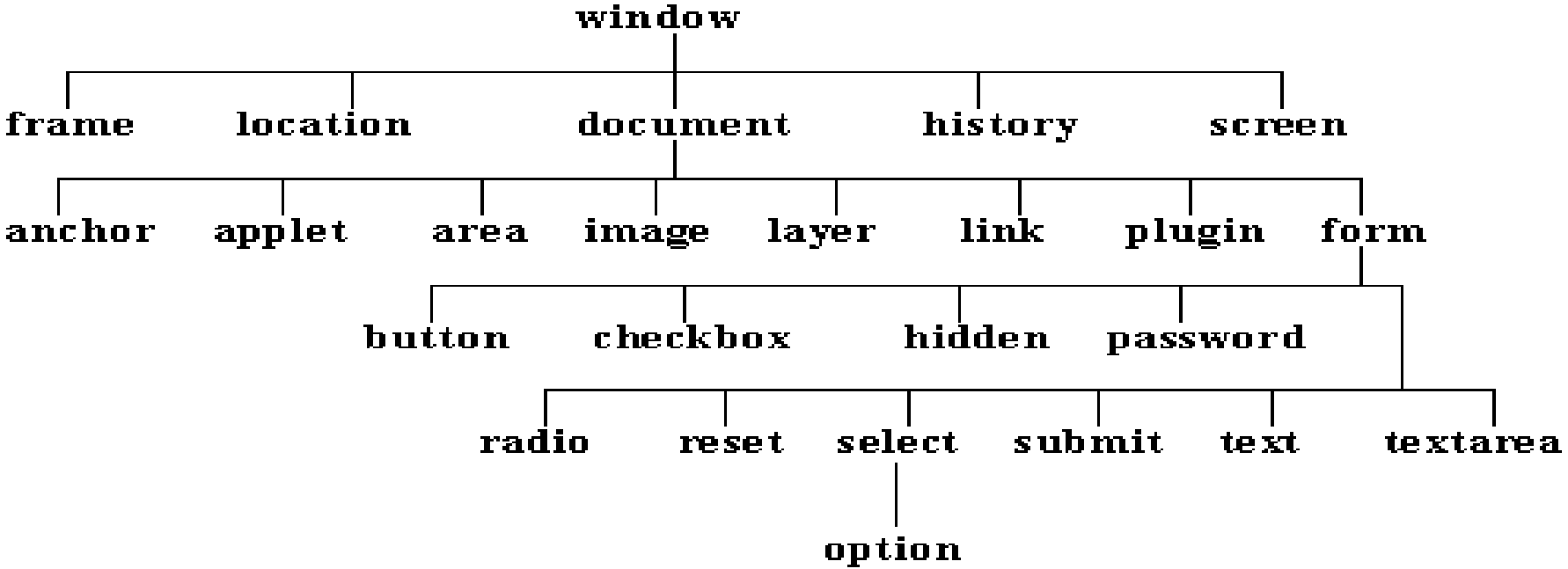
CGI (common gateway interface): protocol for web servers to execute programs

runtime engine: used as interpreter in the Web browsers to execute JavaScript code

property vs. method

- object.**property**
  - assign a value
  - document.**bgColor**=blue;
- object.**method**
  - execute an action
  - document.**write** (‘Hello World’);

Object Hierarchy

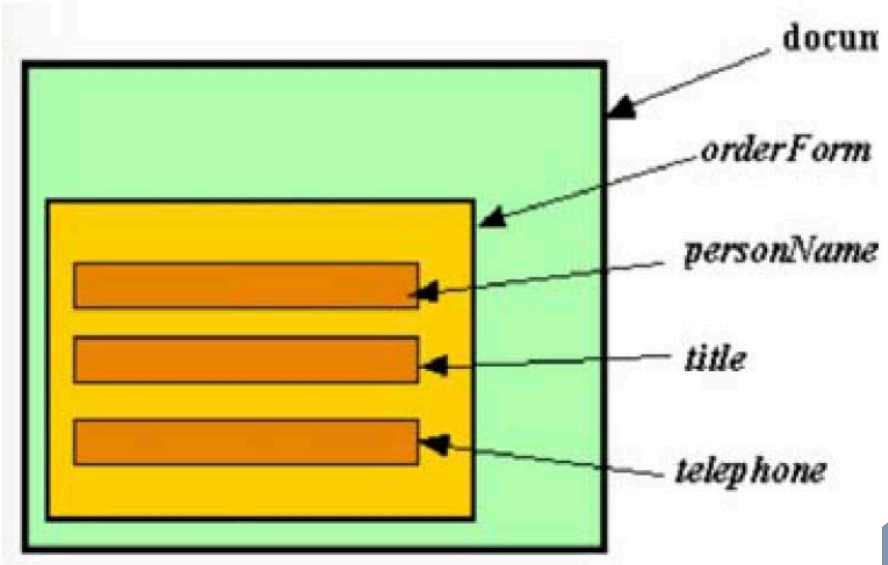


document.form.text

Identifier

document.**orderForm**.personName

- Object: document. **orderForm**
- Property: personName
- More examples:
  - **document.bgcolor**
  - **document.location**
  - **document.images.length**



Display Text in a Web Page

- <script> Tag: The text that follows is part of a script
  - <script type = “text/javascript”> … </script>
  - <script> ... </script>
- Content can be contained between double quotation (“) or single quotation (‘) marks
  - document.write(“Hello world.”);
  - document.write(‘Hello world.’);

```
<html>
<body>
  <P>This is a paragraph.</P>

  <script>
    document.write("Hello world!")
  </script>

  <P>This is a paragraph.</P>
</body>
</html>
```

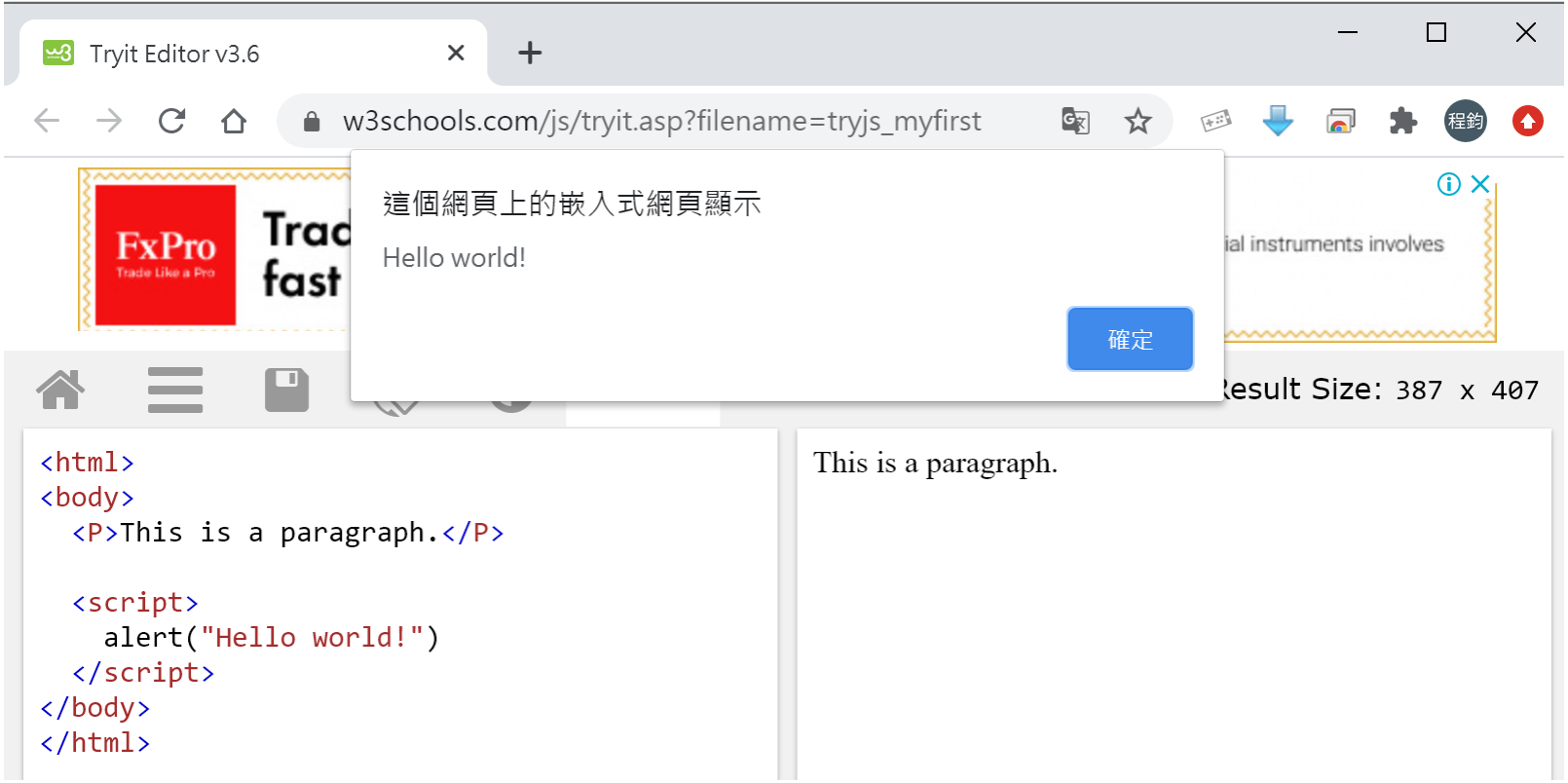
This is a paragraph.

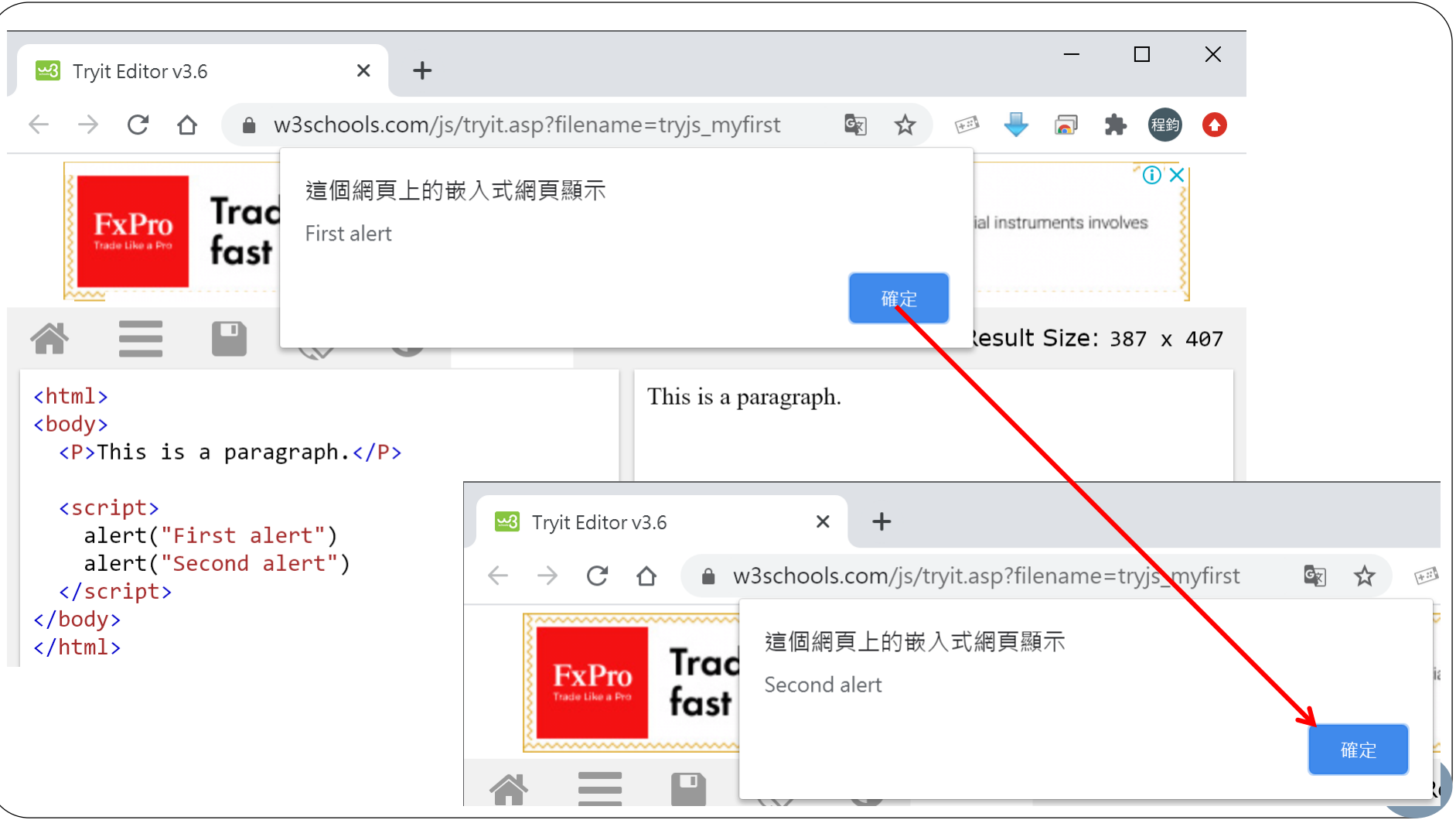
Hello world!

This is a paragraph.

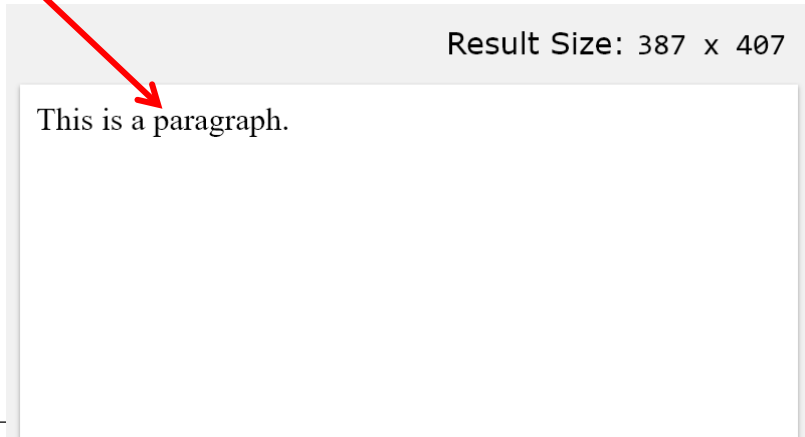
Displaying Text in an Alert Dialog

- Display important messages
- "Pop up" on the screen to grab the user's attention
- Browser's window object
  - Use method alert to display an alert dialog
- Method alert
  - Require as its argument the string to be displayed

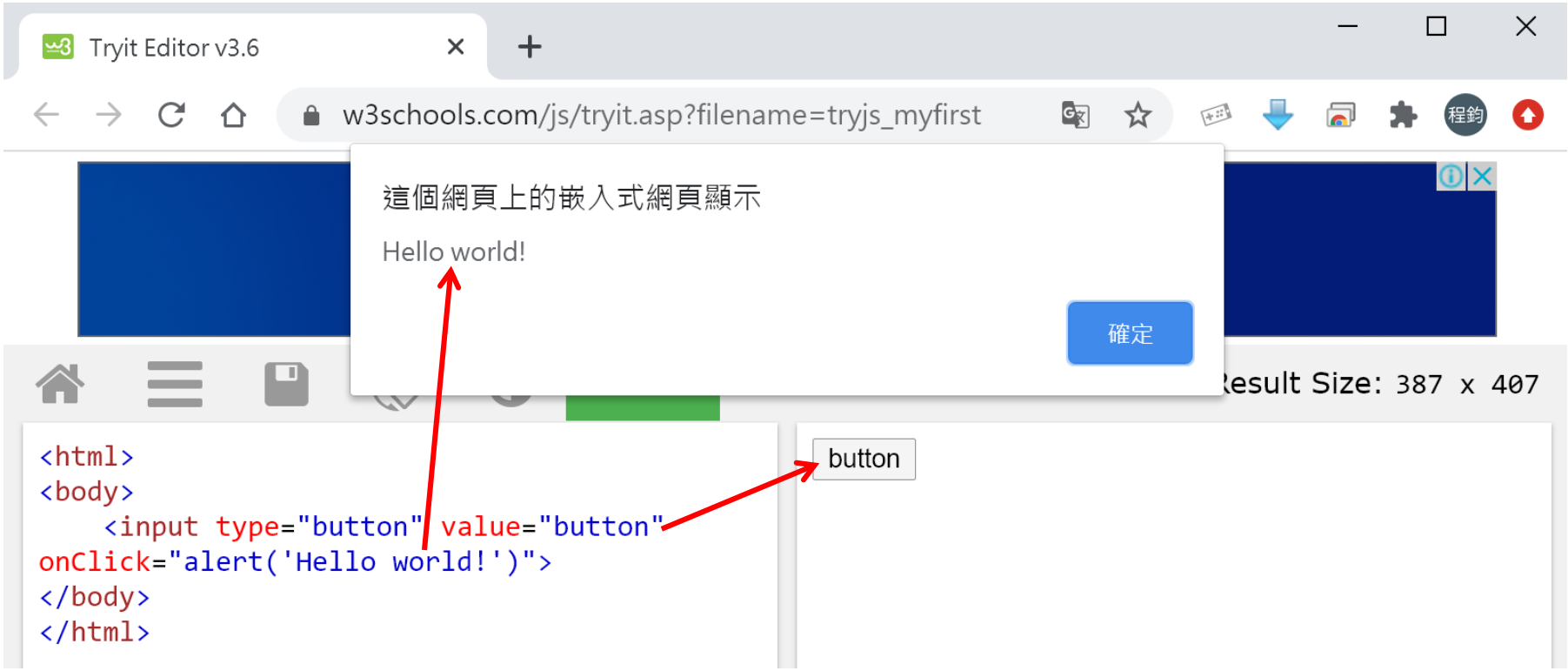

Tryit Editor v3.6
w3schools.com/js/tryit.asp?filename=tryjs_myfirst
這個網頁上的嵌入式網頁顯示
Hello world!
確定
FxPro
Trade Like a Pro
ial instruments involves
Result Size: 387 x 407
<html>
<body>
<P>This is a paragraph.</P>
<script>
alert("Hello world!")
</script>
</body>
</html>
This is a paragraph.


Tryit Editor v3.6
w3schools.com/js/tryit.asp?filename=tryjs_myfirst
這個網頁上的嵌入式網頁顯示
First alert
確定
Result Size: 387 x 407
<html>
<body>
<P>This is a paragraph.</P>
<script>
alert("First alert")
alert("Second alert")
</script>
</body>
</html>
This is a paragraph.
Tryit Editor v3.6
w3schools.com/js/tryit.asp?filename=tryjs_myfirst
這個網頁上的嵌入式網頁顯示
Second alert
確定

Dialog – Alert (pop-up when loading)



Dialog – Alert (trigger by button)



Exercise 1

Create several buttons on a webpage

Click the buttons to pop up different alert messages

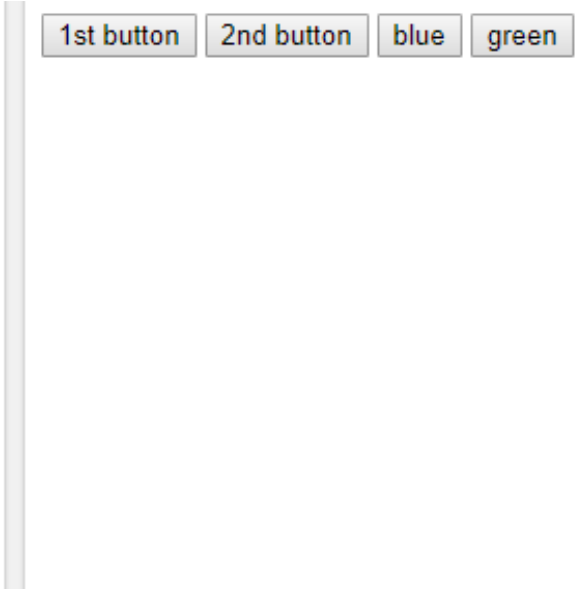
Click the buttons to change the background colors

```
<input type="button" value="blue"
onclick="document.body.style.backgroundColor = 'blue';">
```

```
<html>
<body>
    <input type="button" value="yellow"
onClick="document.body.style.backgroundColor
= 'yellow'">
</body>
</html>
```

Exercise 1

```
<html>
<body>

<input type="button" value="1st button" onClick="alert('wahaha!')">

<input type="button" value="2nd button" onClick="alert('muhaha!');">

<input type="button" value="blue"
 onclick="document.body.style.backgroundColor = 'blue';">

<input type="button" value="green"
 onclick="document.body.style.backgroundColor = 'green';">

</body>
</html>
```



Concatenation Operator

The + operator (called “concatenation operator”) joins multiple strings into a single string

```
<html>
<body>
    <script>
        var a = 10
        var b = 20
        var c = a + b
        var d = "a" + "b"
        document.write(c)
        document.write("<p>")
        document.write(d)
    </script>
</body>
</html>
```

```
30

ab
```

Escape Sequences

The backslash(\) combine with the next character to form an escape sequence.

- \n : New line
- \t : Horizontal tab
- \\ : Backslash
- \” : Double quote

Obtaining Input with prompt Dialogs

Scripting

- Generate part or all of a web page's content
- Dynamic web pages
  - Their content has the ability to change
- Allow the user to enter a value that the script can use

Obtaining Inputs with Prompt Dialogs


```
<html>
<body>
    <script>
        var name
        name = window.prompt("Please enter
your name!!!")

        document.writeln("<p>Hi " + name
+"!!</p>")
    </script>
</body>
</html>
```

Hi kevin!!

Exercise 2

Good Evening, Jim, welcome to JavaScript programming!

Declare a variable

JavaScript **does not** require variables to have a type

- Automatically converts between values of different types
- Loosely typed language

Undefined value

- If variable is declared but is not given a value
- Attempting to use the value of such a variable will be a logic error
- To indicate that a variable does not contain a value, you can assign the value null to it

Variable Names

- **Keywords**: words with special meanings in JavaScript
- All variables should be declared with a var statement before they are used
  - Valid format: consisting of letters, digits, underscores (_) and dollar signs ($)
  - Does not begin with a digit
  - Is not a reserved JavaScript keyword
- First word: lowercase first letter
- Subsequent word: begins with capital first letter
  - itemPrice, firstNumber

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#Keywords

Declarations and Comments

- Declarations end with a semicolon (;)
  - Can be split over several lines, with each variable in the declaration separated by a comma

```
var x = 5;
var y = 6;
var z = x + y;

var x = 5, y = 12, z = x + y;
```

- Comments
  - A single-line comment : //
  - Multiline comments : /* … */

```
// single-line comment

/*
    multiline comments
    Multiline comments
*/
```

Get User's Inputs

The sum is 444

```
<script>

   var firstNumber; // first string entered by user
   var secondNumber; // second string entered by user
   var number1; // first number to add
   var number2; // second number to add
   var sum; // sum of number1 and number2

   // read in first number from user as a string
   firstNumber = window.prompt( "Enter first integer" );

   // read in second number from user as a string
   secondNumber = window.prompt( "Enter second integer" );

   // convert numbers from strings to integers
   number1 = parseInt( firstNumber );
   number2 = parseInt( secondNumber );

   sum = number1 + number2; // add the numbers

   // display the results
   document.writeln( "<h1>The sum is " + sum + "</h1>" );

</script>
```

Date object

- Acquire the current local time
- Create a new instance of an object by using the `new` operator
  - followed by the type of the object, `Date`, and a pair of parentheses

creates a new Date object

```
<script>

   var name; // string entered by the user
   var now = new Date(); // current date and time
   var hour = now.getHours(); // current hour (0-23)

   // read the name from the prompt box as a string
   name = window.prompt( "Please enter your name" );

   // determine whether it's morning
   if ( hour < 12 )
      document.write( "<h1>Good Morning, " );

   // determine whether the time is PM
   if ( hour >= 12 )
   {
      // convert to a 12-hour clock
      hour = hour - 12;

      // determine whether it is before 6 PM
      if ( hour < 6 )
         document.write( "<h1>Good Afternoon, " );

      // determine whether it is after 6 PM
      if ( hour >= 6 )
         document.write( "<h1>Good Evening, " );
   } // end if

   document.writeln( name +
      ", welcome to JavaScript programming!</h1>" );

</script>
```

lastModified Property

document.lastModified

- Get the date and time the current document was last modified (e.g., shown on your personal website)

```
<script>
    var x = document.lastModified;
    document.write( x);
</script>
```

```
<script>
    document.write( "last modified: " + document.lastModified );
</script>
```

Decision Making: Equality and Relational Operators

- Equality operators both have the same level of precedence
- Lower than the precedence of the relational operators.
- The equality operators associate from left to right.

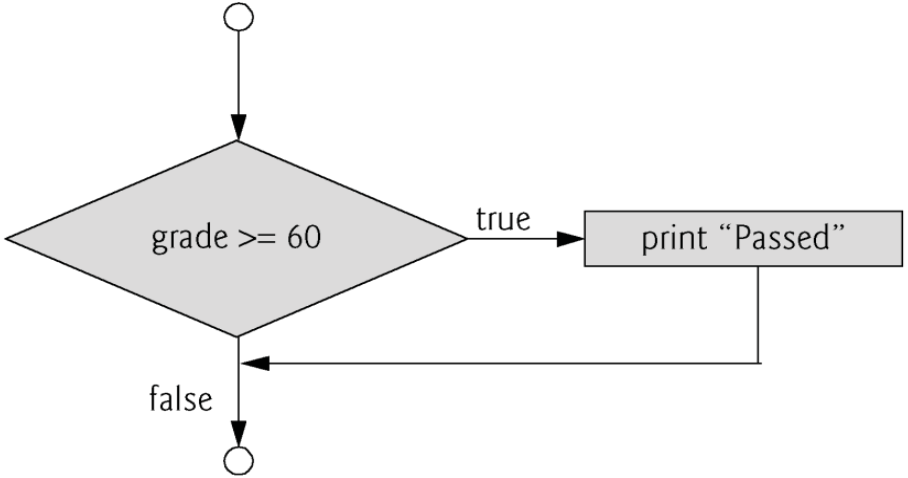
==	A is equal to B
!=	A is not equal to B
>	A is greater than B
<	A is less than to B
>=	A is greater than or equal to B
<=	A is less than or equal to B

Control Statements

- for: loops through a block of code a number of times
- if statement
  - either performs (selects) an action if a condition is true or skips the action if the condition is false
- if-else statement
  - performs an action if a condition is true and performs a different action if the condition is false
- while: loops through a block of code while a specified condition is true

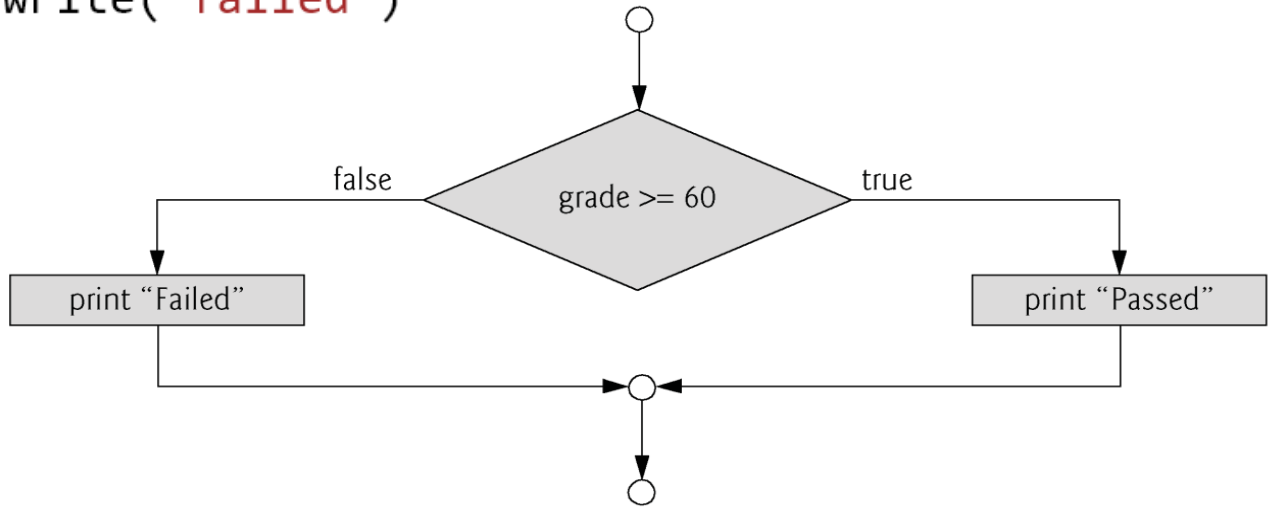
if Selection Statement

```
if(grade >= 60){
    document.write('passed')
}
```



if-else Selection Statement

```
if(grade >= 60){
    document.write('passed')
}else{
    document.write('failed')
}
```



`else if` Selection Statement

```
if (time < 10) {
    greeting = "Good morning";
} else if (time < 20) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

If less than 10, the result will be Good morning

If time = 15, the result will be Good day

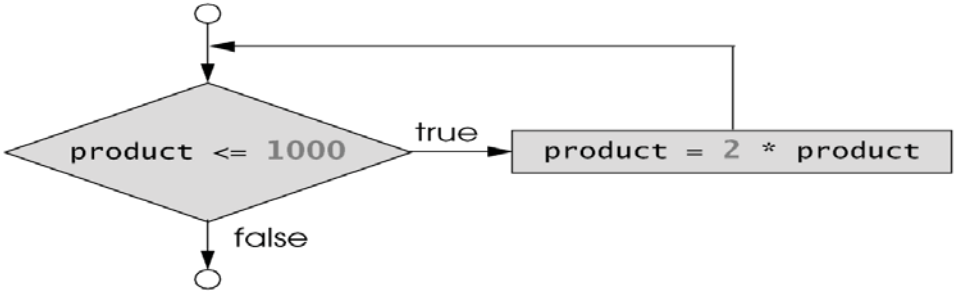
`else if` Selection Statement

- Braces `{ }`
  - Include several statements
- Nested `if-else`
  - Placing `if-else` statements inside other `if-else` statement

```
if (grade > 60) {
     if(grade > 90) {
     score = "Excellent";
     } else {
     score = "Passed";
     }
} else {
     score = "Failed";
}
```

`while` Repetition Statement

```
while (product <= 1000) {
     product = 2 * product;
}
```



JavaScript: Functions

Program Modules in JavaScript

- New functions that you write
  - triggered by its name, counter( ) { … }, output( ) { … }
- Prepackaged functions and objects
  - also triggered by its name
  - such as Math.random(), parseInt(), Date(), getTime(), document.write()

Programmer-Defined Function

- `Return` statement
  - passes information from inside a function back to the point in the program where it was called
- The format of a function definition is

function *function-name*(*parameter-list*)

{

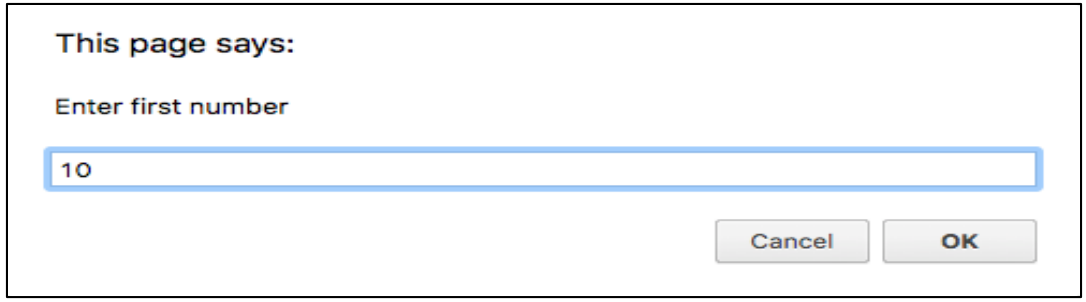
declarations and statements

}

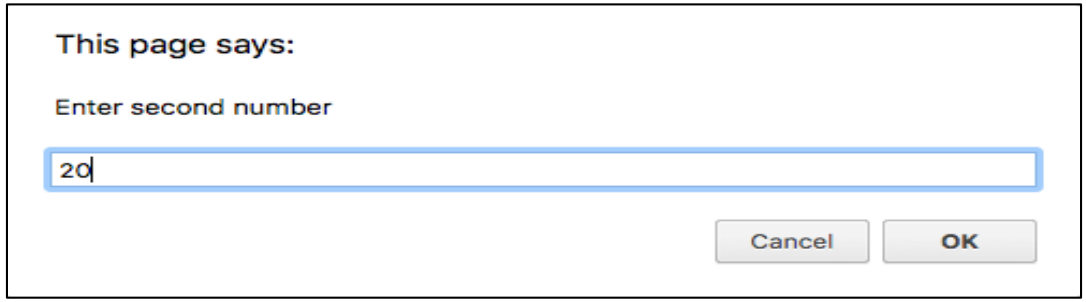
Programmer-Defined Function

- Executing the "return *expression*;" to return the value of expression to the caller
  - control returns to the point at which the function was invoked

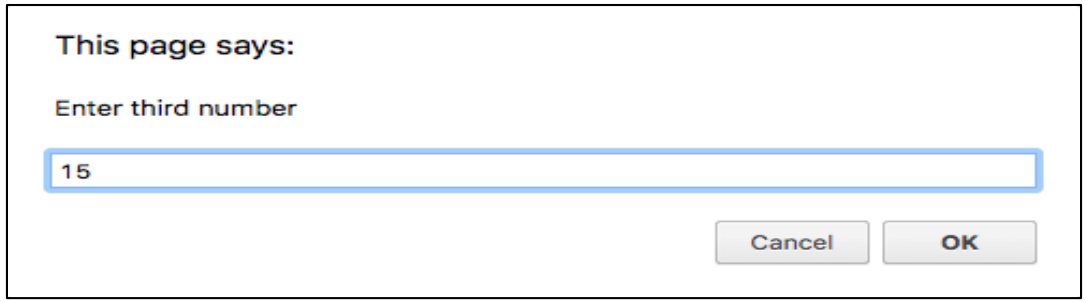
```
function myFunction(p1, p2) {
    return p1 * p2;              // The function returns the product of p1 and p2
}
```



Compare and get the max number



First number: 10
Second number: 20
Third number: 15
Maximum is: 20



```
<script>

   var input1 = window.prompt( "Enter first number", "0" );
   var input2 = window.prompt( "Enter second number", "0" );
   var input3 = window.prompt( "Enter third number", "0" );

   var value1 = parseFloat( input1 );
   var value2 = parseFloat( input2 );
   var value3 = parseFloat( input3 );

   var maxValue = maximum( value1, value2, value3 );

   document.writeln(
     "<p>First number: " + value1 + "</p>" +
      "<p>Second number: " + value2 + "</p>" +
      "<p>Third number: " + value3 + "</p>" +
      "<p>Maximum is: " + maxValue + "</p>" );

   // maximum function definition (called from line 22)
   function maximum( x, y, z )
   {
      return Math.max(x, y, z);
   } // end function maximum

</script>
```

function call

return value

Random Number Generation

- Math.random() ;
  - Returns a random number between 0 (inclusive) and 1 (exclusive)
  - Return a value range: 0.00...01 < num < 0.99..99
- Generate and return a random number range: 1.0 < num < 11.0
  - Math.random() * 10 + 1;
- Generate and return an Integer random number between 1 and 10
- Math.floor() function returns the largest int less than/equal to a given num
  - Math.floor((Math.random() * 10)); //0-9
  - Math.floor((Math.random() * 10) + 1); //1-10

```
<script>
document.write('<p>' + Math.floor(Math.random() * 10 + 1) + '</p>')
</script>
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_random

```
<script>

   var value;

   document.writeln( "<p>Random Numbers</p><ol>" );

   for ( var i = 1; i <= 30; ++i )
   {
      value = Math.random() * 10 +1;
      document.writeln( "<li>" + value + "</li>" );
   } // end for

   document.writeln( "</ol>" );

</script>
```

Random Numbers

4.173754462006356
10.030484058871902
1.5825914985236325
7.30658642804410 5
2.4287343117438733

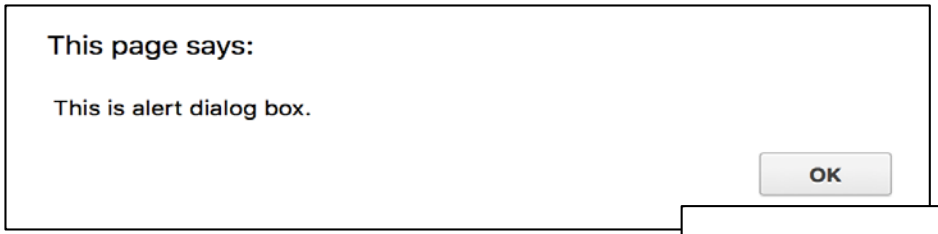
```
<script>

   var value;

   document.writeln( "<p>Random Numbers</p><ol>" );

   for ( var i = 1; i <= 30; ++i )
   {
      value = Math.floor(Math.random() * 10 + 1);

      document.writeln( "<li>" + value + "</li>" );
   } // end for

   document.writeln( "</ol>" );

</script>
```

Random Numbers

3 3 10 4 9 6 9 4

Function

```
  <script>
    function simpleDemo( )
    {  alert(" This is alert dialog box. ")
       confirm(" This is confirm dialog box. ")
    }
  </script>
</head>

<body>
    <input type="button" name="" value=" start " onClick="simpleDemo( )" >
</body>
</html>
```



HW: due by Dec. 17, 11:59pm

JS -- Number guessing

Use the “window.prompt()” function for users to input a number to guess the correct number that is randomly selected from 1 to 10. The game will stop when the user guesses the number right. If the user guesses incorrectly, prompt another window for the user to continue guessing.

Tips :

1. Use “var rand = Math.floor((Math.random() * 10) + 1);” to randonly select a value from 1 to 10.
2. Use while loop to control the window.prompt() function.

Final Group Project- Preparation

1. Fill up the time table (**https://ppt.cc/fWz7Tx**)
2. Work on the info sheet (word file can be found on Moodle)
3. Bring the info sheet and attend the discussion next week (Dec. 18)
   - Be on time! (at least 5 mins early)
4. Based on the feedback, discuss with your teammates and work on the info sheet and poster
5. Submit the info sheet right after next week's class (Dec. 25, 11:59am)
   - Briefly review this course and feed you the details of final project presentation
   - Discuss your final project (info sheet & poster) with teammates, TAs or instructor
6. Submit the poster by Dec. 31, 1pm (#enjoyNewYearsEve)
   - Details will be introduced on Dec. 25

CH

科目代號(Course #)：306005001
科目名稱：計算機概論
Course Name：Introduction to Computer Science
授課教師：簡士鎰
Instructor：CHIEN SHIH-YI
系所：資管一甲、資管一乙
上課時間 (Session)：五23 (fri09-11)

EN

科目代號(Course #)：306005011
科目名稱：計算機概論
Course Name：Introduction to Computer Science
授課教師：簡士鎰
Instructor：CHIEN SHIH-YI
系所：資管一甲、資管一乙
上課時間 (Session)：五D5 (fri13-15)