



មជ្ឈមណ្ឌល ព័ត៌មានវិទ្យា អិនធឺណេត Enter Information Technology Center

អារម្ភកថា

សព្វថ្ងៃនេះ ការប្រើប្រាស់កុំព្យូទ័រ គឺជាមធ្យោបាយយ៉ាងសំខាន់មួយក្នុងការដោះស្រាយបញ្ហា និងជួយសំរួលដល់ការងារប្រចាំថ្ងៃរបស់មនុស្សដែលរស់នៅលើពិភពលោក។ ភាគច្រើនការងារ ស្ទើរតែទាំងអស់របស់ពួកគេគឺពឹងផ្អែកទៅលើការ ប្រើប្រាស់កុំព្យូទ័រទាំងស្រុង ដែលជាហេតុទាមទារឲ្យអ្នកប្រើប្រាស់ត្រូវមានសមត្ថភាព និងចំនេះដឹងគ្រប់គ្រាន់ក្នុងការប្រើប្រាស់ទៅលើវា។

ដើម្បីចូលរួមលើកស្ទួយវិស័យព័ត៌មានវិទ្យា នៅក្នុងព្រះរាជាណាចក្រកម្ពុជា មជ្ឈមណ្ឌលព័ត៌មានវិទ្យា អិនធឺណេត ត្រូវបានបង្កើតឡើងផងដែរ ក្នុងគោលបំណង ដើម្បីជួយបណ្តុះបណ្តាលចំនេះដឹងផ្នែកព័ត៌មានវិទ្យាដល់ សិស្ស និស្សិត និងអ្នកសិក្សា ឲ្យមានចំនេះដឹងពិតប្រាកដក្នុងផ្នែកនេះ ជាពិសេសមានសមត្ថភាពច្បាស់លាស់ក្នុងការប្រើប្រាស់កុំព្យូទ័រ។ ដូចនេះសូម សិស្ស និស្សិត និងអ្នកសិក្សាទាំងអស់ ខិតខំប្រឹងប្រែង សិក្សា ប្រាសាទ្រព្យ និងចាប់យកនូវចំនេះដឹង ដែលមជ្ឈមណ្ឌល បានបណ្តុះបណ្តាលជូន ដើម្បីពេលវេលាដែលអ្នកបានចំនាយ ក្លាយទៅជាប្រយោជន៍មួយយ៉ាងធំធេងសំរាប់ជីវិតរបស់អ្នកនាពេលអនាគត។

យើងខ្ញុំដែលជាអ្នករៀបរៀងនៃ មជ្ឈមណ្ឌលព័ត៌មានវិទ្យា-អិនធឺណេត សូមអរគុណចំពោះការគាំទ្ររបស់ សិស្ស និស្សិត និងអ្នកសិក្សាទាំងអស់ និងសូមអភ័យ ទោសរាល់កំហុសឆ្គងដែលកើតមានក្នុងករណីណាមួយ ហើយយើងខ្ញុំ នឹងខិតខំរិះរកនូវអ្វីដែលថ្មីក្នុងផ្នែកព័ត៌មានវិទ្យានេះ ដើម្បីផ្តល់ជូនដល់ អ្នកសិក្សាបន្ថែមទៀត សូមអរគុណ និងសូមជំរាបសួរ។

ក្រុមអ្នករៀបរៀងនៃ
មជ្ឈមណ្ឌលព័ត៌មានវិទ្យា អិនធឺណេត

ក្រុមអ្នករៀបរៀង

អ្នករៀបរៀង:

ហ៊ុ ម៉ូឌី

អ្នកចេញគំរូ:

ហ៊ុ ម៉ូឌី

អ្នកវាយអត្ថបទ និងរៀបចំរូបភាព:

ច័ន្ទធាវិទូ

គិល ភក្ដី

ចិន សុវណ្ណ

កែសម្រួលអត្ថបទនិងអក្ខរកម្ម:

ច័ន្ទធាវិទូ

គិល ភក្ដី

ចិន សុវណ្ណ

រៀបរៀងលើកទី ២:

រក្សាសិទ្ធិគ្រប់យ៉ាង © 2013 ដោយ មជ្ឈមណ្ឌលព័ត៌មានវិទ្យា អិនធឺណេត សំរាប់ជា ឯកសារប្រើប្រាស់ផ្ទៃក្នុង។
គ្មានផ្នែកណាមួយនៃសៀវភៅនេះត្រូវបាន ផលិតឡើងវិញ ទោះជាមធ្យោបាយណាមួយដោយ នៅពេលដែលគ្មានការ
អនុញ្ញាតិជាលាយលក្ខណ៍អក្សរ ពីម្ចាស់កម្មសិទ្ធិ។

អាស័យដ្ឋាន:

ផ្ទះលេខ 179 ផ្លូវ 173 សង្កាត់ ទំនប់ទឹក ខណ្ឌចំការមន រាជធានីភ្នំពេញ

Tel: [010-012-016] 603 314

Website: www.enteritc.com | facebook.com/enteritc

មេរៀនទី 1: ការណែនាំ JavaScript

1. និយមន័យ:

JavaScript គឺជាប្រភេទ script language មួយសំរាប់ឲ្យយើងប្រើប្រាស់ភ្ជាប់ជាមួយនឹង HTML ដើម្បីធ្វើការបង្កើត Interactive Web Pages ។ Script language គឺជា Programming Language ដែលមានសមត្ថភាពលើសពី Markup Language (ដូចជា HTML, CSS) ដែលអាចធ្វើការឲ្យ Webpage មានលក្ខណៈជា Dynamic ។ ចំពោះ HTML + CSS គឺត្រូវបានហៅថា Static Webpage ឬ HTML ប៉ុន្តែប្រសិន HTML + CSS + JavaScript គឺត្រូវបានហៅថាជា Dynamic Webpages ឬ DHTML ។ JavaScript គឺជា Script ដែលធ្វើការនៅលើ Web Browsers ដូច្នេះវាត្រូវបានហៅថា Client-side scripting language ផងដែរ។

Dynamic Webpages គឺមាន features ជាច្រើនដូចជា:

- > ដំនើការ នៃ Process គណនាណាមួយ
- > Pop-up message នៅក្នុង Input form ដើម្បីផ្ទៀងផ្ទាត់មើលទិន្នន័យដែលបញ្ចូលក្នុង Form
- > ធ្វើឲ្យ Image ផ្លាស់ប្តូរនៅពេលដែលយើងរំកិល Mouse លើវា
- > រូបភាពមានចលនាជាលក្ខណៈ: Slideshow animations ,

2. JavaScript and HTML:

យើងអាចនិយាយបានថា JavaScript គឺជា extension to HTML (បន្ថែមសមត្ថភាពឲ្យ HTML) ឬក៏ជា add-on របស់ HTML ។ យើងប្រើប្រាស់ HTML tags ដើម្បីធ្វើការបង្កើត objects ហើយប្រើប្រាស់ JavaScript បន្ថែមដើម្បីកំណត់ការងារបន្ថែមទៀតទៅឲ្យ Tags នោះ។

ឧទាហរណ៍: យើងប្រើប្រាស់ HTML <body></body> tag ដើម្បីបង្កើត Web page បន្ទាប់មកយើងអាច Interact (ភ្ជាប់) វាជាមួយនឹង JavaScript ដោយបង្កើតជា Event Onload មួយដើម្បីបង្ហាញ Message ស្វាគមន៍នៅពេល Web page នោះ ត្រូវបានបើក។

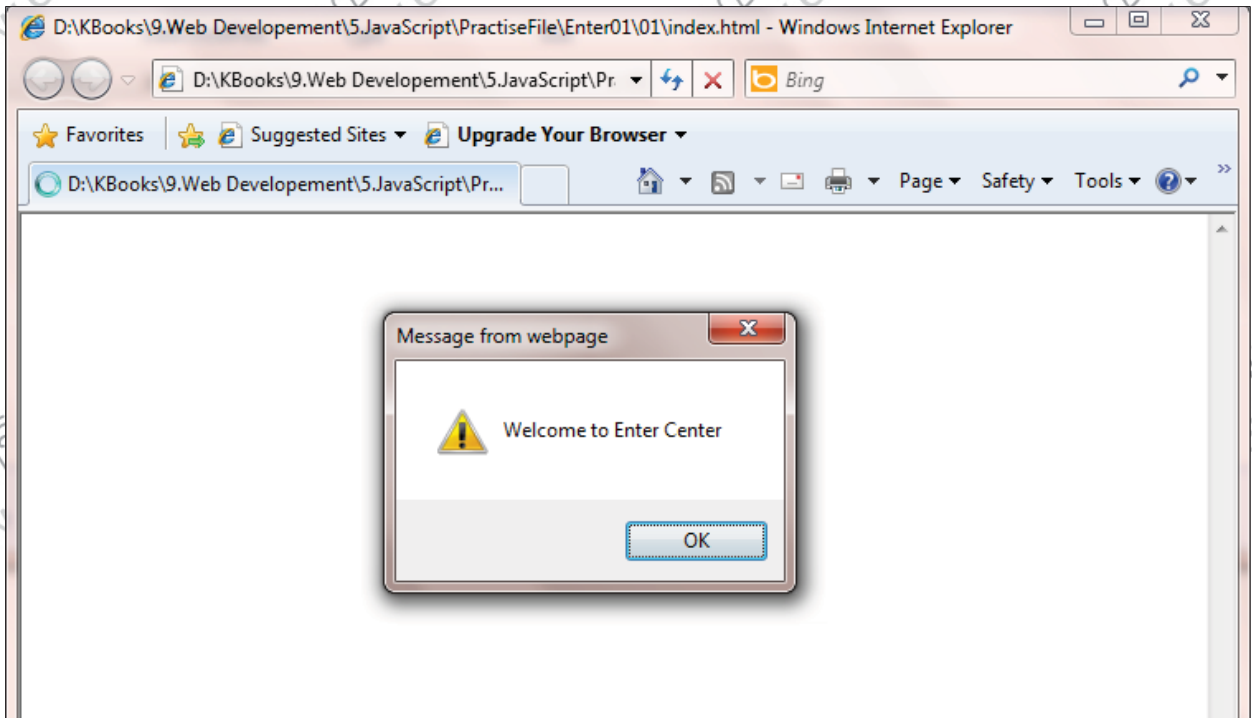
Ex:

```

<html>
<body>
<script type="text/javascript">
alert('Welcome to Enter Center')
</script>
<h1>Hello World</h1>
</body>
</html>

```

Output:



3. ការបកស្រាយពីកូដរបស់ JavaScript and HTML:

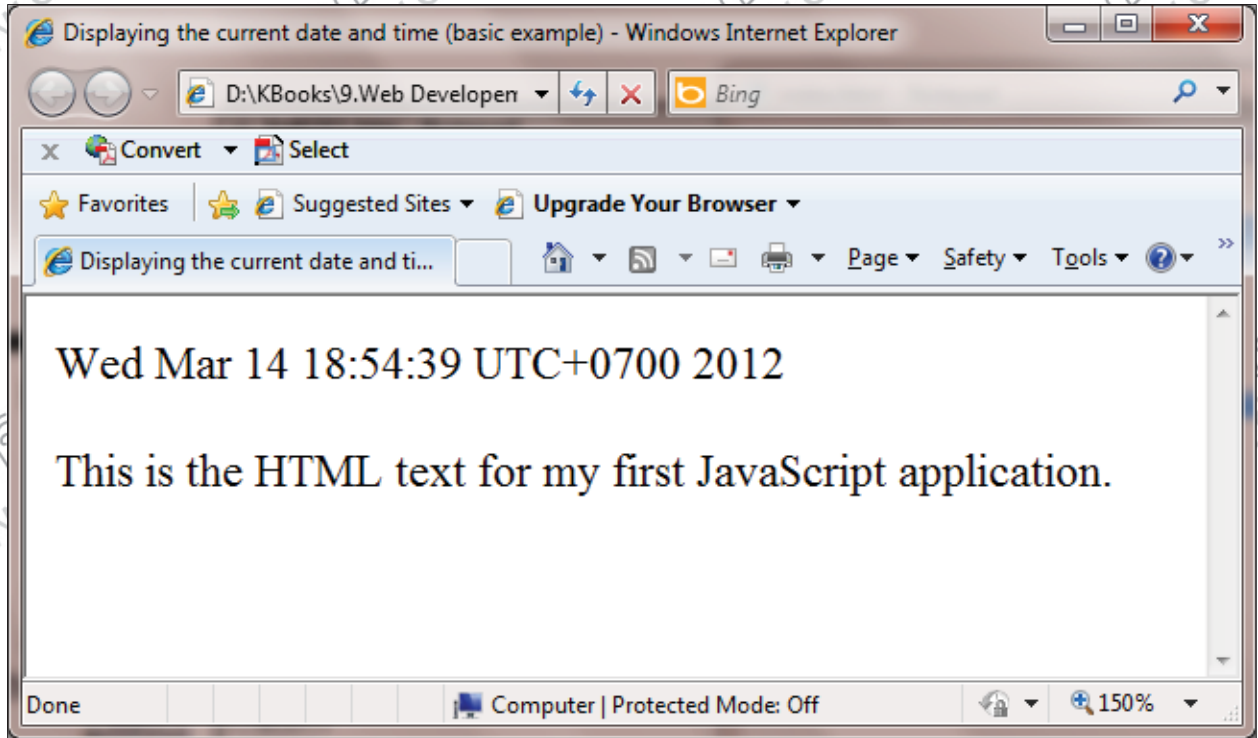
Ex:

```

<html>
<head>
<title>Displaying the current date and time (basic
example)</title>
<script language="JavaScript" type="text/javascript">
<!-- Hide from browsers that do not support JavaScript
// Capture the current date and time from the system clock
var todays_date = new Date();
// Display the current date and time on the Web page
document.writeln(todays_date);
// --> Finish hiding
</script>
</head>
<body>
<p>This is the HTML text for my first JavaScript
application.</p>
</body>
</html>

```

Output:



ការបកស្រាយ:

> គ្រប់ JavaScript កូដទាំងអស់ដែលស្ថិតនៅក្នុង HTML File គឺត្រូវស្ថិតនៅក្នុង <script></script> tags ជានិច្ច

```

<script language="JavaScript" type="text/javascript">
<!-- Hide from browsers that do not support JavaScript
// Capture the current date and time from the system clock
var todays_date = new Date();
// Display the current date and time on the Web page
document.writeln(todays_date);
// --> Finish hiding
</script>

```

> រាល់ JavaScript កូដដែលសរសេរនៅក្នុង script tag ត្រូវដាក់ស្ថិតនៅក្នុង Header Section (<head></head>) នៃ HTML File ជានិច្ច។ ទន្ទឹមនឹងនោះយើងក៏អាចធ្វើការដាក់ JavaScript នៅក្នុង Body Section (<body></body>) បានផងដែរ ប៉ុន្តែភាគច្រើនគេនិយមដាក់នៅក្នុង <head></head> ពីព្រោះដើម្បីឲ្យ Web Browser បានស្គាល់និង ធ្វើការជាមួយនឹង JavaScript ភ្លាមៗ នៅពេល Web pages ត្រូវបាន Load ។

```

<head>
<title>Displaying the current date and time (basic
example)</title>
<script language="JavaScript" type="text/javascript">
<!-- Hide from browsers that do not support JavaScript
// Capture the current date and time from the system clock
var todays_date = new Date();
// Display the current date and time on the Web page
document.writeln(todays_date);
// --> Finish hiding
</script>
</head>

```

> HTML comments (<!--JavaScript //-->) អាចត្រូវបានដាក់បំបាំង JavaScript ដើម្បីធ្វើការ hide JavaScript ដើម្បីការពារជាមួយនឹង Web Browsers ចាស់ៗ ដែលមិន Support ជាមួយនឹង JavaScript បន្ថែមចំពោះ Web Browsers Version ថ្មីៗ គឺមិនមាន បញ្ហាអ្វីឡើយ ទោះបីជា HTML comments ត្រូវបានដាក់ ឬមិនបានដាក់ក៏ដោយ។

```

<!-- Hide from browsers that do not support JavaScript
// Capture the current date and time from the system clock
var todays_date = new Date();
// Display the current date and time on the Web page
document.writeln(todays_date);
// --> Finish hiding

```

> យើងអាចប្រើប្រាស់បន្ថែមជាមួយនឹង <noscript></noscript> ដើម្បីបង្ហាញព័ត៌មានប្រាប់ដល់ users ដែលបានប្រើប្រាស់ Web Browsers ចាស់ៗ មិន Support ជាមួយ JavaScript ឲ្យបានដឹងផងដែរ។

```

</script>
<NOSCRIPT>
You must be running a JavaScript-enabled Web browser, such
as the latest version of Microsoft Internet Explorer or
Netscape Navigator, to get the most from this Web page.
</NOSCRIPT>

```

> JavaScript Comments អាចត្រូវបានដាក់បន្ថែមជាមួយ JavaScript ក្នុងដើម្បីដាក់ជា កំនត់សំគាល់ខ្លីៗ សំរាប់ឲ្យ Web Developer ងាយស្រួលចងចាំ ហើយវាមិនត្រូវបាន display ទៅក្នុង Web Browser នោះទេ។ Comments ត្រូវបានបែងចែកជាពីរប្រភេទគឺ Single-line Comment (//) និង Multiple-line Comment (/* */) ។

```

// Single-line comments don't require an ending slash.

/* This comment can span multiple lines. Always remember
to close it, though; if you forget, you'll get weird errors
when you try to display your script. */

```

4. ទីតាំងនៃការសរសេរ JavaScript

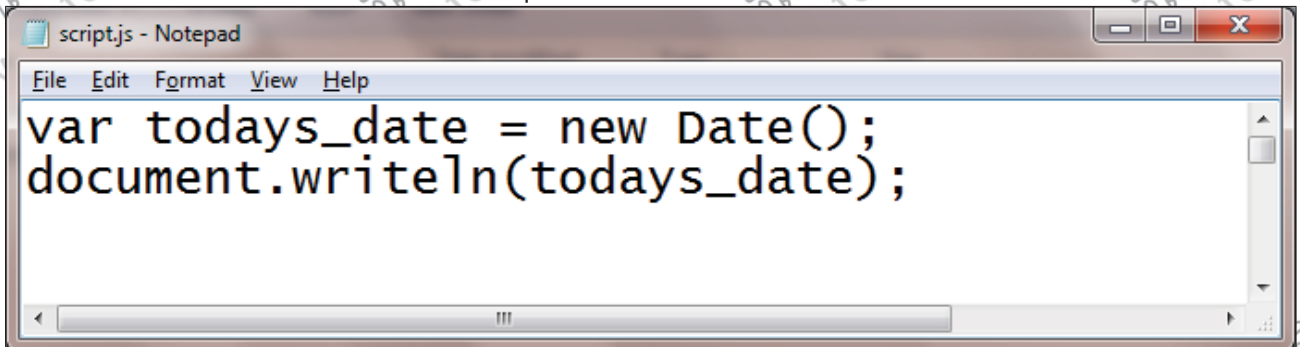
យើងអាចសរសេរ JavaScript ជាមួយនឹង HTML File ដោយផ្ទាល់តែម្តងក៏បាន នៅក្នុង Header Section ដែលគេហៅថាជា Internal ឬក៏យើងអាចធ្វើការសរសេរ JavaScript នៅលើ File ផ្សេងជាចំរើមពី HTML File ផងដែរ ដែលគេហៅថា External ។

ក្នុងនោះជំហាននៃការសរសេរកូដ JavaScript ជា External (នៅលើ File ផ្សេងពី HTML File) រួមមាន:

- > សរសេរកូដ JavaScript នៅលើ Text Editor ហើយ Save វាជា *.js file extension
- > បន្ទាប់មកស្វែង link វាទៅក្នុង HTML File ជាមួយនឹង tag <script type="text/javascript" src="?"> ដែល ? ជាឈ្មោះរបស់ JavaScript File បន្ទាប់មកស្វែងបិទ </script> ។

Ex:

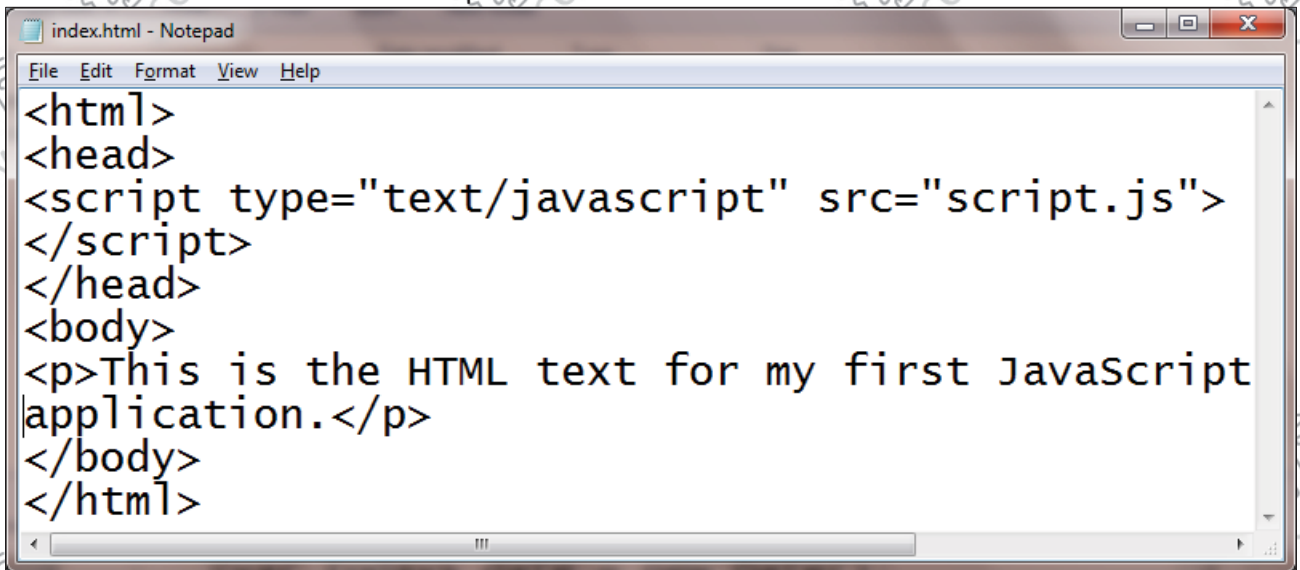
> ខាងក្រោមនេះ JavaScript កូដដែលសរសេរក្នុង File ឈ្មោះ script.js



```

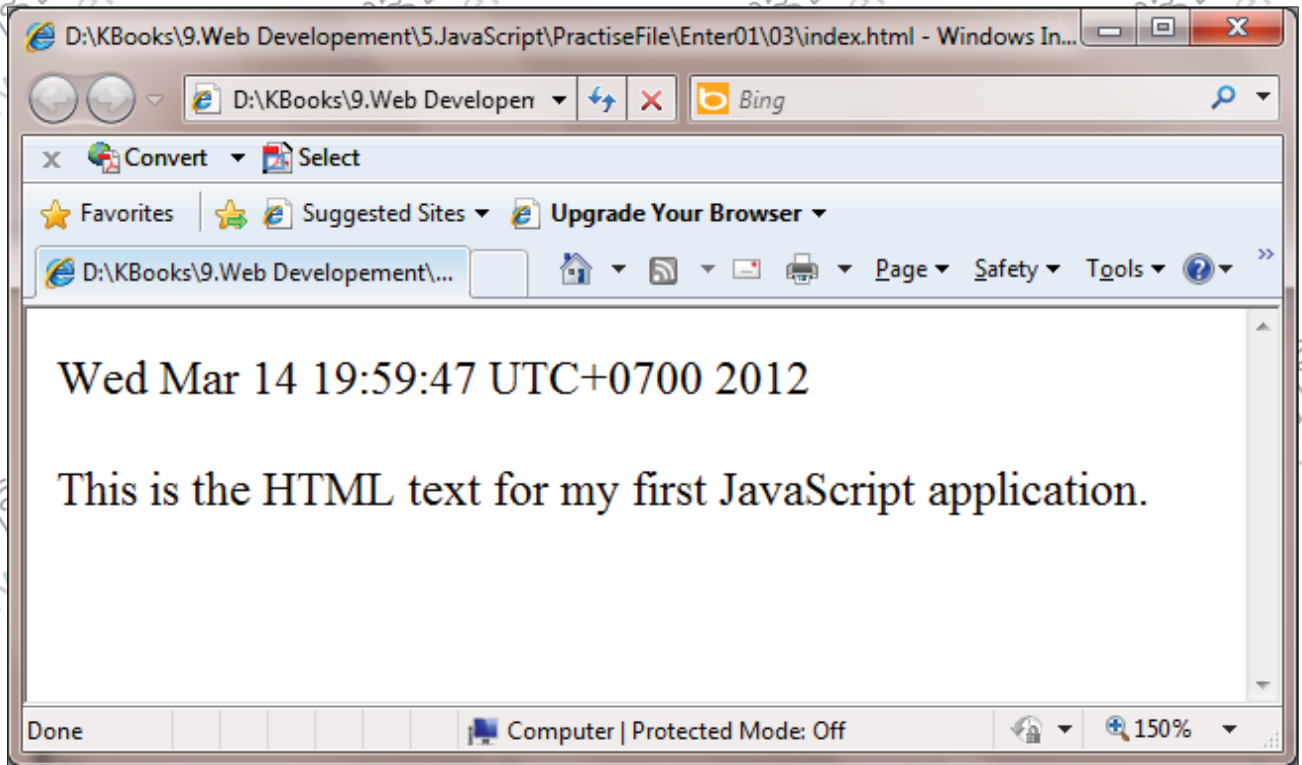
script.js - Notepad
File Edit Format View Help
var todays_date = new Date();
document.writeln(todays_date);
  
```

> ខាងក្រោមនេះ HTML កូដដែលសរសេរក្នុង File ឈ្មោះ index.html

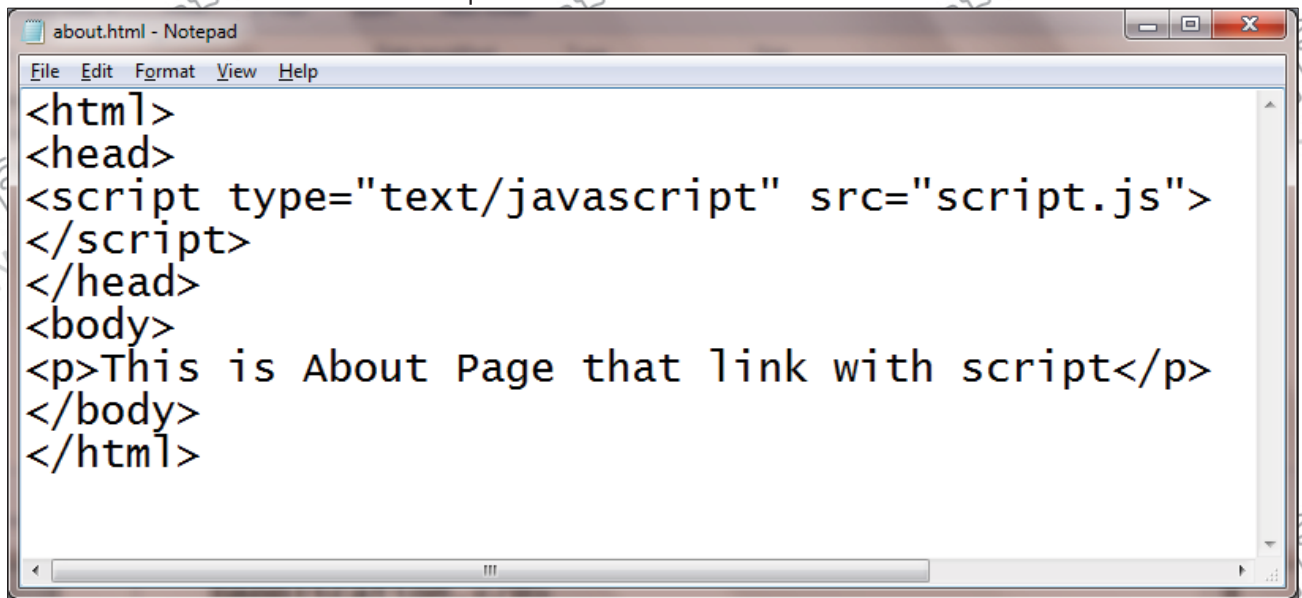


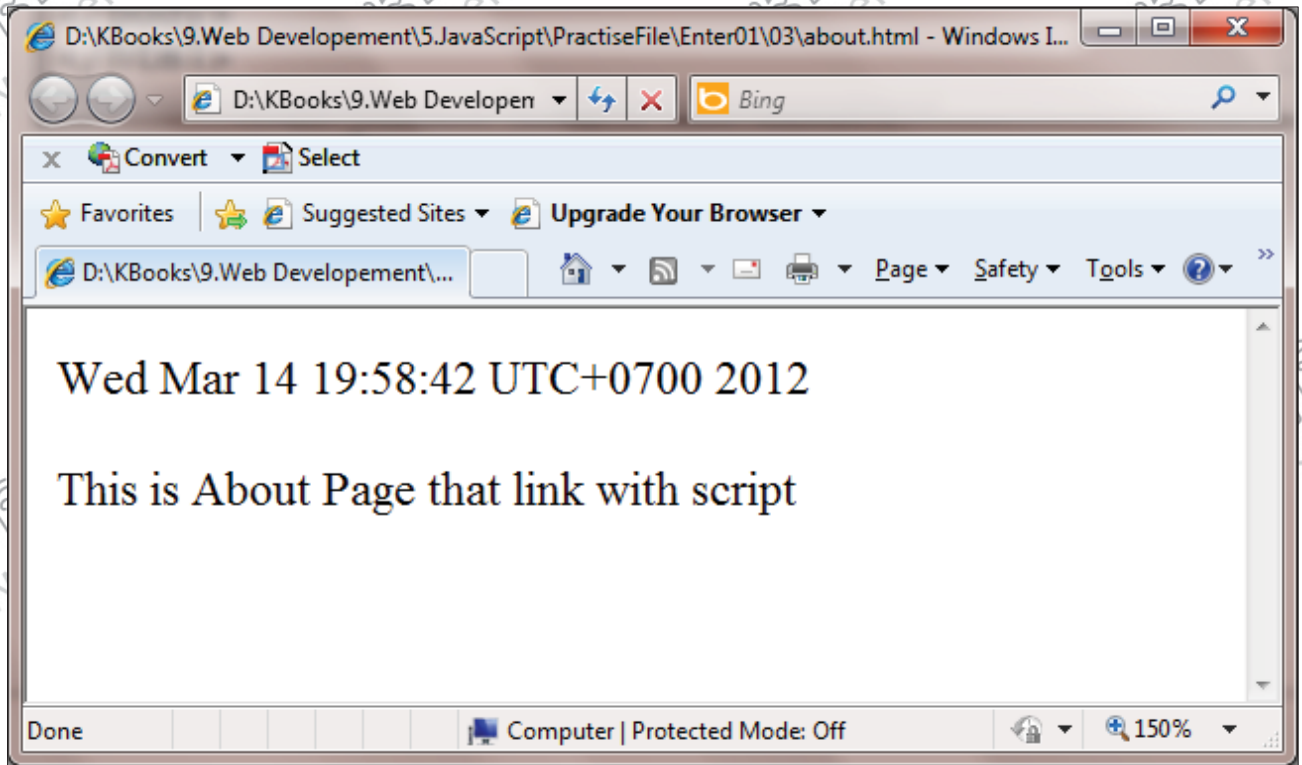
```

index.html - Notepad
File Edit Format View Help
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>This is the HTML text for my first JavaScript
application.</p>
</body>
</html>
  
```



> ខាងក្រោមនេះ HTML កូដដែលសរសេរក្នុង File ឈ្មោះ about.html





5. លំហាត់:

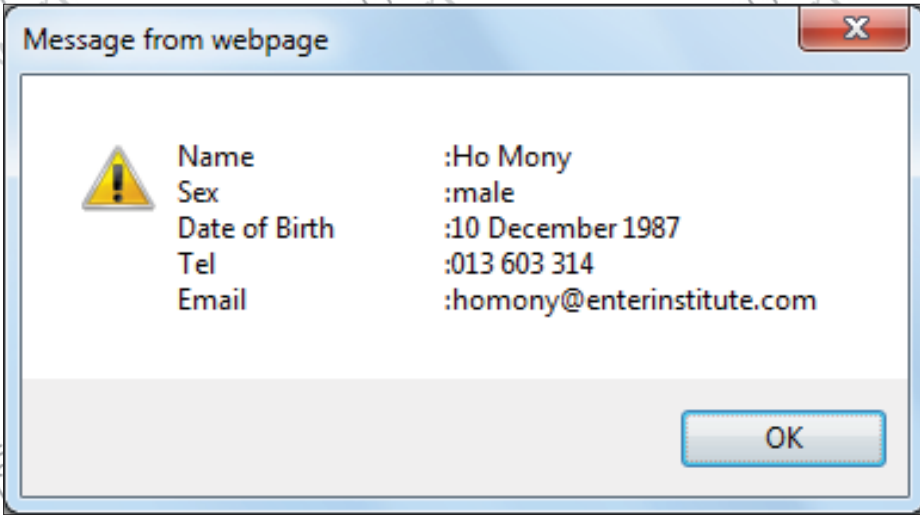
ចូរសរសេរ code មួយដើម្បី display ព័ត៌មានមួយចំនួនដូចខាងក្រោម:

- > ឈ្មោះរបស់អ្នក
- > ភេទរបស់អ្នក
- > ឆ្នាំកំណើត
- > លេខទូរស័ព្ទ
- > Email

```

Name           :Ho Mony
Sex            :Male
Date of Birth  :10 December 1987
Tel           : 013 603 314
Email         : homony@enterinstitute.com
  
```

Output:



មេរៀនទី 2: JavaScript Language Basics

1. Statement:

Statement គឺជាក្បួន ឬឃ្លាបញ្ជាទៅកាន់ Browsers ធ្វើការងារជាក់លាក់ណាមួយ។ Statements ជាច្រើនត្រូវបាន បែងចែកជាថ្មីគ្នាដោយប្រើប្រាស់ ការចុះបន្ទាត់ ឬប្រើប្រាស់សញ្ញា semicolon (;)។

> ការប្រើប្រាស់ ការចុះបន្ទាត់:

```
alert('Welcome')
alert('JavaScript')
```

> ការប្រើប្រាស់ Semicolon:

```
alert('Welcome');alert('JavaScript');
```

ប៉ុន្តែដើម្បីងាយស្រួល សំរាប់ Web Developer ឬ Programmer នោះគឺគេនិយមប្រើ Semicolon និង ចុះបន្ទាត់ជាមួយ គ្នាតែម្តង។

```
alert('Welcome');
alert('JavaScript');
```

Ex:

```
alert('Welcome');
alert('JavaScript');
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<h1>01.Statments</h1>
</body>
</html>
```

2. Comment:

JavaScript Comments អាចត្រូវបានដាក់បន្ថែមជាមួយ JavaScript ក្នុងដើម្បីដាក់ជា កំនត់សំគាល់ខ្លីៗ សំរាប់ឲ្យ Web Developer ងាយស្រួលចងចាំនៅពេលក្រោយ ហើយវាមិនត្រូវបាន display ទៅក្នុង Web Browser នោះទេ។
Comments ត្រូវ បានបែងចែកជាពីរប្រភេទគឺ Single-line Comment (//) និង Multiple-line Comment (/* */) ។

```
// Single-line comments don't require an ending slash.

/* This comment can span multiple lines. Always remember
to close it, though; if you forget, you'll get weird errors
when you try to display your script. */
```

JavaScript Comments មានវិសាលភាពប្រើប្រាស់បាននៅក្នុង <script></script> របស់តំបន់ JavaScript Code តែប៉ុណ្ណោះ។ យើងមិនអាចប្រើប្រាស់ JavaScript Comments ជាមួយនឹង HTML Code បាននោះទេ។
យ៉ាងណាមិញ ចំពោះ HTML Comment វិញ <!-- Comment here --> គឺអាចប្រើប្រាស់បានជាមួយនឹង HTML Code តែប៉ុណ្ណោះ <body></body> ដោយមិនអាចយកវាទៅប្រើប្រាស់នៅក្នុងតំបន់របស់ JavaScript Code នោះទេ។

3. Line Break:

ដើម្បីធ្វើការចុះបន្ទាត់នោះ គឺត្រូវប្រើប្រាស់ Escape Character (\n) នៅត្រង់តំបន់នៃ String ណាដែលត្រូវការ ។

```
alert("Hello\nHow are you?");
```

4. Insert Spacial Character:

យើងអាចប្រើប្រាស់ Back Slash (\) ជាមួយនឹង Special Character មួយចំនួនទៀតដើម្បីធ្វើការ display អក្សរមួយចំនួនដែលមិន Support នៅក្នុង Double Quote String (" ") ។

Code	Outputs
'	single quote
"	double quote
\\	backslash
\n	new line
\t	tab
\b	backspace

Ex:

```
var txt="We are the so-called \"Vikings\" from the
north.";
document.write(txt);
```

5. ការប្រើប្រាស់ **document.write**:

document.write គឺជាការ display ព័ត៌មានចូលទៅកាន់ Webpage ទៅតាមតំរូវការដែលចង់បាន។

Ex:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Web Page</h1>
<script type="text/javascript">
document.write("<p>My First JavaScript</p>");
</script>

</body>
</html>
```

Output:



6. ការប្រើប្រាស់ Prompt Method:

Prompt គឺជា Method មួយដែលត្រូវបានប្រើប្រាស់ដើម្បីចាប់យកទិន្នន័យតាមរយៈការ Input តាម Keyword ដោយ

User ។

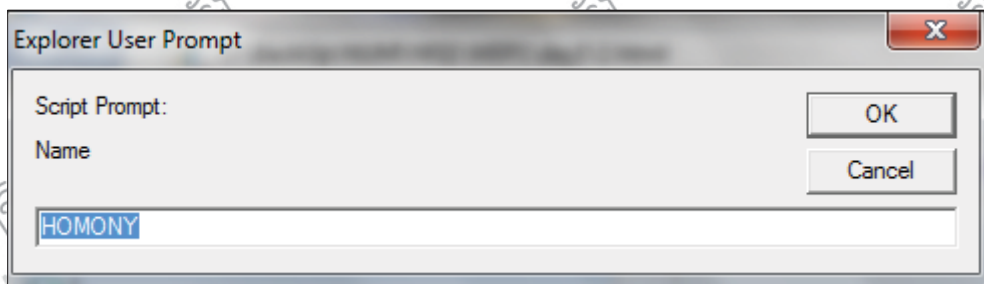
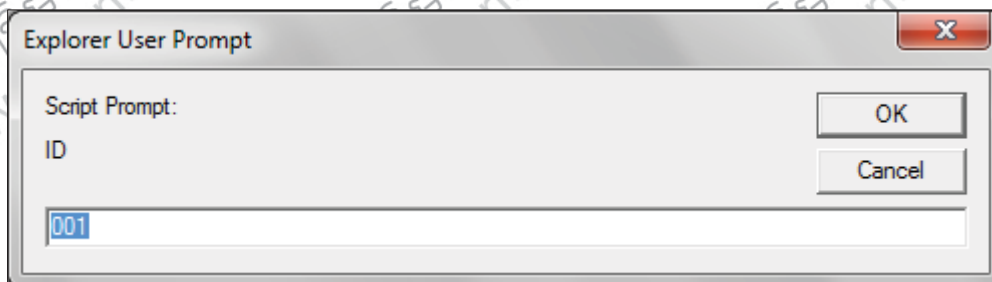
Ex:

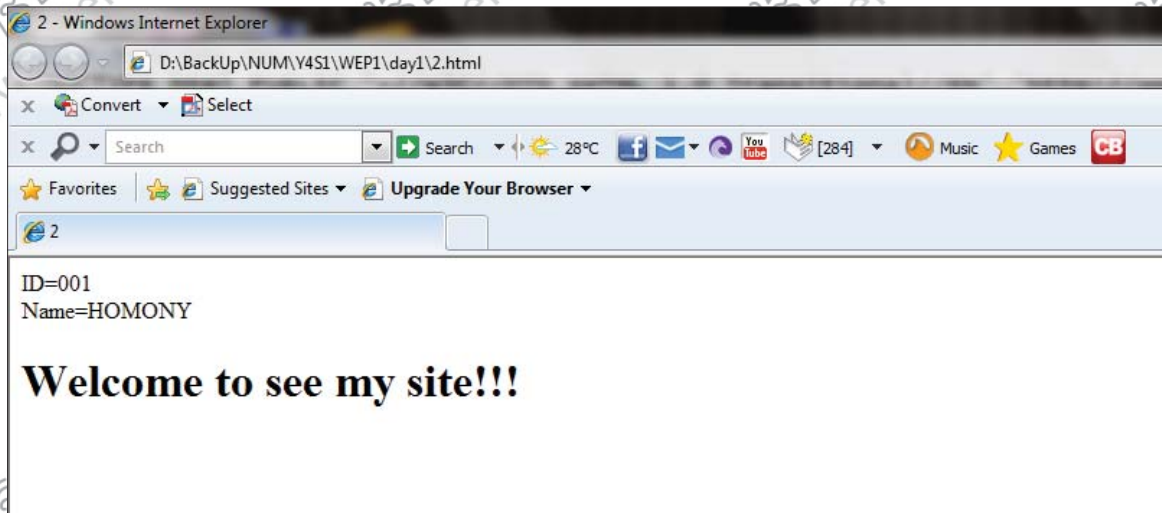
```

<html>
<head>
<title>Practise</title>
<script language="JavaScript">
var id,name;
id=prompt("ID","001");
name=prompt("Name","HOMONY");
document.write("ID="+ id + "<br/> Name=" + name);
</script>
</head>
<body>
<h1> Welcome to see my site!!!</h1>
</body>
</html>

```

Output:





7. ការប្រើប្រាស់ Confirm Box Method:

Confirm Box គឺត្រូវបានប្រើប្រាស់ដើម្បីកំណត់ User ជ្រើសរើសយកដំរើសណាមួយនៅលើ Message Box ដែលបានបង្ហាញ។

Ex:

```
var r=confirm("Press a button");
if (r==true)
{
x="You pressed OK!";
}
else
{
x="You pressed Cancel!";
}
```

8. របៀបបង្ហាញទំហំរបស់ Screen:

Ex:

```
<script language="javascript">
document.write('Your screen width and height are ' +
screen.width+ ' and ' + screen.height+ '.')
</script>
```

9. របៀបចុះបញ្ជីក្នុង String:

Ex:

```
document.write("Hello \
World!");
```

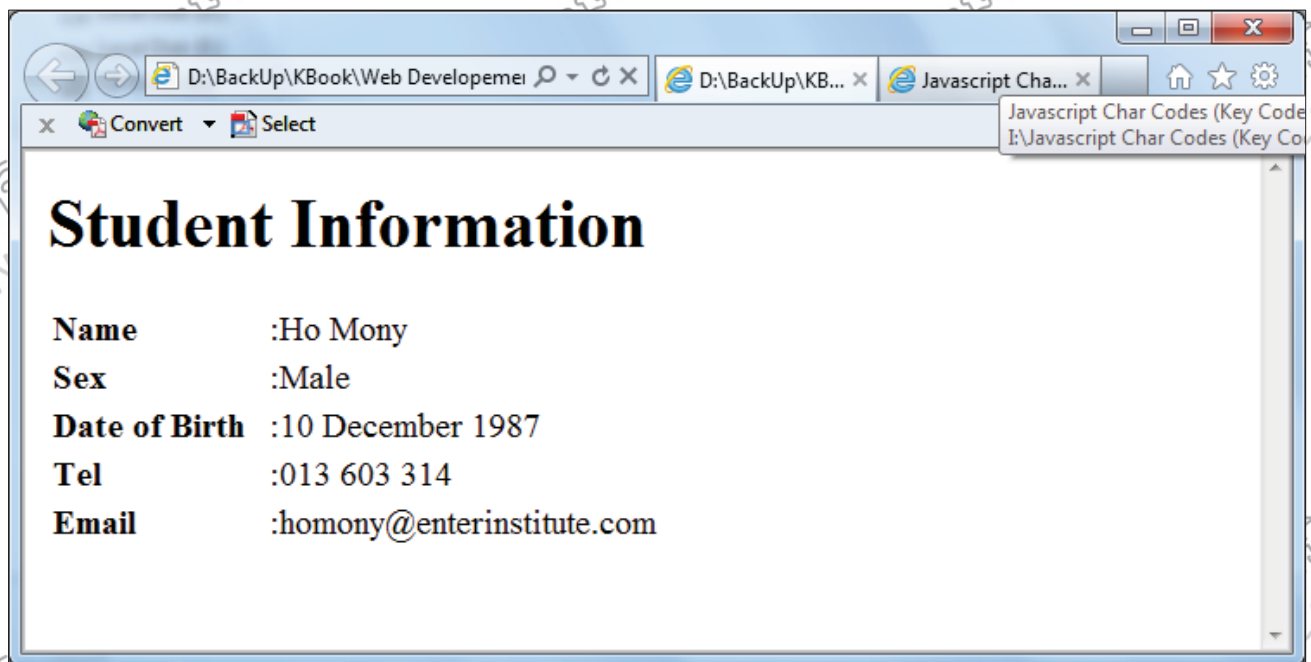
10. លំហាត់:

ចូរសរសេរ code មួយដើម្បី display ព័ត៌មានមួយចំនួនទៅក្នុង webpage ដូចខាងក្រោម:

- > ឈ្មោះរបស់អ្នក
- > ភេទរបស់អ្នក
- > ឆ្នាំកំណើត
- > លេខទូរស័ព្ទ
- > Email

Name	:Ho Mony
Sex	:Male
Date of Birth	:10 December 1987
Tel	:013 603 314
Email	:homony@enterinstitute.com

Output:



មេរៀនទី 3: ការប្រើប្រាស់ Variable និង Data Type ជាមួយ DOM

1. Variable:

នៅក្នុង JavaScript, Variables គឺត្រូវបានប្រើប្រាស់ដើម្បីរក្សាទុកទិន្នន័យសំរាប់យកទៅប្រើប្រាស់នៅក្នុង Source Code ។ ការបង្កើត Variable គឺយើងត្រូវប្រើប្រាស់ Keyword មួយគឺ var និងកំនត់ឈ្មោះរបស់វាផងដែរ។

```
var enter;
enter = "black";
```

ក្នុងការប្រើប្រាស់ Variable ២ក៏អាចប្រើប្រាស់ជាទំរង់កាត់ បានផងដែរ:

```
var enter = "black";
```

ចំពោះការដាក់ឈ្មោះ Variable គឺអាចសរសេរជាមួយនឹង តួអក្សរឬលេខ រួមទាំង និមិត្តសញ្ញា \$ និង Underscore() ។ ក្នុងនោះគឺមិនអនុញ្ញាតឱ្យប្រើប្រាស់ជាមួយនឹង Space ឬ និមិត្តសញ្ញាដទៃទៀតឡើយ។ ចំពោះ Variable ក្នុង JavaScript មានលក្ខណៈជា Case Sensitive មានន័យថា enter ខុសពី Enter ខុសពី ENTER ។

```
var chameleon = "blue";
var Chameleon = "red";
var CHAMELEON = "green";
var yellow_chameleon = "yellow";
var orangeChameleon = "orange";
var chameleon$ = "greedy";
```

Ex:

```
var enter = "black";
alert(enter);
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>02.Variable</p>
</body>
</html>
```

2. Scope Variable:

Scope Variable គឺជាការកំណត់ទីតាំងដែល Variable អាចប្រើប្រាស់បាននៅក្នុង Program ហើយវាត្រូវបានបែងចែក ជាពីរប្រភេទរួមមាន: Global Variable និង Local Variable ។

> Global Variable គឺជា Variable ដែលបានប្រកាសជាមួយ var Keyword នៅក្នុង body របស់ JavaScript ដោយវា មានវិសាលភាពអាចប្រើប្រាស់ទូទាំង Program ។

> Local Variable គឺជា Variable ដែលបានប្រកាសនៅក្នុង Function ឬនៅក្នុងសញ្ញា Braces { } ដែលមានអាចប្រើ ប្រាស់តែក្នុងតំបន់ប្រកាសវាតែប៉ុណ្ណោះ។

Ex:

```
function countWiis()
{
var stock = 5;
var sales = 3;
return stock - sales;
}
var stock = 0;
var wiis = countWiis();
alert(stock);
alert(wiis);
```

3. ការប្រើប្រាស់ DOM ជាមួយនឹង Variable:

DOM (Document Object Model) គឺជាវិធីសាស្ត្រក្នុងការ access ទិន្នន័យជាមួយនឹង Tag របស់ HTML ពោលគឺជាការសិក្សាពីការយក Data ចេញឬចូលជាមួយនឹង Tag របស់ HTML ដោយសន្មត់ថាគ្រប់ Tag ទាំងអស់គឺជា Object ។ ការដាក់ឈ្មោះរបស់ Object នីមួយៗ នោះគឺតាមរយៈ Attribute ថ្មីមួយឈ្មោះ id ហើយយើងអាចធ្វើការទាញយកតំលៃពី Object នីមួយៗមក store នៅក្នុង Variable បានផងដែរ។

Ex:

```
DemoDom.html - Notepad
File Edit Format View Help
<html>
<head>
<title> Demo Dom </title>
</head>
<body>
<h1 id="dis"></h1>
Name: <input type="text" id="m"/><br/>
<Script language="JavaScript">
    document.getElementById("dis").innerHTML = "welcome to HTML";
    document.getElementById("m").value="Ho Mony";
</script>
</body>
</html>
```

Output:



- > ចំពោះ innerHTML គឺត្រូវបានប្រើប្រាស់ដើម្បី access ទៅកាន់ Object ដែលប្រើសំរាប់បង្ហាញព័ត៌មាននៅលើ Webpage ដូចជា <h1> ឬ <p> តែប៉ុណ្ណោះ។
- > ចំពោះ value វិញគឺត្រូវបានប្រើប្រាស់ដើម្បី access ទៅកាន់ Object ដែលប្រើសំរាប់បង្ហាញព័ត៌មាននៅលើ Webpage ដូចជា <h1> ឬ <p> ផង នឹងអាច access ទៅកាន់ Object ដែល store តំលៃដូចជា TextBox បានផងដែរ។ ដូច្នេះភាគច្រើនក្នុងការ access ទៅកាន់ Object របស់ Form គឺតែងតែប្រើ value ជានិច្ច។

4. របៀបទាញទិន្នន័យពី Object តាមរយៈ DOM:

Ex:

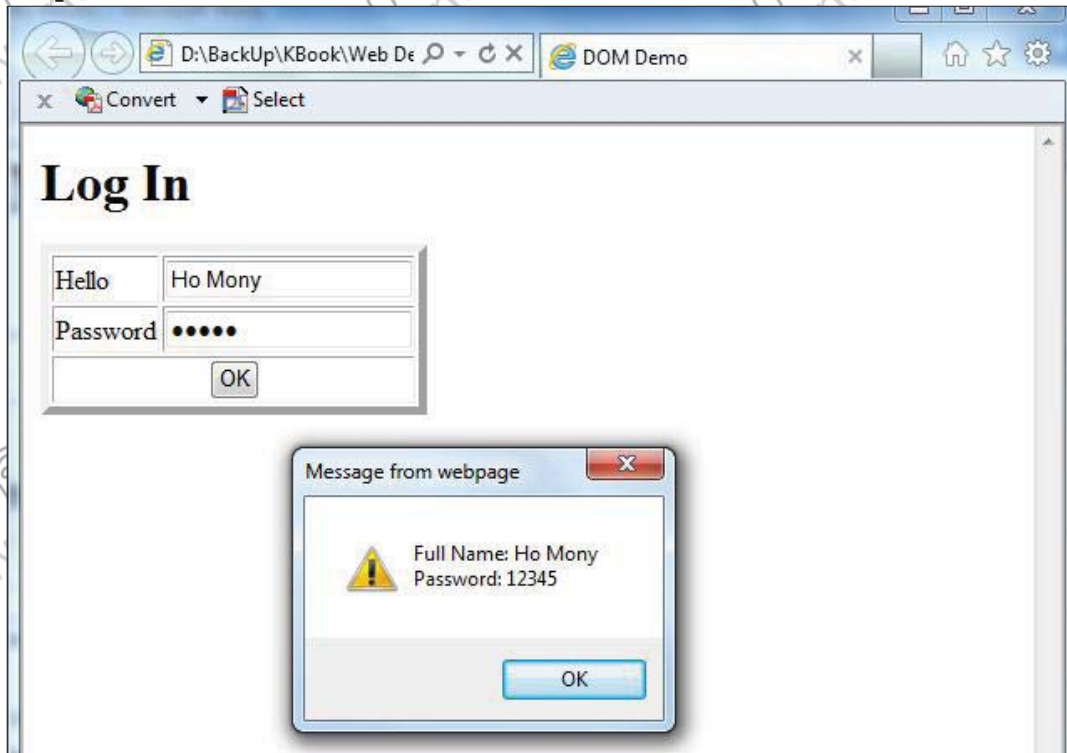
```

<html>
<head>
<title>DOM Demo</title>
<body>
<h1 >Log In</h1>
<table border="5">
<tr>
    <td>Full Name </td>
    <td><input type="text" id="fullname"/></td>
</tr>
<tr>
    <td>Password</td>
    <td><input type="password" id="password"/></td>
</tr>
<tr>
    <td colspan="2" align="center">
        <input type ="button" style="width:30px" value="OK"
onclick="FunctionOK() "/>
    </td>
</tr>
</table>

<script language="JavaScript">
function FunctionOK(){
    var name,psw;
    name=document.getElementById("fullname").value;
    psw=document.getElementById("password").value;
    alert("Full Name: "+name+"\n" +"Password: "+psw);
}
</script>
</body>
</html>

```

Output:



5. Input and Output Data with DOM:

Ex:

```

<html>
<head>
<title>Calclator</title>
<center>
<h1>Calclator</h1>
<table border="5">
<tr>
  <td>a </td>
  <td><input type="text" id="tfa"/></td>
</tr>
<tr>
  <td>b</td>
  <td><input type="text" id="tfb"/></td>
</tr>
<tr>
  <td colspan="2" align="center">
    <input type ="button" style="width:30px" value="+"
    onclick="calAdd()" />
  </td>
</tr>
<tr>
  <td>Result</td>
  <td><input type="text" id="tfr"/>
</td>
</tr>
</table>
</center>

```

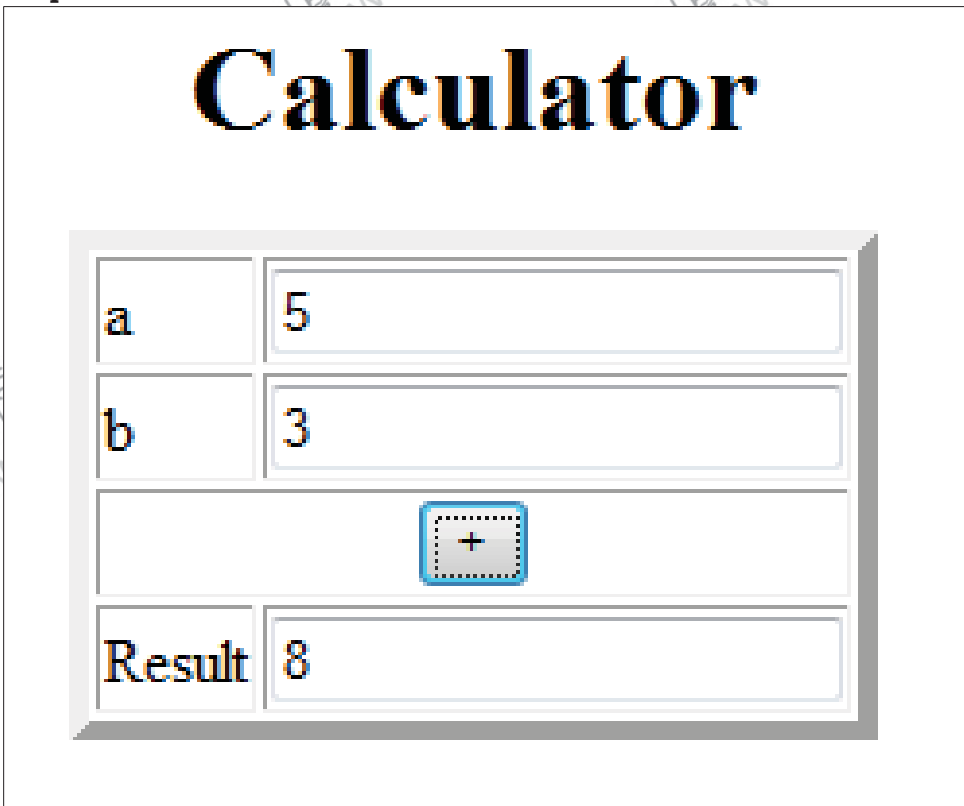
```

<script language="JavaScript">
  function calAdd(){
    var a,b,s;
    a=parseFloat(document.getElementById("tfa").value);
    b=parseFloat(document.getElementById("tfb").value);
    s=a+b;
    document.getElementById("tfr").value=s;
  }
</script>

</body>
</html>

```

Output:



6. Data Type:

នៅក្នុងភាសា Programming Language មួយចំនួនគឺយើងត្រូវធ្វើការដាក់ Variable ដោយមាន Data Type ត្រឹមត្រូវ ដើម្បីធ្វើការ Store ទិន្នន័យ ដែលលក្ខណៈនេះត្រូវបានហៅថា Strictly Typed ។ ប៉ុន្តែចំពោះ JavaScript វិញវាមានលក្ខណៈ Loosly Typed មានន័យយើងមិនចាំបាច់កំណត់ពី Data Type ទៅ៤ Variable នោះទេ ដោយ Variable នីមួយៗវាក៏នត់ Data Type ដោយខ្លួនឯងទៅតាមទិន្នន័យដែលវាទទួលយក។ Variable អាច Store ទិន្នន័យដូចជា:

- > Numbers
- > Strings
- > Booleans
- > Arrays
- > និង Objects ។

7. Number Data Type:

នៅពេលដែល Variable ទទួលយកទិន្នន័យដែលមានតំលៃជាលេខ នោះមានន័យថា data type របស់វាគឺមានប្រភេទលេខដែលនៅក្នុង JavaScript ត្រូវបានបែងចែកជាពីរគឺ:

- > int (Integer ជាលេខចំនួនគត់)
- > float (Floating Point Number ជាលេខមានក្បៀស) ។

```
var whole = 3;
var decimal = 3.14159265;
```

Ex:

```
var whole = 3;
var decimal = 3.14159265;
alert(whole + "\n" + decimal);
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>05.Number</p>
</body>
</html>
```

8. Mathematical (Arithmetic) Operation:

យើងអាចយកតំលៃ ឬ Variable ណាមួយទៅធ្វើការគណនាចូលគ្នា ដោយប្រើប្រាស់ជាមួយនឹង Mathematical Operator ដែលមានដូចជា + , - , * , / ជាដើម។

```
var addition = 4 + 6;
var subtraction = 6 - 4;
var multiplication = 5 * 9;
var division = 100 / 10;
var longEquation = 4 + 6 + 5 * 9 - 100 / 10;
```

Operator	Description	Example	Result of x	Result of y
+	Addition	x=y+2	7	5
-	Subtraction	x=y-2	3	5
*	Multiplication	x=y*2	10	5

/	Division	$x=y/2$	2.5	5
%	Modulus	$x=y\%2$	1	5
++	Increment	$x=++y$	6	6
		$x=y++$	5	6
--	Decrement	$x=--y$	4	4
		$x=y--$	5	4

ចំពោះការគណនាដែលមានសញ្ញា ជាច្រើនចូលគ្នានោះ:

- > សញ្ញា គុណ និង ចែក ជំនើរការមុន
- > សញ្ញា បូក និង ដក ជំនើរការក្រោយ

```
var longEquation = 4 + 6 + 5 * 9 - 100 / 10;
var longEquation = 4 + 6 + 45 - 100 / 10;
var longEquation = 4 + 6 + 45 - 10;
var longEquation = 10 + 45 - 10;
var longEquation = 55 - 10;
var longEquation = 45;
```

Ex:

```
var longEqu = 4+6+5*9-100/10;
alert(longEqu) ;
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>06 . Precedence</p>
</body>
</html>
```


ក្នុងនោះយើងអាចប្រើប្រាស់ Bracket () ដើម្បីកំណត់តំបន់ណាមួយអាចធ្វើការគណនាមុនគេ និងមុនសញ្ញា គុណ និងចែកបានផងដែរ។

```
var unbracketed = 4 + 6 * 5;
var bracketed = (4 + 6) * 5;
```

Ex:

```
var unbracketed = 4 + 6 * 5;
var bracketed = (4 + 6) * 5;
alert(unbracketed);
alert(bracketed);
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>06.Bracket</p>
</body>
</html>
```

9. Increment & Decrement:

នៅក្នុង JavaScript មាន operators ពិសេសចំនួនពីរទៀត សំរាប់ធ្វើការបន្ថែម 1 តំលៃ Variable (++) ឬ បន្ថយ 1 តំលៃ Variable (--)

> វិធីសាស្ត្រក្នុងបន្ថែម 1 តំលៃ Variable:

```
var age = 26;
age = age + 1;
```

```
var age = 26;
age += 1;
```

```
var age = 26;
age++;
```

> វិធីសាស្ត្រក្នុងបន្ថយ 1 តំលៃ Variable:

```
var age = 26;
age = age - 1;
```

```
var age = 26;
age -= 1;
```

```
var age = 26;
age--;
```

10. Prefix & Postfix:

Prefix មានន័យថាគឺជាការដាក់សញ្ញា Increment (++) ឬ Decrement (--) នៅផ្នែកខាងមុខ Variable ចំនែក Postfix មានន័យថាគឺជាការដាក់សញ្ញា Increment (++) ឬ Decrement (--) នៅផ្នែកខាងក្រោយ Variable ។ ចំពោះលក្ខណៈនេះគឺមានភាពខុសគ្នា នៅពេលដែលវាត្រូវ assign ទៅ Variable ផ្សេងៗ។

Operator	Description	Example	Result of x	Result of y
++	Increment	x=++y	6	6
		x=y++	5	6
--	Decrement	x=--y	4	4
		x=y--	5	4

Ex:

```
//Prefix
var i, j=5;
i = ++j;
alert("i =" + i + "\n"+"j =" + j);

//Postfix
var i, j=5;
i = j++;
alert("i =" + i + "\n"+"j =" + j);
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>08.Prefix and Postfix</p>
</body>
</html>
```

ចំពោះ សញ្ញា ++ ឬ -- អាចដាក់ខាងមុខ ឬខាងក្រោយ Variable បាន ប៉ុន្តែអាចទទួលបានលទ្ធផលខុសគ្នា:

> នៅពេលដែលសញ្ញា ++ នៅខាងមុខ j មានន័យថា j ត្រូវបន្ថែម 1 តំលៃចំនួនឯងហើយបញ្ជូនទៅ i តាមក្រោយ
ដូច្នោះ i = ++j មានន័យថា j កើន 1 តំលៃរួច ទើបបោះទៅ i

> នៅពេលដែលសញ្ញា ++ នៅខាងក្រោយ j វិញមានន័យថា j ត្រូវបន្ថែម 1 តំលៃចំនួនឯងក្រោយពេលបញ្ជូនទៅ i
ដូច្នោះ i = j++ មានន័យថា j បោះតំលៃចាស់ទៅ i សិន ទើបកើន 1 តំលៃចំនួនឯង ។

11. String:

String គឺជាបណ្តុំនៃកូអក្សរ ដែលអាចមានដូចជា letters, numbers, symbols, punctuation marks, ឬ spaces ។ វា
ទូទៅ String គឺត្រូវហ៊ុំព័ទ្ធដោយសញ្ញា quote marks ដែលអាចប្រើប្រាស់ជាមួយនឹង single quote ឬ double quote ។

Ex:

```
var single = 'Enter Information Technology Center';
alert(single);
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>09.String</p>
</body>
</html>
```

12. String Concaternation:

យើងអាចធ្វើការភ្ជាប់ String បន្តគ្នាបាន ដោយប្រើប្រាស់ជាមួយនឹងសញ្ញា បូក (+) ។

Ex:

```
var a = 'Enter';
var b = 'Center';
alert(a + "and" + b );
```

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>10.String Concatenation</p>
</body>
</html>
```

13. **Boolean:**

Boolean គឺជាប្រភេទ Data Type មួយដែល store ទិន្នន័យតែ 2 ប៉ុណ្ណោះ គឺ true និង false ។ ភាគច្រើនវាត្រូវបានប្រើប្រាស់ភ្ជាប់ជាមួយនឹងការដាក់លក្ខខណ្ឌដើម្បីកត់ត្រា True ឬ False សំរាប់ធ្វើការងារណាមួយ។

Ex:

```
var lying = true;
var truthful = false;
```

14. **Array:**

Array គឺជា បណ្តុំនៃធាតុ (Series of elements) ជាច្រើនដែលមាន Data Type ដូចគ្នាហើយដាក់បន្តគ្នានៅក្នុង memory location ។ ទីតាំងរបស់ធាតុនីមួយៗ ដែលស្ថិតនៅក្នុង memory location គឺមានលេខទីតាំង (index) មួយៗខុសៗគ្នាជានិច្ច។

អត្ថប្រយោជន៍របស់ Array គឺអាច Store តំលៃជាច្រើនដែលមានប្រភេទដូចគ្នានៅក្នុង array តែមួយ ដោយមិនចាំបាច់បង្កើត 5 Variables ផ្សេងៗគ្នាដើម្បី Store នោះទេ។

```
var rack = [];
rack[0] = "First";
rack[1] = "Second";
```



Ex:

```
//Declaring Array
var rack = [];
rack[0] = "First";
rack[1] = "Second";

//Showing Array
alert(rack[0]);
alert(rack[1]);
```

```

<html>
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
<body>
<p>11.Array</p>
</body>
</html>

```

យើងក៏អាចធ្វើការផ្តល់តំលៃនីមួយៗដោយផ្ទាល់ទៅក្នុង Index នីមួយៗរបស់ Array បាននៅពេលបង្កើត Array បានផងដែរ។

```

var rack = ["First", "Second", "Third", "Fourth"];

```



យើងក៏អាចធ្វើការបង្កើត Array ដែលមាន Data Type ខុសពី String បានផងដែរ ឬក៏អាចបង្កើត Array ដែលទទួលតំលៃជាប្រភេទខុសៗគ្នាបានផងដែរ។

```

var numberArray = [1, 2, 3, 5, 8, 13, 21, 34];
var stringArray = ["Veni", "Vidi", "Vici"];
var mixedArray = [235, "Parramatta", "Road"];

```

15. **Multi-Dimension Array:**

Multi-Dimension Arrays គឺជា array ដែលស្ថិតនៅក្នុង array ឬជា array ដែលមានច្រើនវិមាត្រជា array មួយដែលស្ថិតនៅក្នុង array មួយទៀត ប្រសិនបើវាត្រូវបាន Store នៅក្នុង memory location មានរាងដូចជា Table ។

```
var subArray1 = ["Paris", "Lyon", "Nice"];
var subArray2 = ["Amsterdam", "Eindhoven", "Utrecht"];
var subArray3 = ["Madrid", "Barcelona", "Seville"];
var superArray = [subArray1, subArray2, subArray3];

var city = superArray[0][2];
alert(city);

var town = superArray[2][1];
alert(town);
```

ក្នុងករណីដែលយើងត្រូវការបង្ហាញចំនួនធាតុដែល Array មាននោះ យើងត្រូវប្រើប្រាស់ length ដើម្បីបង្ហាញពីចំនួនធាតុពិតប្រាកដដែល Array មាន។

```
var shortArray = ["First", "Second", "Third"];
var total = shortArray.length;
alert(total);
```

16. **Associative Array:**

Associative Array គឺជាការផ្លាស់ប្តូរលេខ index របស់ Array ពីលេខ (0, 1, 2,...) ទៅជា String ឈ្មោះណាមួយ ។

```
var postcodes = [];
postcodes["Armadale"] = 3143;
postcodes["North Melbourne"] = 3051;
postcodes["Camperdown"] = 2050;
postcodes["Annandale"] = 2038;

alert(postcodes["Annandale"]);
```

17. **Objects:**

Objects គឺជាការរៀបចំ Variables និង Functions មួយចំនួនទៅជា Group មួយសំរាប់ប្រើប្រាស់ទៅតាមការងារជាក់លាក់របស់វា។ ប្រសិនបើយើងមានគោលបំណងធ្វើការងារជាមួយនឹង Robot គឺយើងអាចបង្កើត Objects ដែលមាន Variables មួយចំនួនពាក់ព័ន្ធនឹង Robot ដូចជា Age, Height, Weight, Metal,... និង Function មួយចំនួនដែលវាអាចធ្វើការជាក់ស្តែងនឹងស្ថានភាពពិតប្រាកដរបស់វាដូចជា Run(), Speak(), KillAllHumans(),... ។ យើងអាច Group Variables និង Functions នានា ផ្តុំបញ្ចូលគ្នាដើម្បីប្រើប្រាស់ជាមួយនឹង Robot តែប៉ុណ្ណោះ។

> Variables ដែលមាននៅក្នុង Objects ត្រូវបានហៅថា Properties

> Functions ដែលមាននៅក្នុង Objects ត្រូវបានហៅថា Methods

Ex:

```

var Robot = new Object();
Robot.metal = "Titanium";
Robot.killAllHumans = function()
{
  alert("Exterminate!");
};
Robot.killAllHumans ();

```

18. Object Literal Syntax:

Object Literal Syntax គឺអនុញ្ញាតឱ្យយើងធ្វើការបង្កើត Object ជាមួយនឹង Properties ឬ Methods បានក្នុងពេលតែមួយ។

Ex:

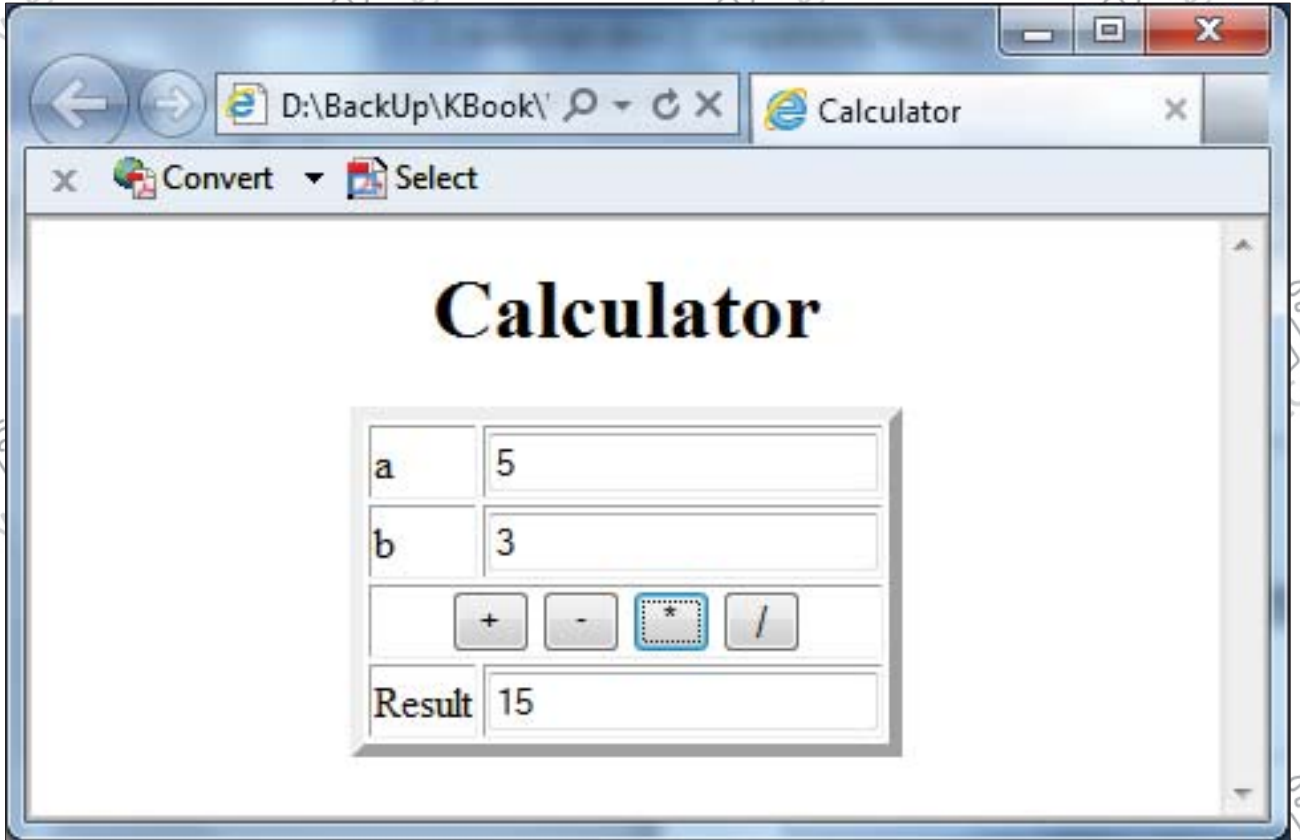
```

var Robot =
{
  metal: "Titanium",
  killAllHumans: function()
  {
    alert("Exterminate!");
  }
};

alert(Robot.metal);
Robot.killAllHumans ();

```

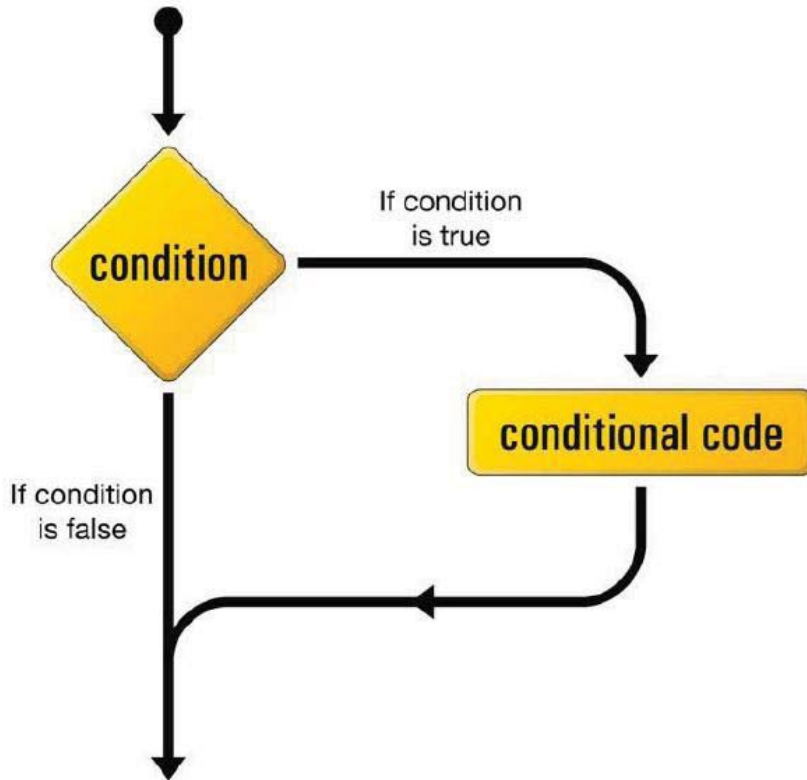
16. សំណាក:



មេរៀនទី 4: Conditions and Loops

1. ការប្រើប្រាស់ if:

If Keyword គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការប្រើប្រាស់ execute នូវ block នៃ code នៅពេលដែលលក្ខខណ្ឌរបស់វាពិត។



Syntax:

```

if (condition)
{
    conditional code;
}
  
```

> if គឺជា Keyword សំរាប់ប្រើប្រាស់ដើម្បីដាក់លក្ខខណ្ឌនៅក្នុង Source Code

> condition គឺជាលក្ខខណ្ឌដើម្បីស្វែងរកលទ្ធផល True

> statement គឺជាបណ្តុំនៃ Code ដែលត្រូវធ្វើការនៅពេលដែល condition ទទួលបានតំលៃ True ប៉ុន្តែបណ្តុំនៃ Code នឹងត្រូវបានរំលងចោល ប្រសិនបើ condition នៅក្នុង if ផ្តល់លទ្ធផល False វិញ។

Ex1:

```

var age = 27;
if (age > 20)
{
  alert("Drink to get drunk");
}

```

Ex2:

```

<html>
<body>
<script>
var r=confirm("Press a button");
if (r==true)
{
  x="You pressed OK!";
  alert(x);
}
else
{
  x="You pressed Cancel!";
  alert(x);
}
</script>
</body>
</html>

```

2. Comparison Operator:

Comparison Operator គឺត្រូវបានប្រើប្រាស់សំរាប់ធ្វើការប្រៀបធៀបរវាង តំលៃ Variable ឬ Expression ពីរ ជាមួយគ្នាដើម្បីធ្វើការស្វែងរកលទ្ធផល ពិត ឬមិនពិត។ ជាទូទៅវាត្រូវបានប្រើប្រាស់ជាមួយនឹង Condition របស់ if Statement ។

1 Comparison Operators ដែលត្រូវបានប្រើប្រាស់មានសញ្ញាមួយចំនួនដូចជា:

Comparison Operator	អត្ថន័យ	ឧទាហរណ៍: x=5,y=9	លទ្ធផល
<	តូចជាង	x < y	True
<=	តូចជាងឬស្មើ	x <= y	True
>	ធំជាង	x > y	False
>=	ធំជាងឬស្មើ	x >= y	False
==	ស្មើគ្នា	x == y	False
!=	មិនស្មើគ្នា (ខុសពី)	x != y	True

Ex1:

```

var name = "Maximus";
if (name == "Maximus")
{
  alert("Good afternoon, General.");
}

```

Ex2:

```

var name = "Decimus";
if (name != "Maximus")
{
  alert("You are not allowed in.");
}

```

3. **Multiple Conditions:**

Logical Operator គឺត្រូវបានប្រើប្រាស់ដើម្បីភ្ជាប់ជាមួយនឹង Comparison Operator ឬប្រើក្នុងលក្ខខណ្ឌដទៃទៀត ដើម្បីធ្វើការ ស្វែងរកលទ្ធផល ពិត ឬមិនពិត។

ក្នុងនោះ សញ្ញាដែលត្រូវប្រើមានបីប្រភេទដូចជា: && (and), || (or), ! (not)។

Ex1:

```
var age = 27;
if (age > 17 && age < 21)
{
    alert("Old enough to vote, too young to drink");
}
```

Ex2:

```
var sport = "Skydiving";
if (sport == "Bungee jumping" || sport == "Cliff diving" ||
sport == "Skydiving")
{
    alert("You're extreme!");
}
```

4. **Else if & Else Statement:**

else if គឺត្រូវបានប្រើប្រាស់សំរាប់ដាក់លក្ខខណ្ឌ បន្ទាប់ពី if មានន័យថាប្រសិនបើ លក្ខខណ្ឌមិនបាន True នៅក្នុង if ទេនោះវានឹងបន្តមកត្រួតពិនិត្យលក្ខខណ្ឌនៅក្នុង else if ជាបន្តទៀត ហើយប្រសិនបើលក្ខខណ្ឌពិតនោះវានឹងដំនើ គ្រាន់រូប Block នៃ code ដែលបានកំណត់។

ជាមួយនឹង else វិញគឺត្រូវបានប្រើប្រាស់នៅក្រោម if ឬ else if ដើម្បីធ្វើការកំណត់លក្ខខណ្ឌពិត នៅពេលដែលលក្ខ ខណ្ឌត្រូវបាន False នៅក្នុង if ឬ else if ។

Ex:

```
var name = "Marcus";

if (name == "Maximus")
{
    alert("Good afternoon, General.");
}

else if (name == "Marcus")
{
    alert("Good afternoon, Emperor.");
}
else
{
    alert("You are not allowed in.");
}
```

5. ការប្រើប្រាស់ Conditional Operator:

Conditional Operator គឺជា Operator ដែលត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការដាក់លក្ខខណ្ឌ ដូចជាទៅនឹង If ផងដែរ។

Syntax:

variableName = (condition)?valueTrue:valueFalse

Ex:

```
voteable=(age<18)?"Too young":"Old enough";
```

6. ការប្រើប្រាស់ Switch:

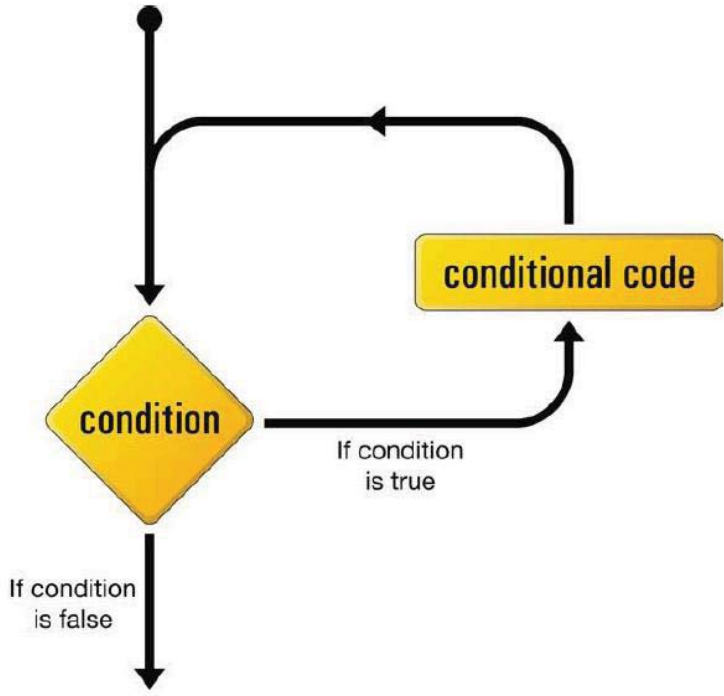
Switch Keyword គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការប្រើ execute នូវ block នៃ code នៅពេលដែលលក្ខខណ្ឌរបស់វាពិត ដូចគ្នាទៅនឹង if ដែរ។ វាមានភាពងាយស្រួលជាង if នៅពេលធ្វើការត្រួតពិនិត្យលើលក្ខខណ្ឌ ដែលមានច្រើនករណី ប៉ុន្តែវាមានភាពពិបាកជាង if នៅពេលលក្ខខណ្ឌត្រូវបានភ្ជាប់ជាមួយនឹង and, or, ឬ not ។

Ex:

```
<html>
<body>
<script>
var day=new Date().getDay();
switch (day)
{
case 0:
x="Today it's Sunday";
break;
case 1:
x="Today it's Monday";
break;
case 2:
x="Today it's Tuesday";
break;
case 3:
x="Today it's Wednesday";
break;
case 4:
x="Today it's Thursday";
break;
case 5:
x="Today it's Friday";
break;
case 6:
x="Today it's Saturday";
break;
default:
x="Looking forward to the Weekend";
}
alert(x);
</script></body></html>
```

7. While Loop:

while គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការដំនើការនូវ Block នៃ Code ដដែលៗ នៅពេលដែលលក្ខខណ្ឌរបស់ True ។ មុននឹងវាដំនើការទៅលើ Block នៃ Code គឺវាតែកមើលលក្ខខណ្ឌជាមុនសិន ប្រសិនបើ True ធ្វើ ប៉ុន្តែបើ False គឺមិនធ្វើ សូម្បីតែមួយដងក៏ដោយ។



```

while (condition)
{
    conditional code;
}
  
```

Ex:

```

var numbers = [1, 2, 3, 4, 5];
var incremter = 0;
while (incremter < numbers.length)
{
    numbers[incremter] *= 2;
    alert(numbers[incremter]);
    incremter++;
}
  
```

8. do while:

do while គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការដំនើការនូវ Block នៃ Code ដដែលៗ នៅពេលដែលលក្ខខណ្ឌរបស់ True ។ មុននឹងតែកមើលលក្ខខណ្ឌ វាដំនើការទៅលើ Block នៃ Code ម្តងជាមុនសិន ប្រសិនបើ True ធ្វើឡើងវិញប៉ុន្តែបើ False គឺនឹងចាកចេញពី Loop ។

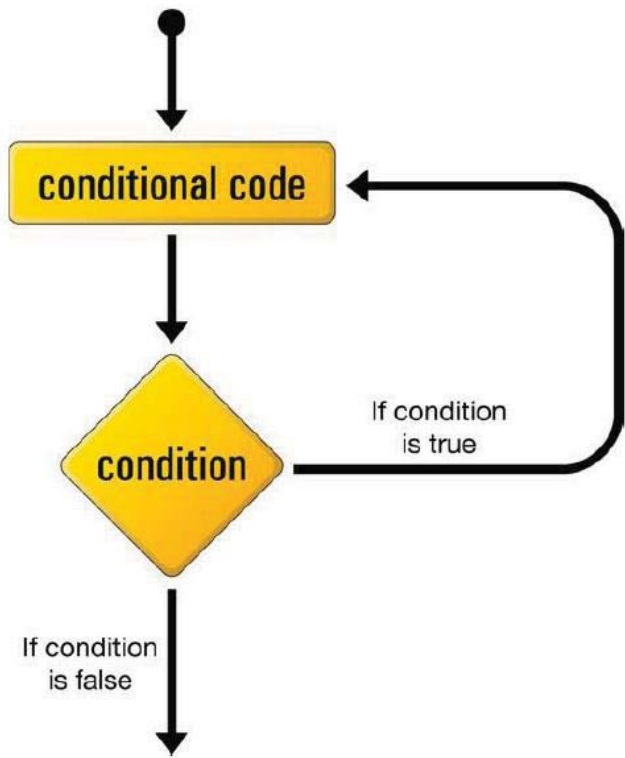
```
do
{
    conditional code;
} while (condition);
```

Ex:

```
var x =1;
do{
    alert(x);
    x+=1;
}while (x<=5);
```

9. **for loop:**

for គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការដំនើរការនូវ Block នៃ Code ដដែលៗ ទៅតាមចំនួនដែលបានកំណត់យ៉ាងត្រឹមត្រូវ នៅពេលដែលលក្ខខណ្ឌរបស់ True ។ ចំពោះ while និង do while គឺធ្វើគិតទៅលើ លក្ខខណ្ឌ ដោយមិនគិតពីចំនួន ដងឡើយ ប៉ុន្តែ for វិញគឺធ្វើគិតទៅលើចំនួនដង ពិតប្រាកដ។ ជាទូទៅចំពោះការ loop គេនិយមប្រើប្រាស់ for ពី ព្រោះយើងអាចដឹងពីចំនួនដែលវាត្រូវធ្វើការនៅក្នុង loop ។



Syntax:

```
for( 1 initialization; 2 condition; 4 increase) {
    3 statement;
}
```

1. Initialization: គឺជាតំលៃចាប់ផ្តើមដំនើរការ loop ហើយវាធ្វើការតែម្តងប៉ុណ្ណោះ។
2. Condition: គឺជាលក្ខខណ្ឌដែលត្រូវត្រួតពិនិត្យ ប្រសិនបើ True loop នឹងបន្តដំនើរការ ប៉ុន្តែប្រសិនបើ False វិញនោះ Loop នឹងបញ្ចប់ដំនើរការ
3. statement: គឺជា Block នៃ code ដែលត្រូវដំនើរការក្នុង braces { } នៅពេលដែល លក្ខខណ្ឌ True
4. increase: គឺជាការតំឡើងឬបន្ថយ value របស់ variable នៅក្នុង initialization ឲ្យកើនឡើង ឬថយចុះ ហើយបន្ទាប់មកវានឹងត្រលប់ទៅដំនើរការនៅក្នុងតំបន់ Condition វិញ។

Ex:

```
var numbers = [1, 2, 3, 4, 5];
for (var i = 0; i < numbers.length; i++)
{
    numbers[i] *= 2;
    alert(numbers[i]);
}
```

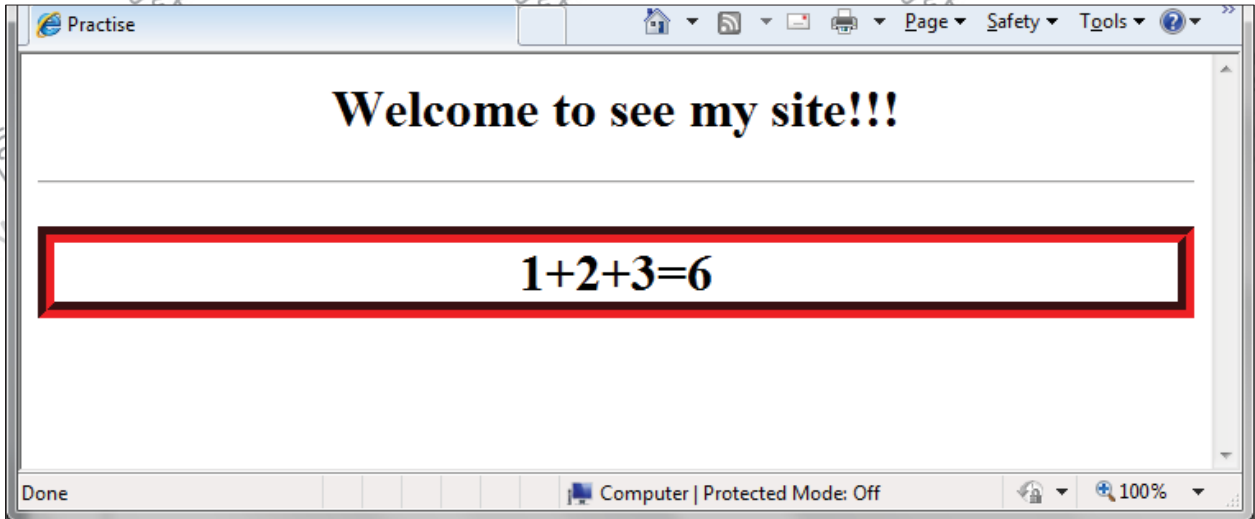
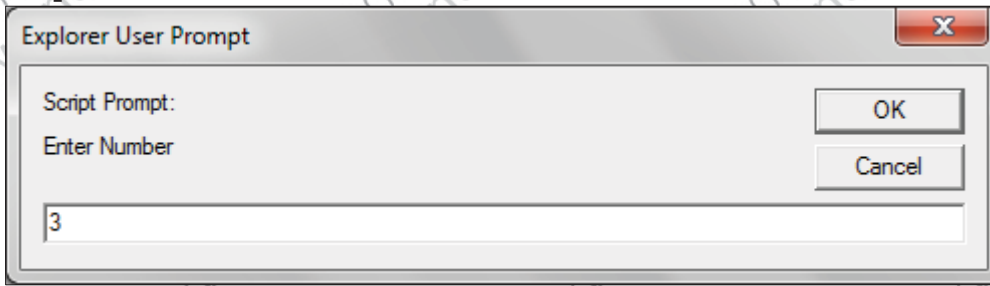
10. ឧទាហរណ៍នៃការប្រើប្រាស់ For & If:

Ex:

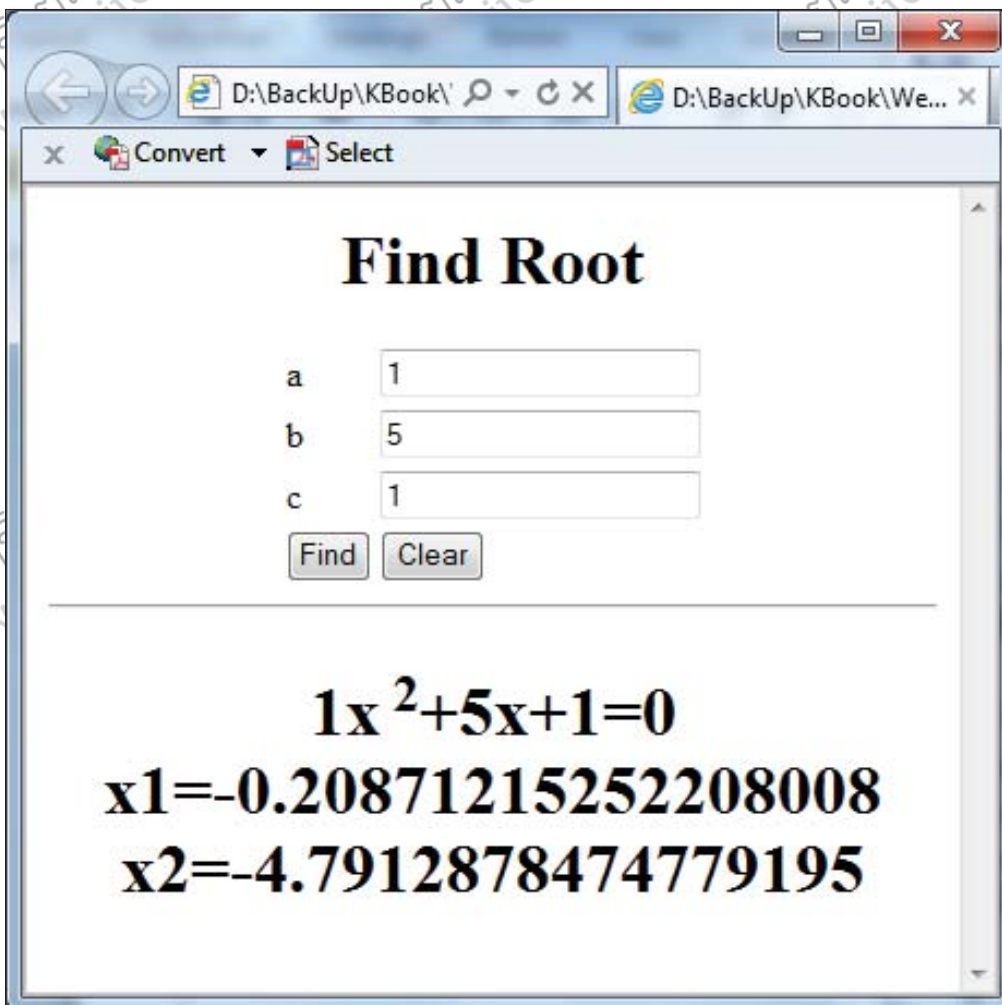
```
<html>
<head>
<title>Practise</title>
<script language="JavaScript">
var n,s,r;
r = "";
s = 0;
n = parseInt(prompt("Enter Number","0"));
for(var i=1;i<=n;i++){
    if(i<n)    r+=i +"";

    else r+= i + "=";
    s+=i;
}
</script>
</head>
<body>
<center>
<h1> Welcome to see my site!!!</h1>
<hr/>
<script language = "JavaScript">
document.write("<h1 style='border : 10 groove red'>" + r + s +
"</h1>");
</script>
</body>
</html>
```

Output:



11. សំណាត់:



មេរៀនទី ៥: ការបង្កើត Function

1. និយមន័យ:

Function គឺជាបណ្តុំនៃ Code ដែលត្រូវធ្វើការងារជាក់លាក់ណាមួយទៅតាមការ Call ហៅរបស់ Object ឬ Element ណាមួយ។ វាត្រូវបានបង្កើតឡើងដើម្បីកាត់បន្ថយនូវការសរសេរកូដ ដដែលៗច្រើនសារ។

Syntax:

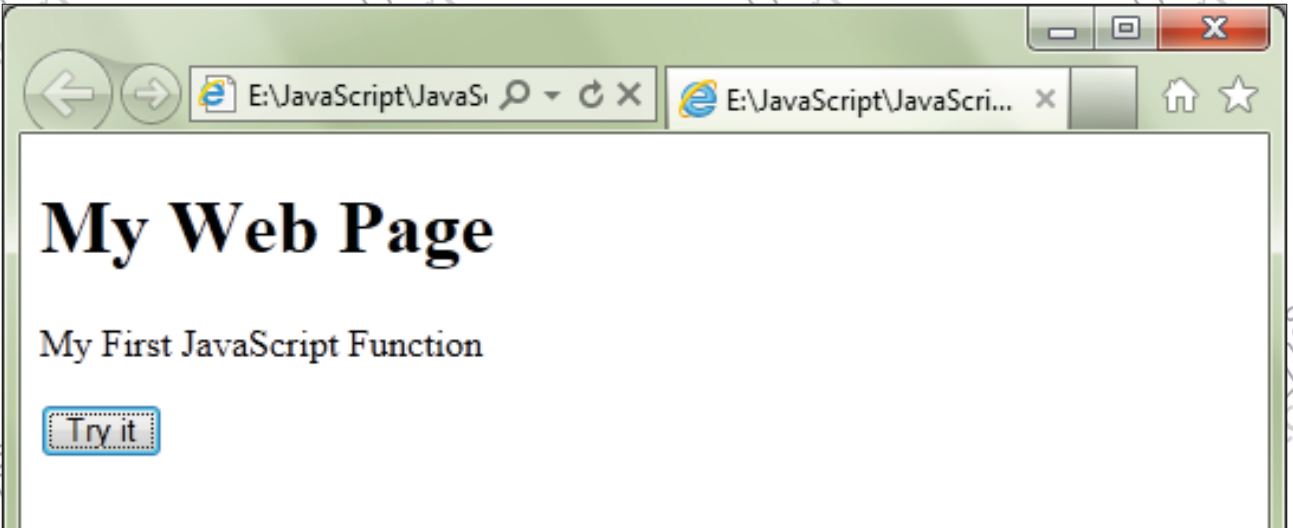
```
function FunctionName()
{
    Executable Code;
}
```

> ចំពោះ function keyword គឺត្រូវសរសេរជាមួយនឹងអក្សរដំបូងតូចជានិច្ច

Ex:

```
<!DOCTYPE html>
<html>
<body>
<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try
it</button>
<script type="text/javascript">
function myFunction()
{
document.getElementById("demo").innerHTML="My First
JavaScript Function";
}
</script>
</body>
</html>
```

Output:



2. Call Fuction ហ្វុន្យុង Arguments:

យើងអាចធ្វើការបោះតំលៃ ឬ Variable មួយឬច្រើនទៅ function យកទៅធ្វើការ ហើយ return តំលៃបានមកវិញផងដែរ។ ក្នុងនោះតំលៃដែលបោះទៅ function ត្រូវបានហៅថា Arguments ដែលក្នុង Function នីមួយៗ Argument អាចមានតែមួយ ឬច្រើនទៅតាមការងារជាក់ស្តែងផងដែរ។ Arguments ជាច្រើនក្នុង Function ត្រូវបានបំបែកដោយប្រើប្រាស់ Comma (,) ។

Syntax:

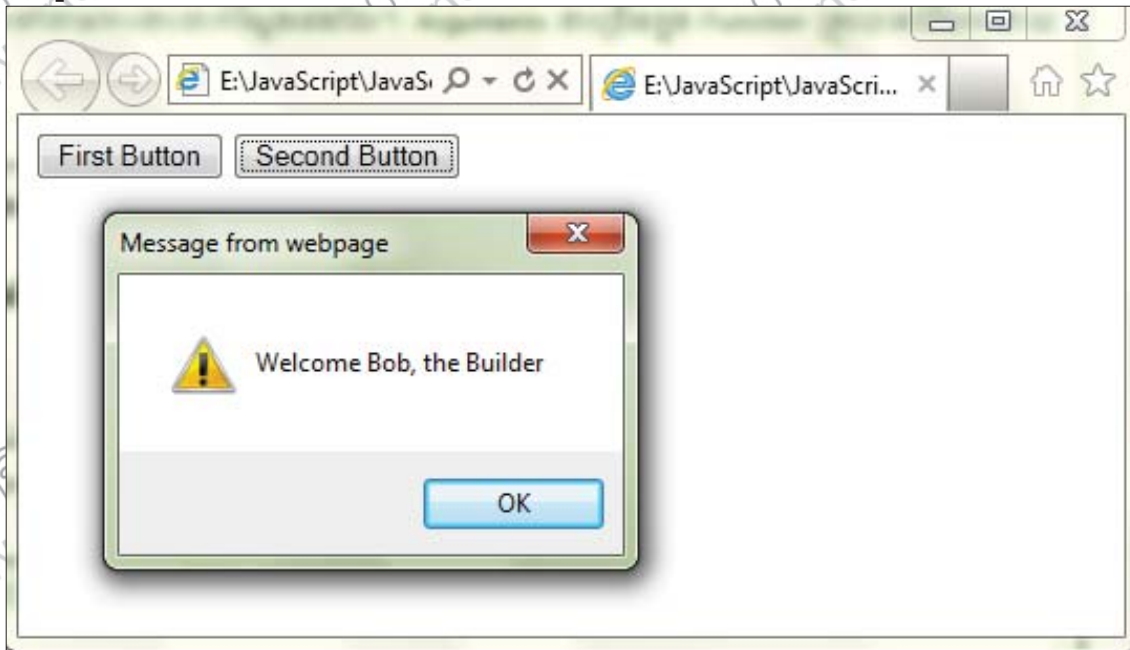
```
function FunctionName(Arg1,Arg2,...)
{
    Executable Code;
}
```

> ចំពោះ function keyword គឺត្រូវសរសេរជាមួយនឹងអក្សរដំបូងតូចជានិច្ច

Ex:

```
<!DOCTYPE html>
<html>
<body>
<button onclick="myFunction('Harry Potter','Wizard')">First Button</button>
<button onclick="myFunction('Bob','Builder')">Second Button</button>
<script type="text/javascript">
function myFunction(name,job)
{
alert("Welcome " + name + ", the " + job);
}
</script>
</body>
</html>
```

Output:



3. Function with Return Value:

Return Keyword ត្រូវបានប្រើប្រាស់នៅក្នុង Function ដើម្បី return (បោះតំលៃ) ទៅកាន់ Object ឬ Element ណាមួយ ដែលបាន Call ហៅវាយកមកប្រើ។

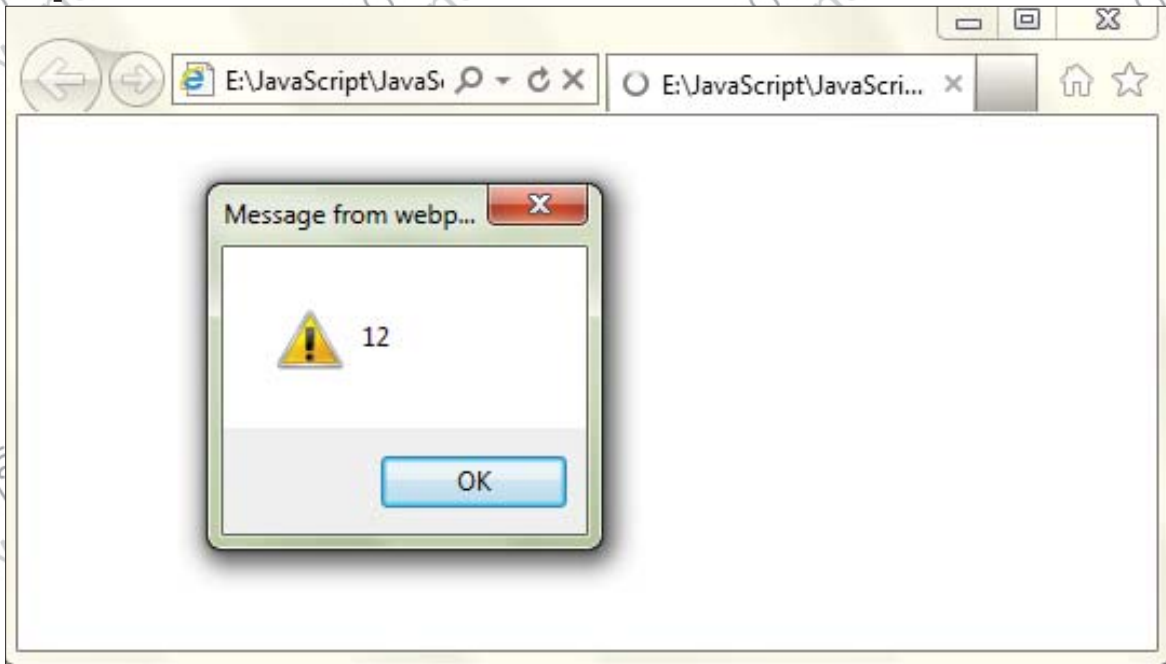
Ex:

```

<!DOCTYPE html>
<html>
<body>
<script>
function myFunction(a,b)
{
return a*b;
}
var x = myFunction(4,3);
alert(x);
</script>
</body>
</html>

```

Output:



4. ការបង្កើត Function Input & Output:

Ex:

> Mystring.js

```
function out(m) {
    document.write(m);
}

function getString(m) {
    return prompt(m, "");
}

function getInt(m) {
    return parseInt(getString(m));
}

function getFloat(m) {
    return parseFloat(getString(m));
}
```

> CallFunction.html

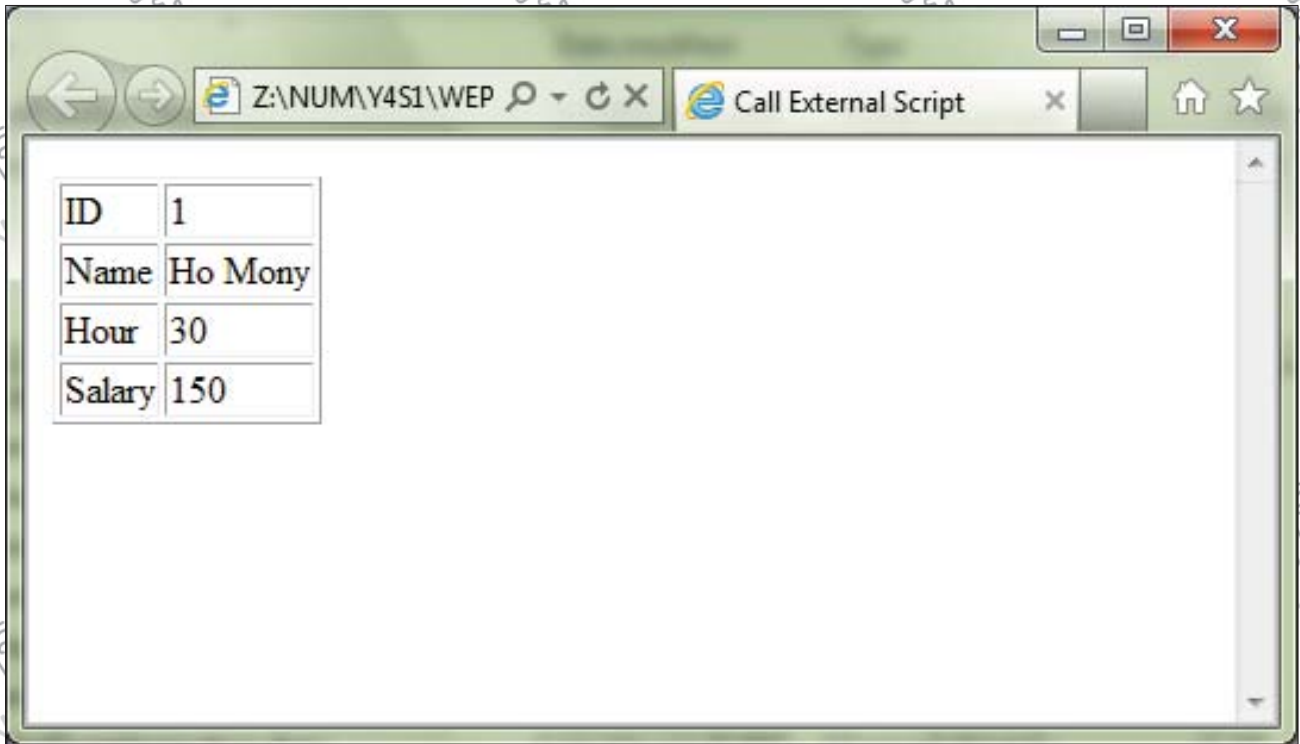
```

<html>
<head>
<title>Call External Script</title>
<script language="JavaScript" src="mystring.js">
</script>

<script language ="JavaScript">
var id, name, hour, rate, salary;
id= getString("ID:");
name= getString("Name:");
hour= getInt("Hour:");
rate= getFloat("Rate:");
salary= hour* rate;
</script>
</head>

```

Output:



មេរៀនទី 6: សិក្សាពី Array

1. និយមន័យ:

Array គឺជា Variable ពិសេសមួយ ដែលអាចផ្ទុកតំលៃបានច្រើន ដែលមាន Data Type ដូចគ្នា។ វាត្រូវបានបង្កើតឡើងដើម្បីកាត់បន្ថយនូវការបង្កើត variable ជាច្រើនដែលត្រូវផ្ទុកទិន្នន័យដែលមាន Data Type ដូចគ្នា។ ក្នុងការរក្សាទុកតំលៃនីមួយៗ នៅក្នុង Array គឺមាន index ខុសៗគ្នា ដើម្បីសំគាល់ពី តំលៃទាំងអស់នោះ។

Ex:

```
//Use Variable
var car1="Saab";
var car2="Volvo";
var car3="BMW";
```

```
//Use Array
var myCars=new Array();
myCars[0]="Saab";
myCars[1]="Volvo";
myCars[2]="BMW";
```

2. ការបង្កើត Array:

ក្នុងការបង្កើត Array គឺអាចបង្កើតតាមរយៈវិធីសាស្ត្រ 3 ប្រភេទផ្សេងៗគ្នា ដែលក្នុងនោះមានដូចជា:

> របៀបទី 1:

```
var myCars=new Array();
myCars[0]="Saab";
myCars[1]="Volvo";
myCars[2]="BMW";
```

> របៀបទី 2:

```
var myCars=new Array("Saab","Volvo","BMW");
```

> របៀបទី 3:

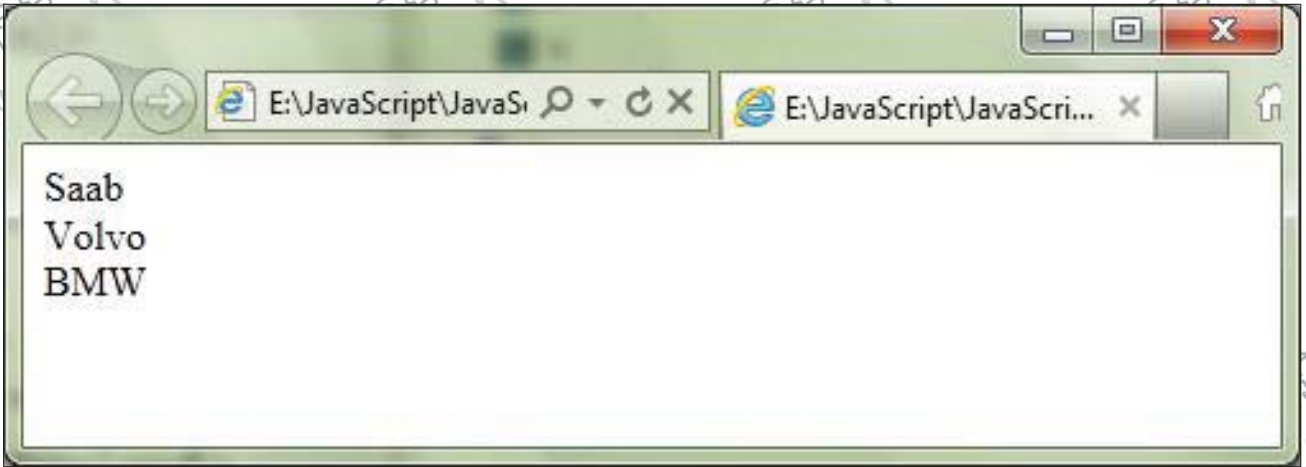
```
var myCars=["Saab","Volvo","BMW"];
```

3. Output Array:

Ex:

```
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
var i;
var mycars = new Array();
mycars[0] = "Saab";
mycars[1] = "Volvo";
mycars[2] = "BMW";
for (i=0;i<mycars.length;i++)
{
document.write(mycars[i] + "<br />");
}
</script>
</body>
</html>
```

Output:



4. Input & Output Array:

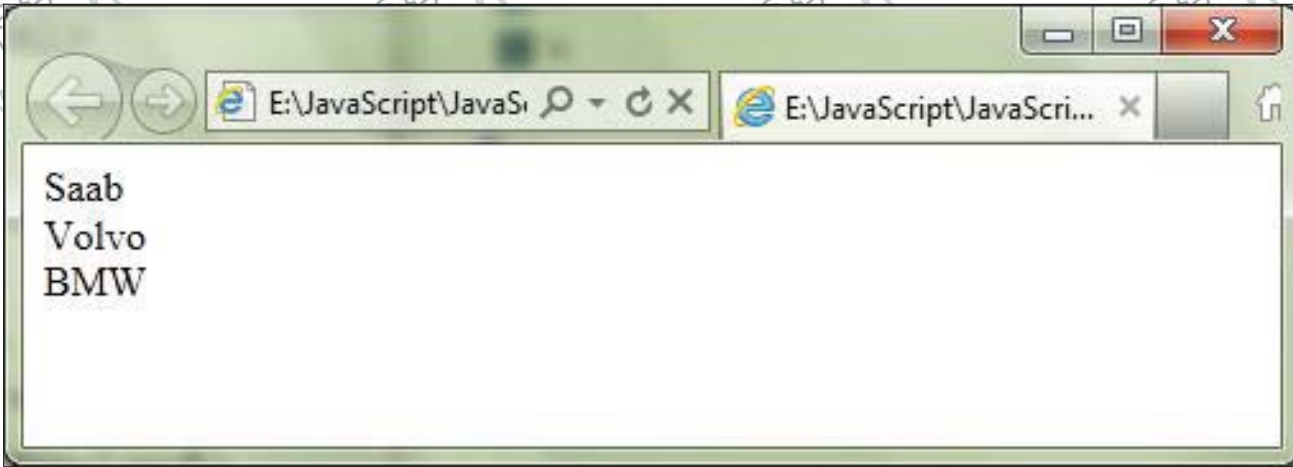
Ex:

```

<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
var i;
var mycars = new Array();
for (i=0;i<5;i++)
{
mycars[i] = prompt("Enter Array:", "");
}
for (i=0;i<mycars.length;i++)
{
document.write(mycars[i] + "<br />");
}
</script>
</body>
</html>

```

Output:



5. Sort Array Ascending:

Ex:

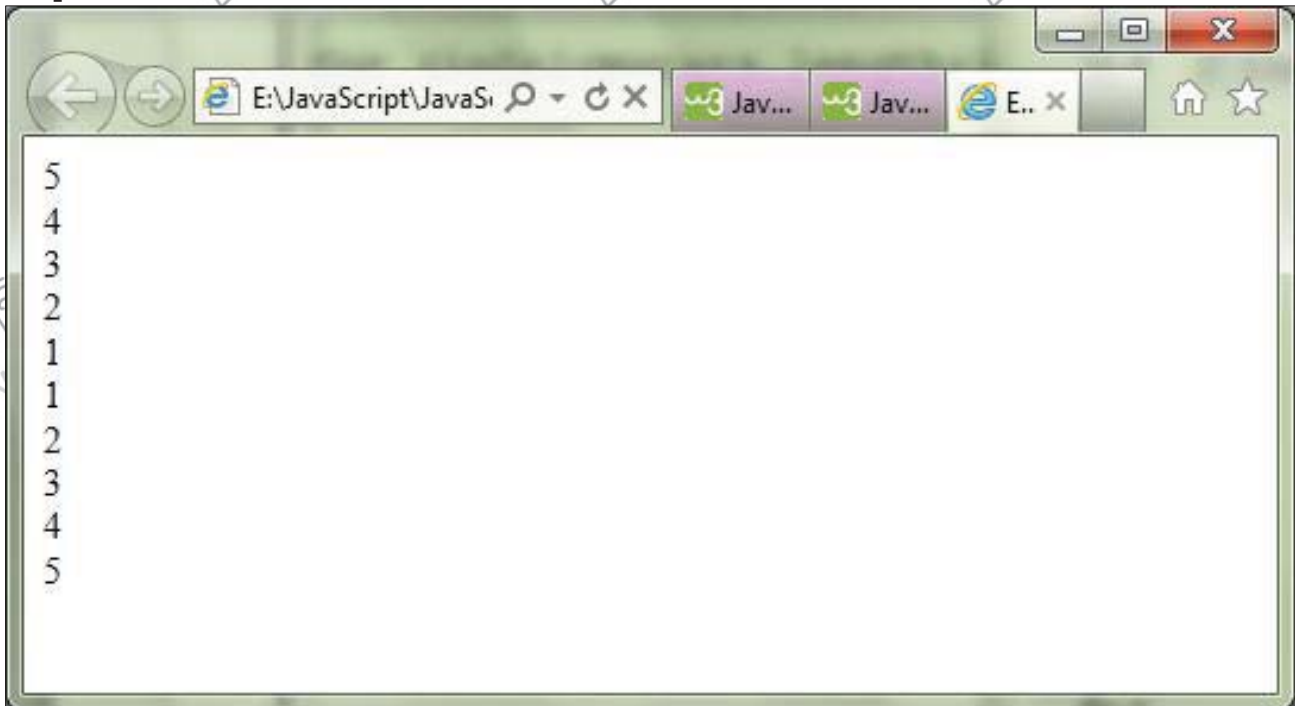
```

<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
var i;
var mycars = new Array();
for (i=0;i<5;i++)
{
mycars[i] = prompt("Enter Array:", "");
}
for (i=0;i<mycars.length;i++)
{
document.write(mycars[i] + "<br />");
}
mycars = mycars.sort();

for (i=0;i<mycars.length;i++)
{
document.write(mycars[i] + "<br />");
}
</script>
</body>
</html>

```

Output:



6. Sort Array Descending:

Ex:

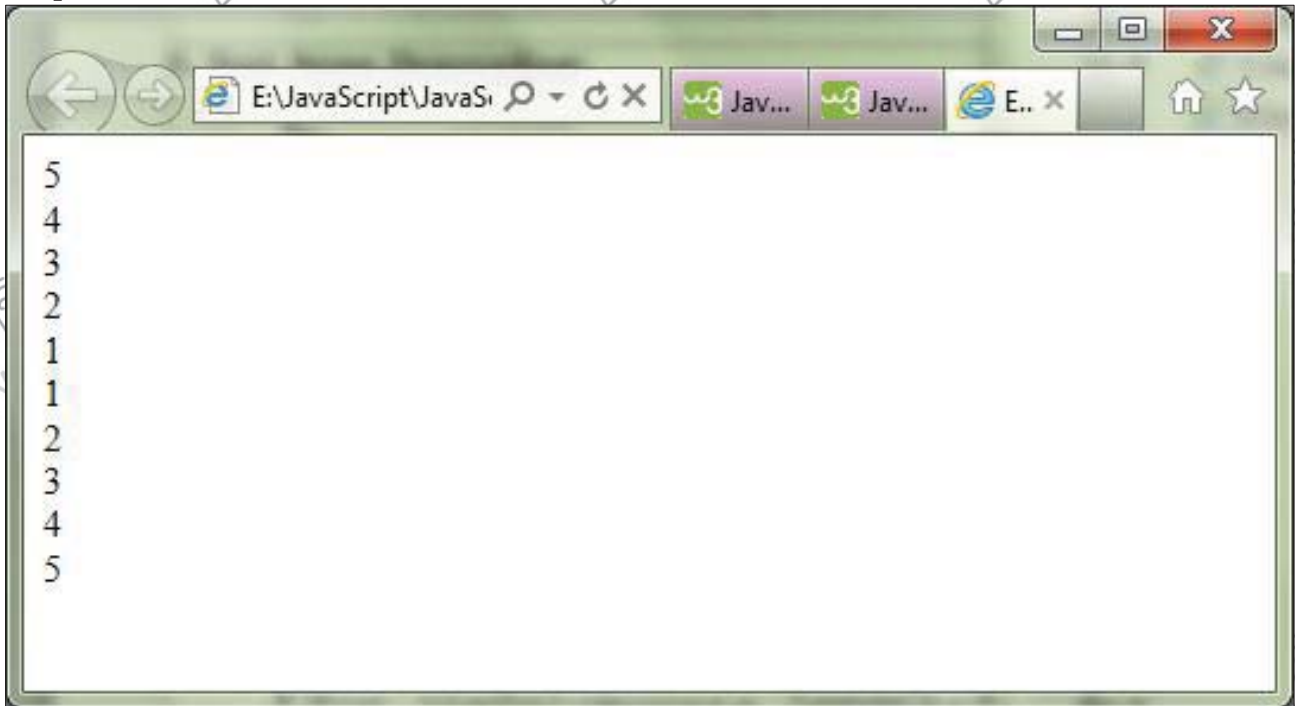
```

<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
var i;
var mycars = new Array();
for (i=0;i<5;i++)
{
mycars[i] = prompt("Enter Array:", "");
}
for (i=0;i<mycars.length;i++)
{
document.write(mycars[i] + "<br />");
}
mycars = mycars.sort();

for (i=0;i<mycars.length;i++)
{
document.write(mycars[i] + "<br />");
}
</script>
</body>
</html>

```

Output:



7. ការបង្កើត Table និង Array:

Ex:

> mystring.js

```
function out(m) {
    document.write(m);
}
function getString(m) {
    return prompt(m, "");
}
function getInt(m) {
    return parseInt(getString(m));
}
function getFloat(m) {
    return parseFloat(getString(m));
}
function getRandom(min,max) {
    return parseInt(Math.random() * (max-min)+1);
}
function createArray(r,c,min,max) {
    var x = new Array(r);
    for(var i=0;i<r;i++)
        x[i]=new Array(c);
    for(var i=0;i<r;i++)
        for(var j=0;j<c;j++)
            x[i][j]=getRandom(min,max);
    return x;
}
function showTable(x,r,c) {
    out("<table border=1>");
    for(var i=0;i<r;i++) {
        out("<tr>");
        for(var j=0;j<c;j++)
            out("<td>" + x[i][j]);
        out("</tr>");
    }
    out("</table>");
}
```

> mystyle.css

```
#mode1 { color:red; text-align:center}

#mode2 { border:10 groove red;

        background-image:url('1.jpg');

        text-align:center

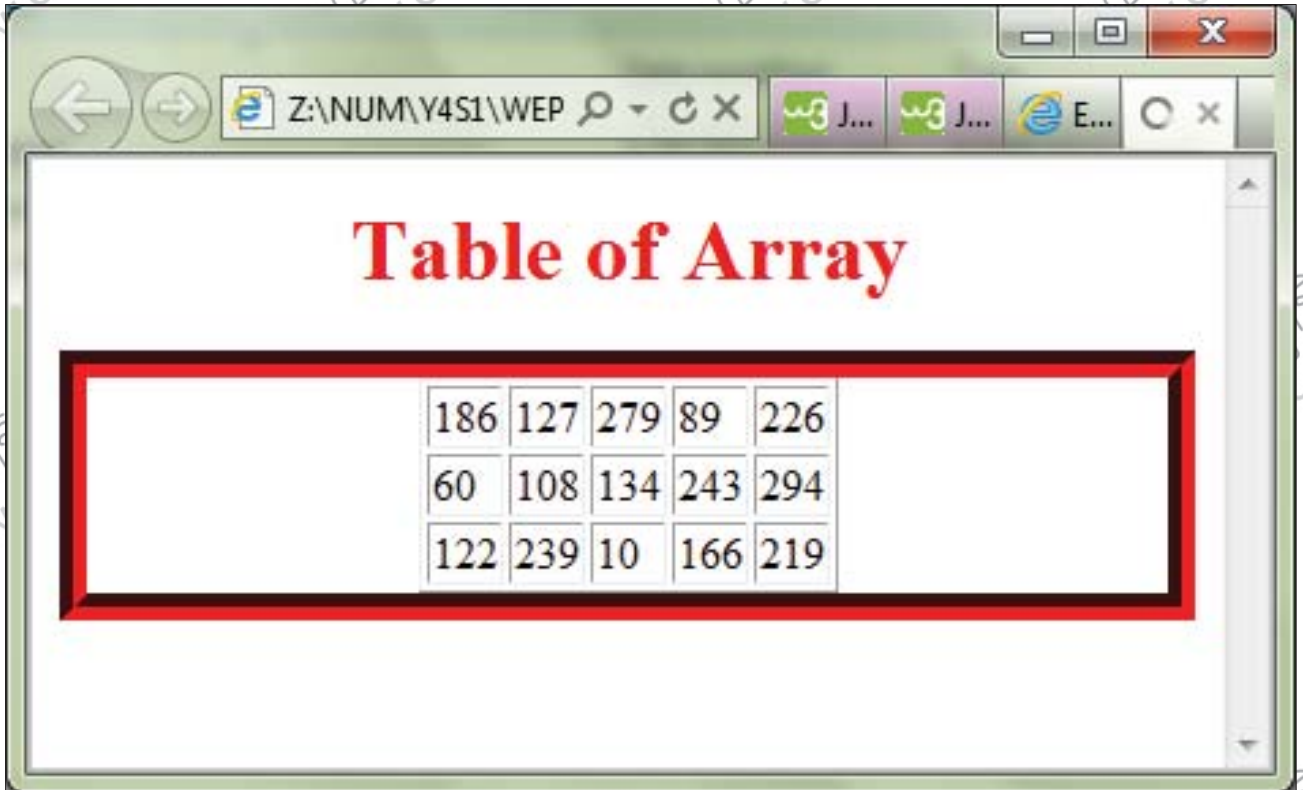
        }
```

> default.html

```
<html>
<head>
<title>Call External Script</title>
<link rel="stylesheet" type="text/css"
href="mystyle.css" />
<script language="JavaScript" src="mystring.js">
</script>
</head>

<body>
<h1 id="mode1"> Table of Array</h1>
<div id="mode2">
<script language="javaScript">
var r,c,x;
r=getInt("Enter row:");
c=getInt("Enter Col:");
x=createArray(r,c,1,300);
showTable(x,r,c);
</script>
</div>
</body>
</html>
```

Output



មេរៀនទី 7: ការប្រើប្រាស់ Event

1. និយមន័យ:

Event គឺជាសកម្មភាព ឬការងារណាមួយដែលជំនើការ នៅពេលវេលាណាមួយដែលបានកំណត់ដោយ JavaScript ។ ឧទាហរណ៍យើងអាចធ្វើការចុចលើ Button ណាមួយនោះវានឹងបង្កើតជា Event ដើម្បីទៅជំនើការកូដណាមួយដែលបានកំណត់ដោយ JavaScript ក្នុង។

Event អាចត្រូវបានកើតឡើងនៅពេល

- > Mouse Click
- > Web page ឬ Image ត្រូវបាន Load
- > Mouse Over នៅលើ តំបន់ណាមួយក្នុង Web Page
- > Submit Data ទៅក្នុង HTML Form ជាដើម ។

Event តែងតែបង្កើតឡើងភ្ជាប់ជាមួយនឹង Function ណាមួយហើយ Function អាចធ្វើការងារណាមួយនៅពេលដែល Event ត្រូវបានកើតឡើងតែប៉ុណ្ណោះ។

Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function displayDate()
{
document.getElementById("demo").innerHTML=Date();
}
</script>
</head>

<body>

<h1>My First Web Page</h1>

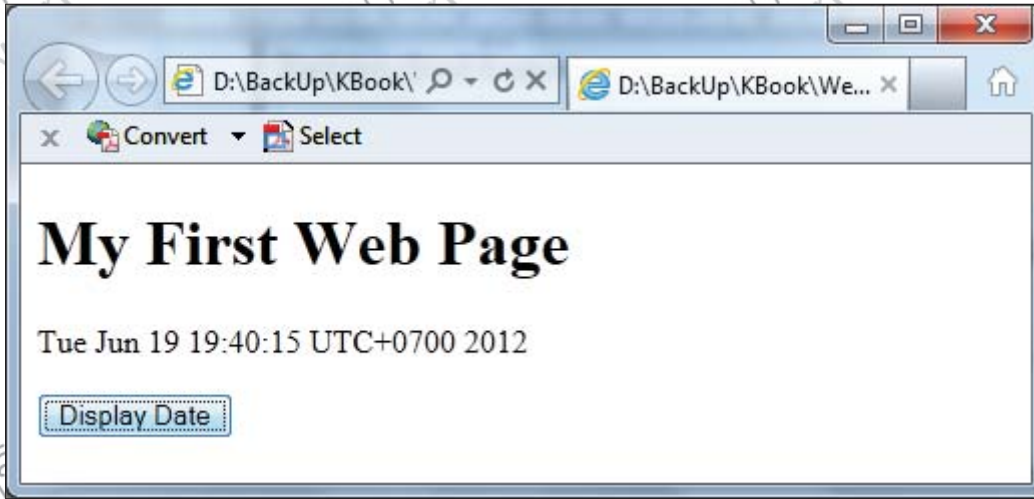
<p id="demo"></p>

<button type="button" onclick="displayDate()">Display
Date</button>

</body>
</html>

```

Output:



2. onLoad Event:

onLoad Event នេះត្រូវបាន Tricker នៅពេលដែល Webpage ត្រូវបានបើកឡើង។

Ex:

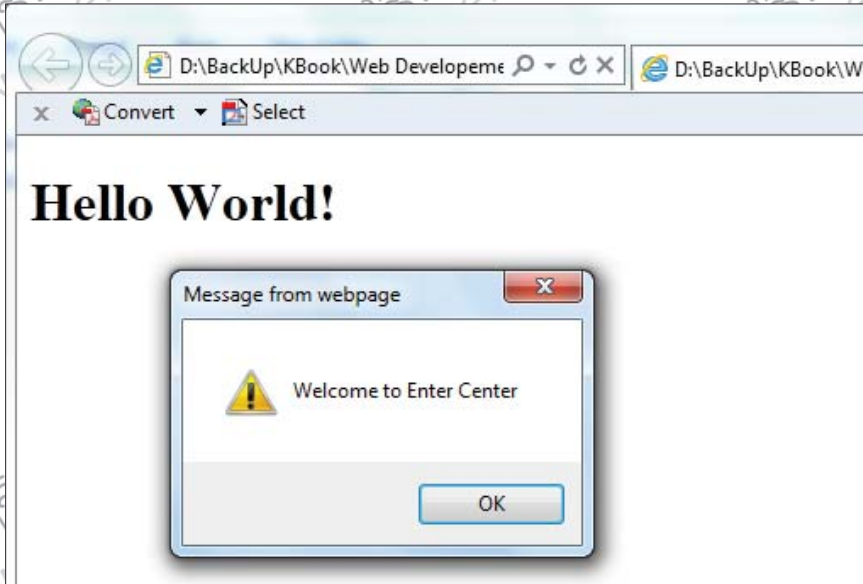
```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function load()
{
alert("Welcome to Enter Center");
}
</script>
</head>

<body onload="load()">
<h1>Hello World!</h1>
</body>
</html>

```

Output:



3. onUnload Event:

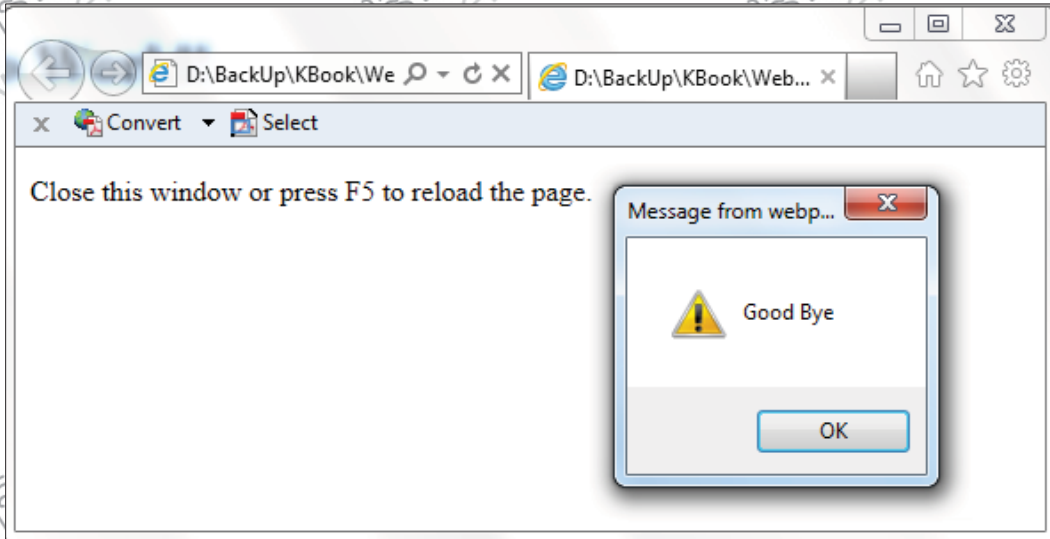
ចំពោះ onUnLoad Event នេះត្រូវបាន Tricker នៅពេលដែល Webpage ត្រូវបាន reload (refresh) ឡើងវិញ ឬ បិទ វាចោលដោយទៅបើក Webpage ដទៃទៀត។

Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function OnUnload()
{
alert ("Good Bye") ;
}
</script>
</head>
<body onunload="OnUnload () ">
<p>Close this window or press F5 to reload the
page.</p>
</body>
</html>
  
```

Output:



4. **onFocus Event:**

ចំពោះ onFocus Event នេះត្រូវបាន Trigger នៅពេលយើងដាក់ Mouse Cursor នៅលើ Element ណាមួយ។

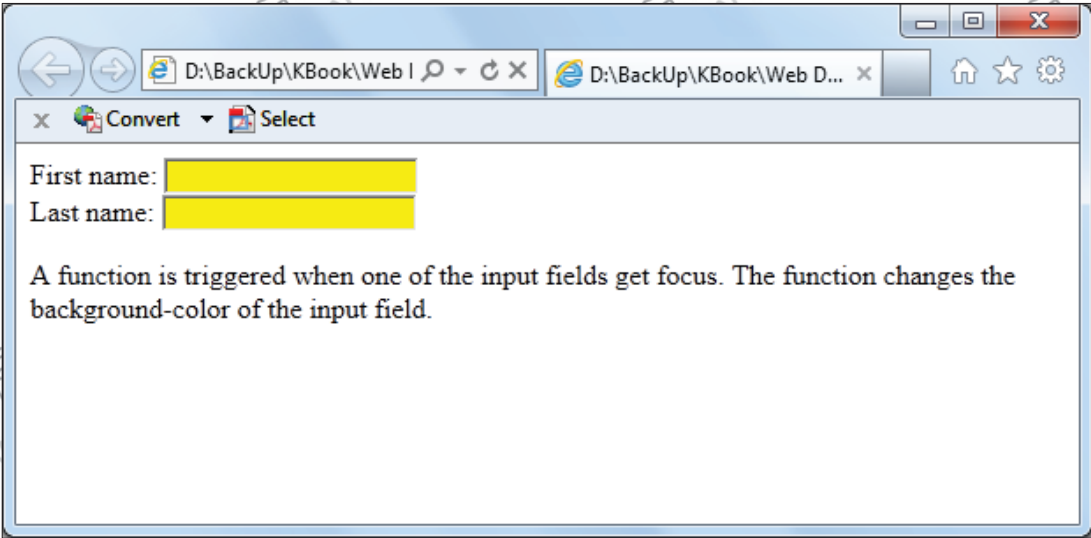
Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function setStyle(x)
{
document.getElementById(x).style.background="yellow";
}
</script>
</head>
<body>
First name: <input type="text" id="fname"
onfocus="setStyle(this.id)" />
<br />
Last name: <input type="text" id="lname"
onfocus="setStyle(this.id)" />
<p>A function is triggered when one of the input fields
get focus. The function changes the background-color of
the input field.</p>
</body>
</html>

```

Output:



5. **onBlur Event:**

ចំពោះ onBlur Event នេះត្រូវបាន Tricker នៅពេលយើងដក Mouse Pointer ចេញពី Object ណាមួយ។

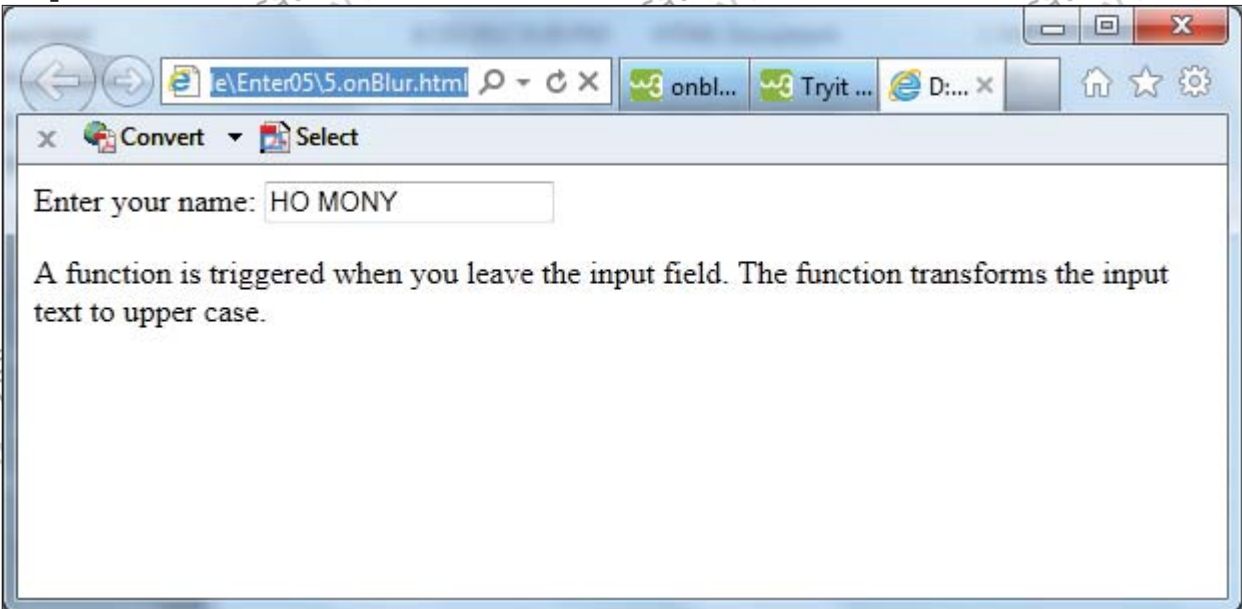
Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function upperCase()
{
var x=document.getElementById("fname");
x.value=x.value.toUpperCase();
}
</script>
</head>
<body>
Enter your name: <input type="text" id="fname"
onchange="upperCase()" />
<p>A function is triggered when the input field is
changed. The function transforms the input text to
upper case.</p>
<p>Write some text in the input field, and the click
outside the input field to run the function.</p>
</body>
</html>

```

Output:



6. onChange Event:

ចំពោះ: onUnLoad Event នេះត្រូវបាន Tricker នៅពេលយើងធ្វើការកែប្រែទិន្នន័យនៅក្នុង Input Field ណាមួយ។

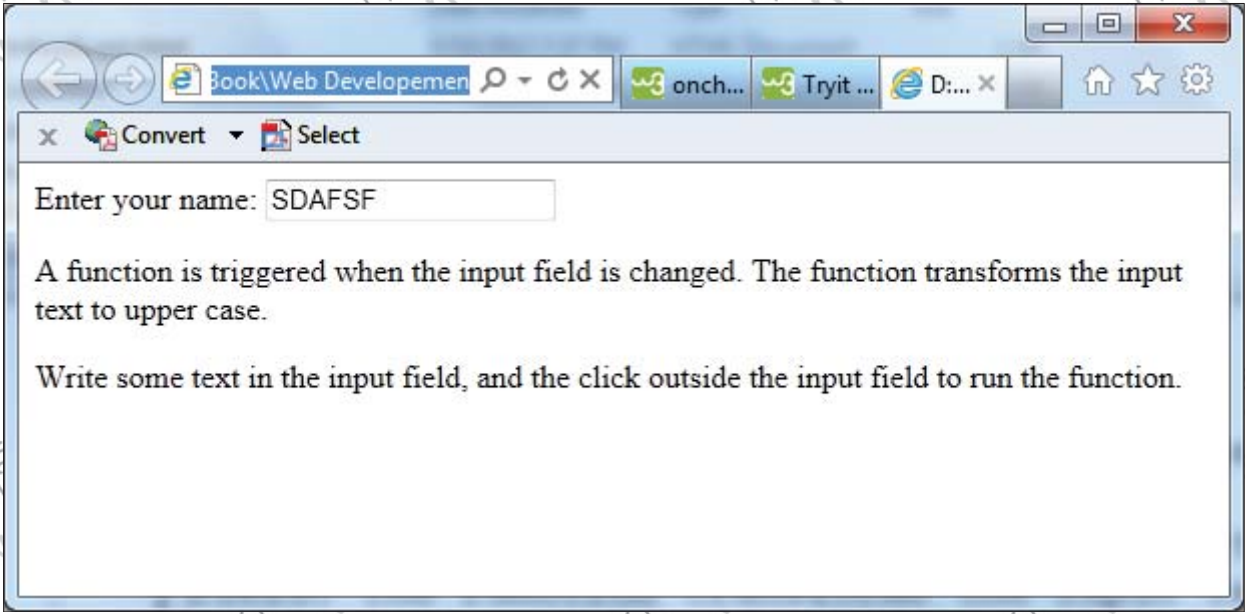
Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function upperCase()
{
var x=document.getElementById("fname");
x.value=x.value.toUpperCase();
}
</script>
</head>
<body>
Enter your name: <input type="text" id="fname"
onblur="upperCase()" />
<p>A function is triggered when you leave the input
field. The function transforms the input text to upper
case.</p>
</body>
</html>

```

Output:



7. onMouseOver Event:

ចំពោះ onUnLoad Event នេះត្រូវបាន Tricker នៅពេលយើងធ្វើការកែប្រែទិន្នន័យនៅក្នុង Input Field ណាមួយ។

Ex1:

```

<html>
<head>
</head>
<body>
<h1>Transparency Image</h1>

</body>
</html>

```

Output1:



Ex2:

```

<!doctype html>
<html>
<head>
</head>
<body>

<div style="height:400px;width:1200px;border:0px solid
black;display:table-cell; vertical-align:bottom;text-
align:center">
<a href="#"
onmouseover="document.Enter1.src='HTMLBig.jpg'"
onmouseout="document.Enter1.src='HTMLSmall.jpg'">
</a>

<a href="#"
onmouseover="document.Enter2.src='CSSBig.jpg'"
onmouseout="document.Enter2.src='CSSSmall.jpg'">
</a>

<a href="#"
onmouseover="document.Enter3.src='PHPBig.jpg'"
onmouseout="document.Enter3.src='PHPSmall.jpg'">
</a>
</div>
</body>
</html>

```

Output2:



8. onSubmit Event:

onSubmit Event គឺត្រូវបានដំនើការនៅពេលដែល Submit Button ត្រូវបានចុច។

Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function greeting()
{
alert("Welcome " +
document.forms["frm1"]["fname"].value + "!")
}
</script>
</head>

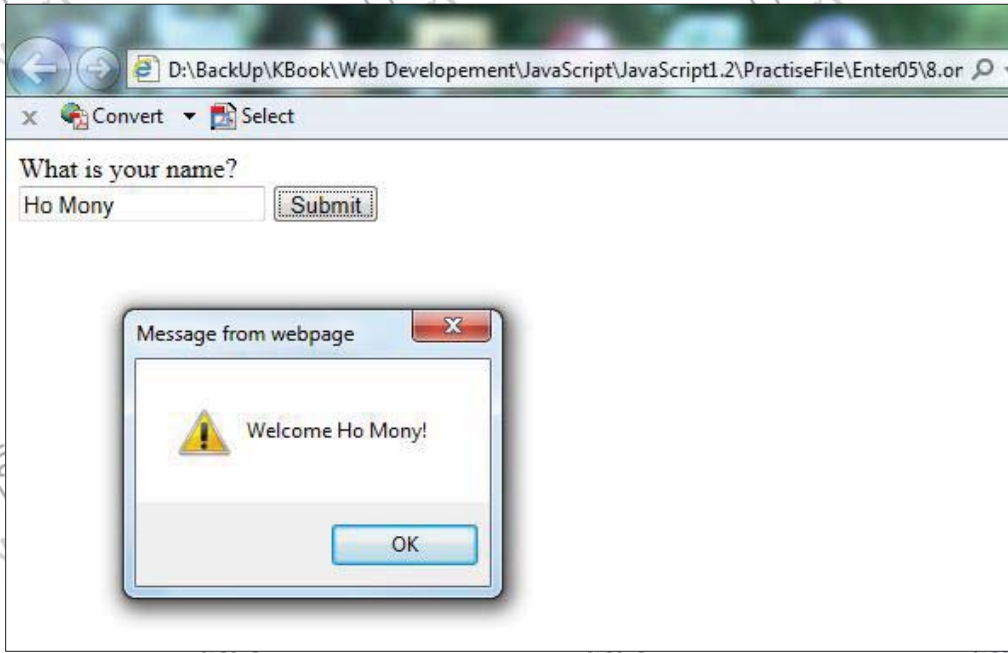
<body>

What is your name?<br />
<form name="frm1" action="tryjsref_form_onsubmit.htm"
onsubmit="greeting()">
<input type="text" name="fname" />
<input type="submit" value="Submit" />
</form>

</body>
</html>

```


Output:

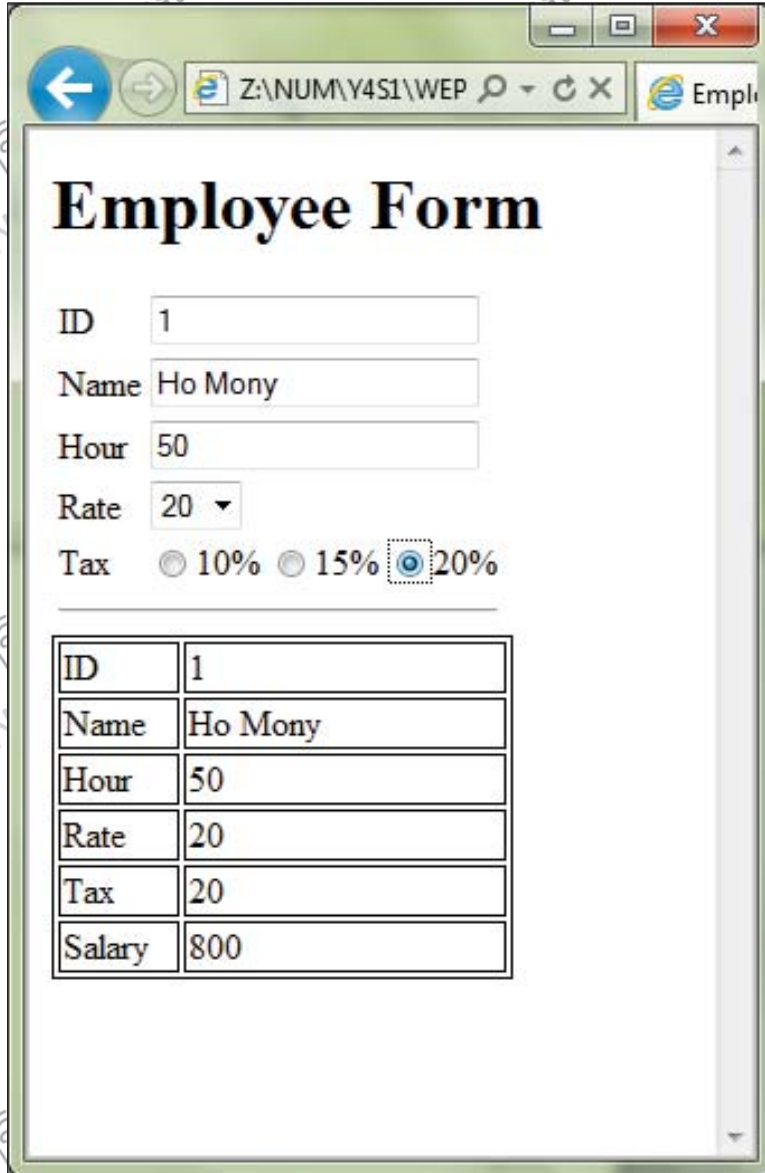


9. ប្រើប្រាស់ `webpage` ដើម្បីបង្កើតការ `onLoad` Event:

Ex:

```
<body onLoad = "javascript:window.open
('deals.html', 'Deals', 'height=250,width=350')">
```

10. សំណាក:



មេរៀនទី ៨: Form Validation

1. និយមន័យ:

ជាទូទៅ JavaScript ក៏ត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការត្រួតពិនិត្យផ្ទៀងផ្ទាត់មើលភាពត្រឹមត្រូវនៅក្នុង Form មុនពេលបញ្ជូនទិន្នន័យទៅកាន់ Server ផងដែរ។

ទិន្នន័យដែលត្រូវត្រួតពិនិត្យគឺអាចពាក់ព័ន្ធនឹងព័ត៌មានមួយចំនួនដូចជា:

- > ត្រួតពិនិត្យទៅលើ Fields សំខាន់ៗមួយចំនួនដែល users មិនបានបំពេញ
- > ផ្ទៀងផ្ទាត់ពីទម្រង់ទិន្នន័យដែលមានប្រភេទជា Email Address
- > ឆែកមើលនូវរាល់ទិន្នន័យដែលបញ្ចូលជាទម្រង់កាលបរិច្ឆេទ
- > ឬអនុញ្ញាតិច User អាចបញ្ចូលទិន្នន័យប្រភេទជាលេខតែប៉ុណ្ណោះ ។

2. Required Fields:

តាមឧទាហរណ៍ខាងក្រោមបង្ហាញពីការដាក់លក្ខខណ្ឌ គឺប្រសិនបើ Field ណាមួយគ្មានទិន្នន័យនោះវានឹង alert message ដើម្បីបញ្ជាក់ថា Field គ្មានទិន្នន័យ ហើយមិនធ្វើការ Submit ទៅកាន់ server នោះទេ។

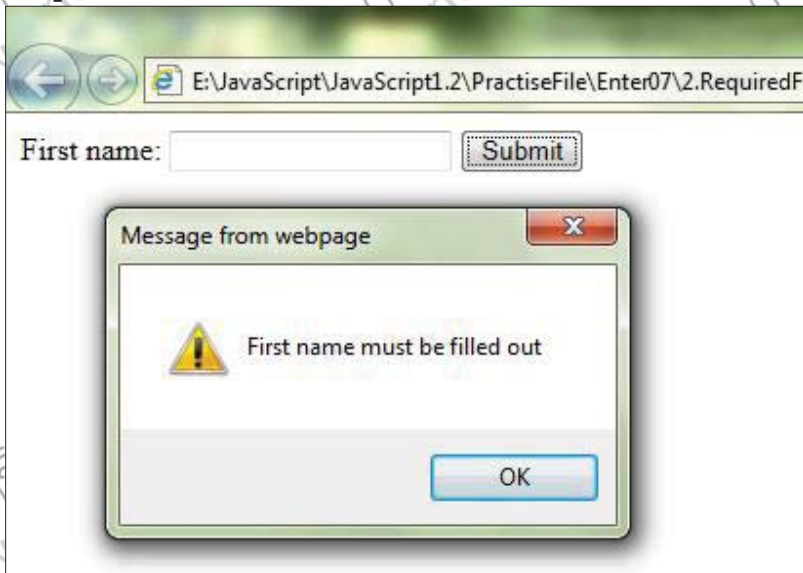
Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function validateForm()
{
var x=document.forms["myForm"]["fname"].value;
if (x==null || x=="")
{
alert("First name must be filled out");
return false;
}
}
</script>
</head>
<body>
<form name="myForm" action="demo_form.asp"
onsubmit="return validateForm()" method="post">
First name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
</body>
</html>

```

Output:



3. Email Validation:

តាមឧទាហរណ៍ខាងក្រោមបង្ហាញពីការដាក់លក្ខខណ្ឌ ដើម្បីរកមើលភាពត្រឹមត្រូវនៃ Email ដែលបានបញ្ចូល ដោយក្នុងនោះ: Email ត្រូវមាន @ Sign និង Dot (.) Sign យ៉ាងតិចមួយ។

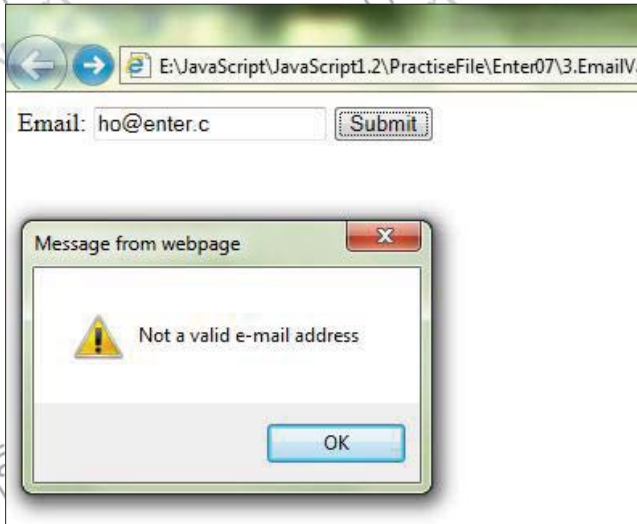
Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function validateForm()
{
var x=document.forms["myForm"]["email"].value;
var atpos=x.indexOf("@");
var dotpos=x.lastIndexOf(".");
if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)
{
alert("Not a valid e-mail address");
return false;
}
}
</script>
</head>
<body>
<form name="myForm" action="demo_form.asp"
onsubmit="return validateForm();" method="post">
Email: <input type="text" name="email">
<input type="submit" value="Submit">
</form>
</body>
</html>

```

Output:



4. Validate Input Character:

តាមឧទាហរណ៍ខាងក្រោមបង្ហាញពីការដាក់លក្ខខណ្ឌ ដោយកំណត់ទិន្នន័យដែលបញ្ជូនត្រូវតែជា អក្សរអូប៉ុណ្ណោះ។

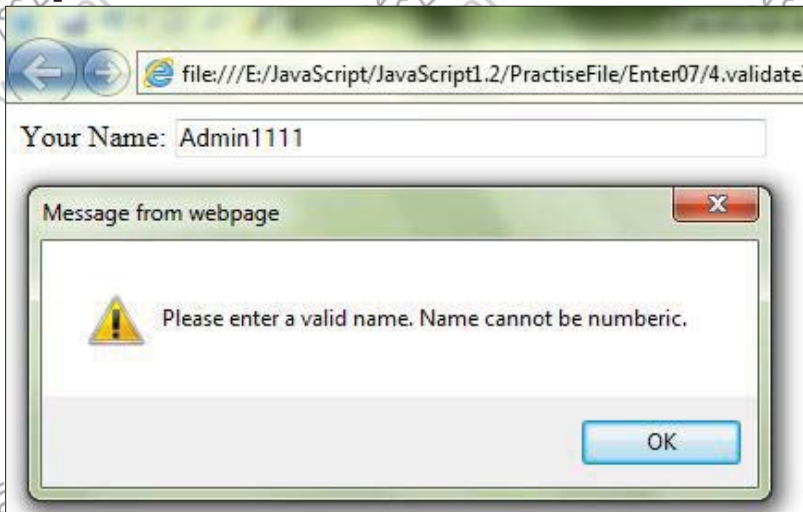
Ex:

```

<!DOCTYPE html>
<html>
<head>
</head>
<body>
<form>
Your Name:
<input type="text" name="custname" size="50"
maxlength="45"
onChange="var pattern=/[0-9]/;
if(pattern.test(this.value))
alert('Please enter a valid name. Name cannot be
numeric.')">
</form>
</body>
</html>

```

Output:



ក្នុងនោះយើងក៏អាចធ្វើការកំណត់លក្ខខណ្ឌមួយចំនួនផ្សេងៗទៀតដូចជា:

Value	Users Cannot Enter
A-Z	Uppercase letters
a-z	Lowercase letters
0-9	Any number
\d	Any number
\s	Any spacing
\w	Any letters, any numbers, or the underscore character

5. Validate Input Number:

តាមឧទាហរណ៍ខាងក្រោមបង្ហាញពីការដាក់លក្ខខណ្ឌ ដោយកំណត់ទិន្នន័យដែលបញ្ជូនត្រូវតែជា លេខប៉ុណ្ណោះ។

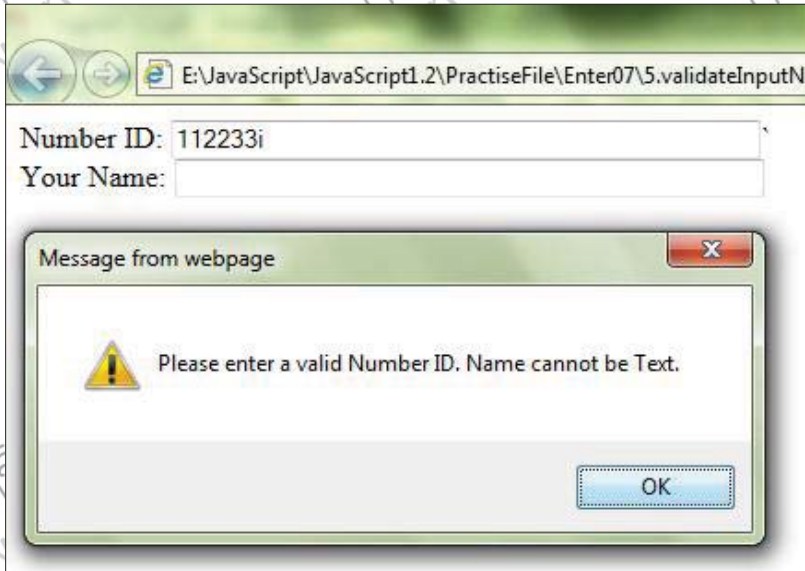
Ex:

```

<!DOCTYPE html>
<html>
<head>
</head>
<body>
<form>
Number ID:
<input type="text" name="ID" size="50" maxlength="45"
onChange="var pattern=/[A-Za-
z]/;if(pattern.test(this.value))
alert('Please enter a valid Number ID. Name cannot be
Text.')">`
<br>
Your Name:
<input type="text" name="custname" size="50"
maxlength="45"
onChange="var pattern=/[0-
9]/;if(pattern.test(this.value))
alert('Please enter a valid name. Name cannot be
numeric.')">
</form>
</body>
</html>

```

Output:



6. Validate Minimum/Maximum Length:

តាមឧទាហរណ៍ខាងក្រោមបង្ហាញពីការដាក់លក្ខខណ្ឌ ដោយកំនត់ទទឹងនៃវិញ្ញាណប័ណ្ណដែលបញ្ជូនត្រូវតែជា លេខប៉ុណ្ណោះ។

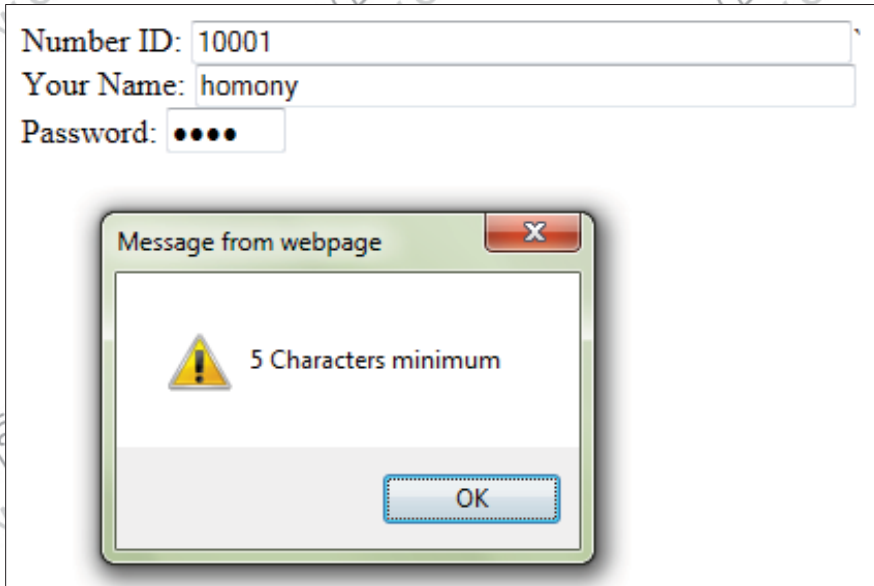
Ex:

```

<!DOCTYPE html>
<html>
<head>
</head>
<body>
<form>
Number ID:
<input type="text" name="ID" size="50" maxlength="45"
onChange="var pattern=/[A-Za-
z]/;if(pattern.test(this.value))
alert('Please enter a valid Number ID. Name cannot be
Text.')">`
<br>
Your Name:
<input type="text" name="custname" size="50"
maxlength="45"
onChange="var pattern=/[0-
9]/;if(pattern.test(this.value))
alert('Please enter a valid name. Name cannot be
numeric.')">
</form>
</body>
</html>

```

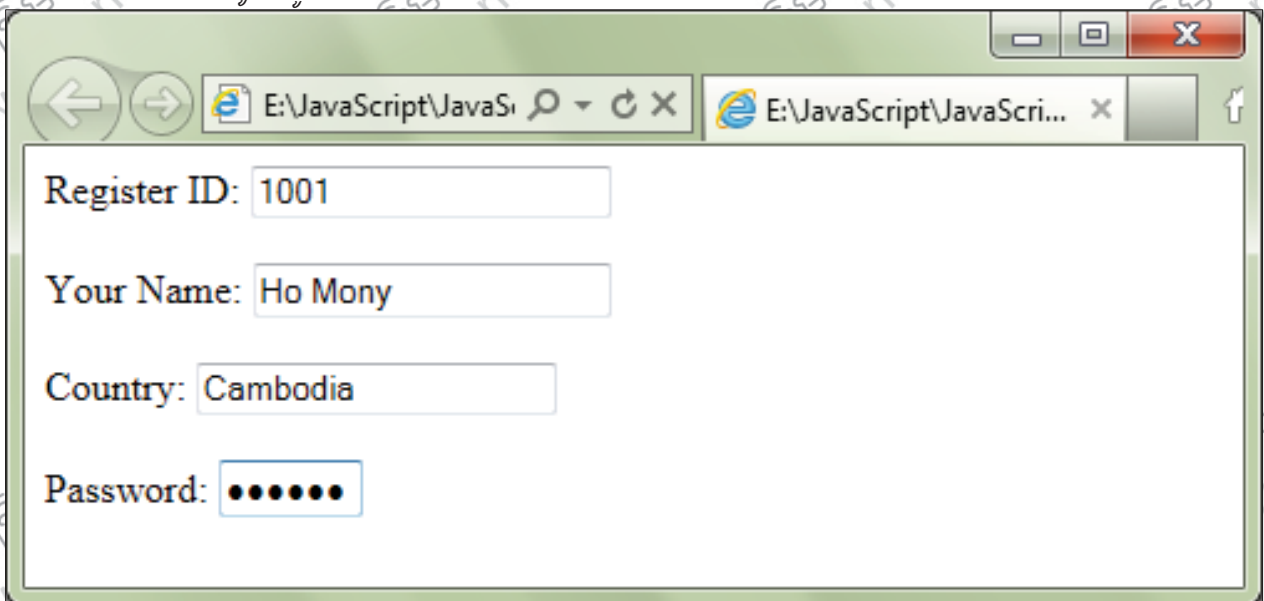
Output:



7. លំហាត់:

ចូរបង្កើត Form មួយហើយកំណត់ Validation ទៅ៤វាដូចខាងក្រោម:

- > Field Register ID: ត្រូវបញ្ចូលជាលេខ
- > Field Your Name: ត្រូវបញ្ចូលជាអក្សរ
- > Field Country: ត្រូវបញ្ចូលជាអក្សរ
- > Field Password: ត្រូវបញ្ចូលចាប់ពី 8 តួឡើងទៅ



មេរៀនទី 9: ការបង្កើត Frame Layout

1. និយមន័យ:

ជាធម្មតា Browser windows គឺតែងតែធ្វើការ display ព័ត៌មានរបស់ Webpage ជាលក្ខណៈ single HTML Document ឬជា Single Frame ។ ប្រសិនបើទិន្នន័យនៅក្នុង Webpage មានទំហំលើសពីទំហំរបស់ Frame នោះវានឹងបង្កើន Scrollbar មកជាមួយដោយស្វ័យប្រវត្តិ ។

យើងក៏អាចធ្វើការបង្ហាញ Webpage ច្រើនក្នុងពេលតែមួយបានផងដែរ ដោយធ្វើការបំបែកវាទៅជា Multiple Frames ហើយកំណត់ទំហំរបស់វានីមួយៗនៅក្នុងតំបន់ផ្សេងៗគ្នានៃ Screen Monitor ។ នៅក្នុង Frames នីមួយៗមានព្រំដែនតំបន់ផ្សេងៗគ្នា និងមាន Scrollbar ផ្សេងៗគ្នាផងដែរ។

ដើម្បីអាចប្រើប្រាស់ Frame បានគឺត្រូវ Declare DocType នៅខាងលើបំផុតនៃ Code ផងដែរ ដោយក្នុងនោះ DocType ត្រូវបានបែងចែកជា 3 ប្រភេទដែលមានដូចជា:

> HTML 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/REChtml4/loose.dtd">
```

> HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
"http://www.w3.org/TR/REC-html4/strict.dtd">
```

> HTML 4.01 Frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/REC-html4/frameset.dtd">
```

ក្នុងចំណោម Doc Type ទាំង 3 ខាងលើ HTML 4.01 Frameset គឺជា Doc Type ដែលត្រូវប្រកាសនៅពេលបង្កើត Frameset ។

2. ការបង្កើត Frameset & Frame:

Framset គឺជា Tag មួយដែលត្រូវបានប្រើប្រាស់ដើម្បីរៀបចំបង្កើតជា Frame Layout ដូច្នោះដើម្បីបង្កើតនោះយើងត្រូវសរសេរ <frameset> </frameset> នៅបន្ទាប់ </head> ។ ចំនែក Frame វិញគឺជាការទាញយក Webpage មួយៗផ្សេងៗគ្នាមកដាក់នៅក្នុងតំបន់ណាមួយនៃ Screen ដើម្បីបង្ហាញមក ដោយត្រូវសរសេរ <frame> ហើយត្រូវដាក់នៅក្នុងចន្លោះ <frameset></frameset> ជានិច្ច។

Ex:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/REC-html4/frameset.dtd">
<html>
<head><title>Creating Frameset</title></head>

<frameset rows="200,*">
<frame name="banner" src="Banner.html">
<frame name="main" src="Content.html">
</frameset>

</html>

```

Output:



3. របៀបកំណត់ Frame Border:

Ex:

```

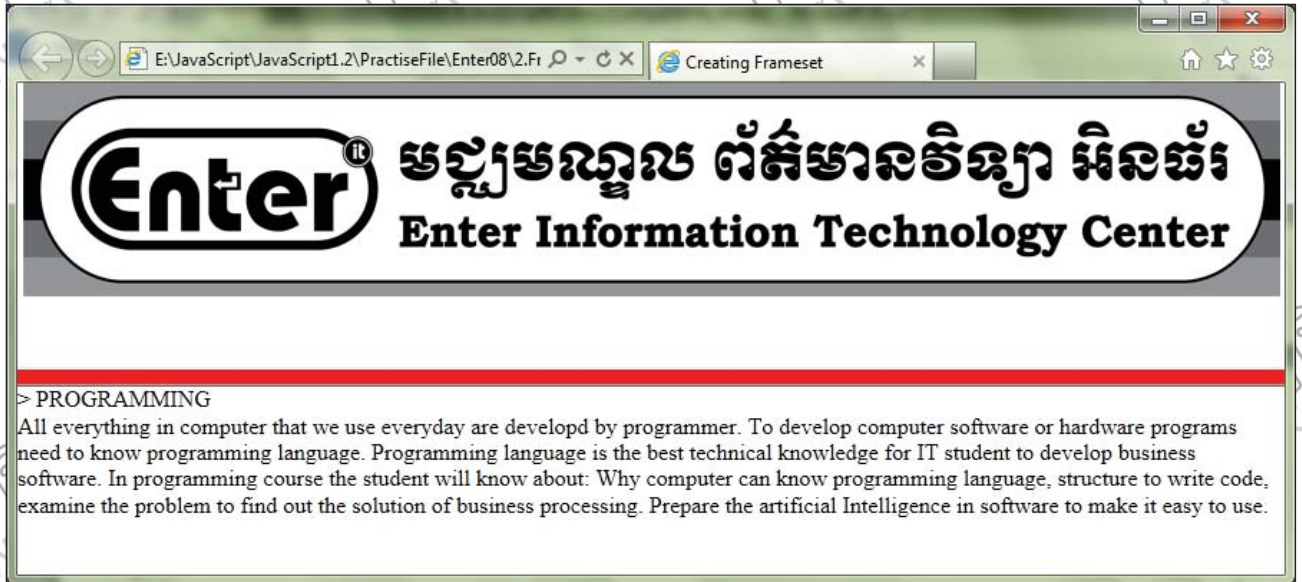
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/REC-html4/frameset.dtd">
<html>
<head><title>Creating Frameset</title></head>

<frameset rows="200,*" border="10" bordercolor="red">
<frame name="banner" src="Banner.html">
<frame name="main" src="Content.html">
</frameset>

</html>

```

Output:



4. របៀបកំណត់ Frame Margin:

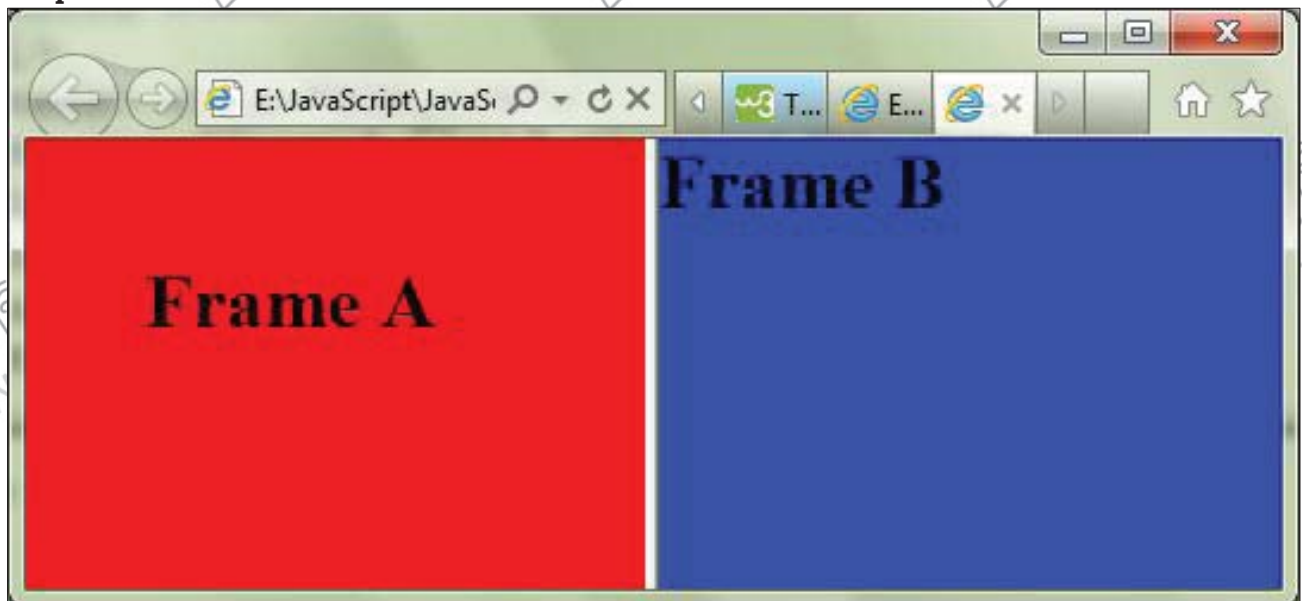
Ex:

```

<!DOCTYPE html>
<html>
<frameset cols="50%,50%">
<frame src="FrameA.html" marginwidth="50"
marginheight="50"/>
<frame src="FrameB.html" marginwidth="0" />
</frameset>
</html>

```

Output:



5. របៀបកំណត់ **Frame មិន ២ Resize:**

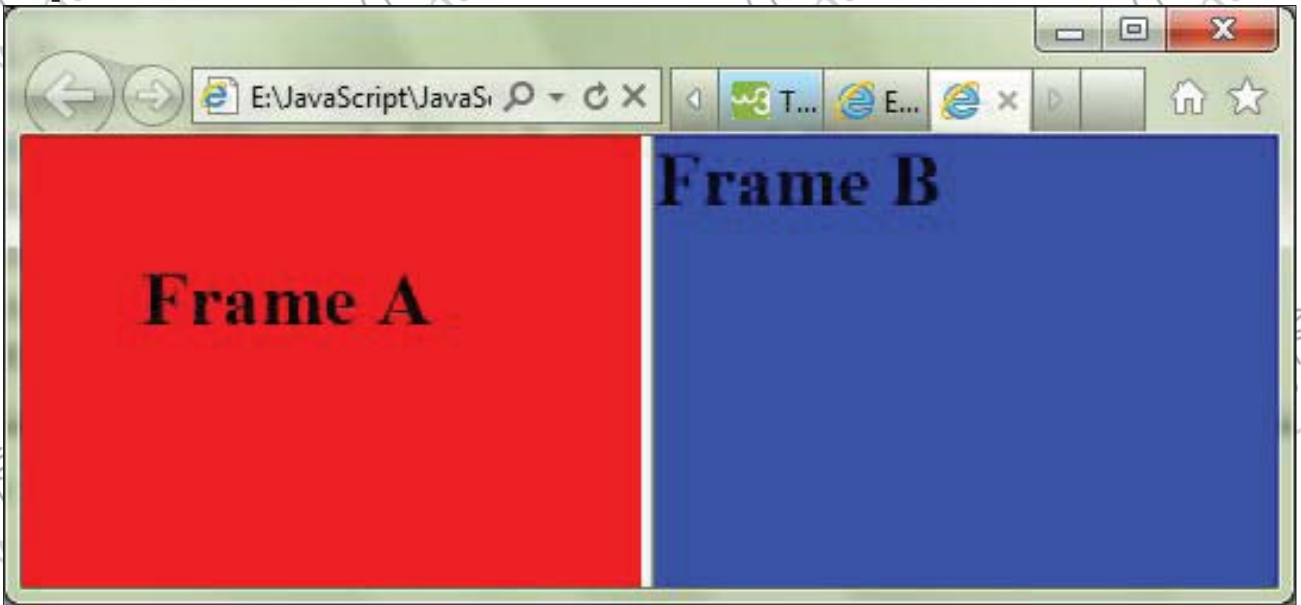
Ex:

```

<!DOCTYPE html>
<html>
<frameset cols="50%,50%">
<frame src="FrameA.html" marginwidth="50"
marginheight="50"/>
<frame src="FrameB.html" marginwidth="0" noresize/>
</frameset>
</html>

```

Output:



6. របៀបកំណត់ **Display Scrollbar:**

តាមធម្មតា Scrollbar នឹងត្រូវបានបង្ហាញជាមួយ Frame នៅពេលណាដែលមានទិន្នន័យលើសពីទំហំរបស់ Frame ហើយយើងអាចកំណត់ Frame នីមួយៗមានបង្ហាញ Scrollbar ឬមិនបង្ហាញបានដោយត្រូវប្រើប្រាស់ scrolling ជាមួយនឹង no ដើម្បីមិនបង្ហាញឬ yes ដើម្បីបង្ហាញ។

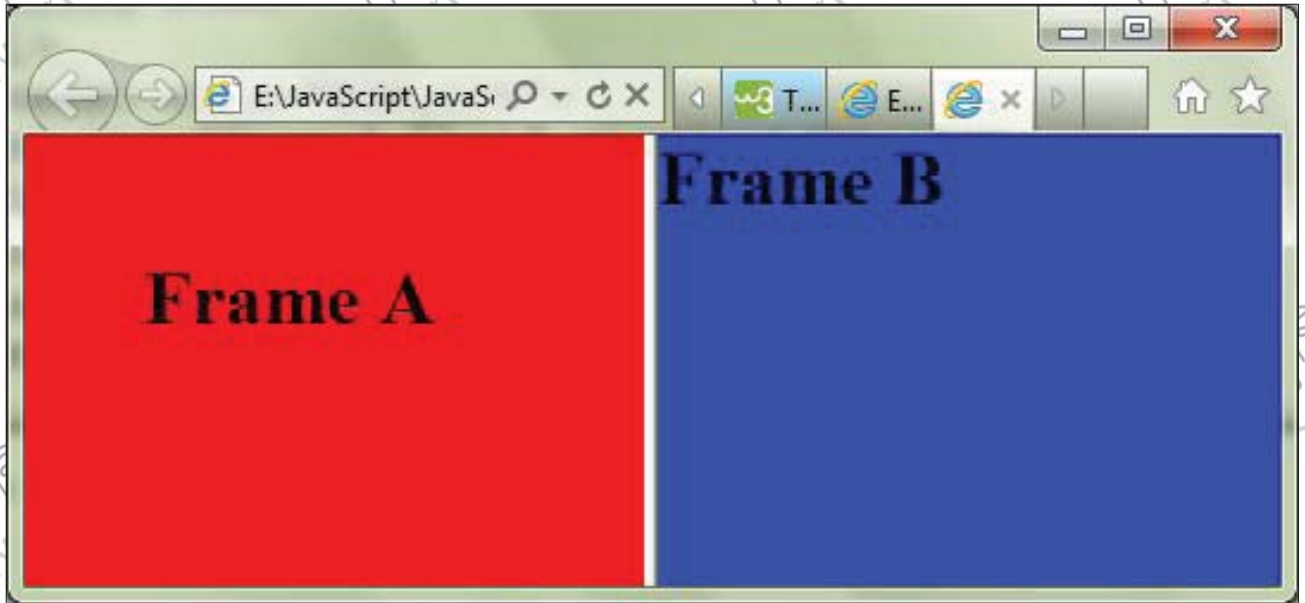
Ex:

```

<!DOCTYPE html>
<html>
<frameset cols="50%,50%">
<frame src="FrameA.html" marginwidth="50"
marginheight="50"/>
<frame src="FrameB.html" marginwidth="0" scrolling="no"/>
</frameset>
</html>

```

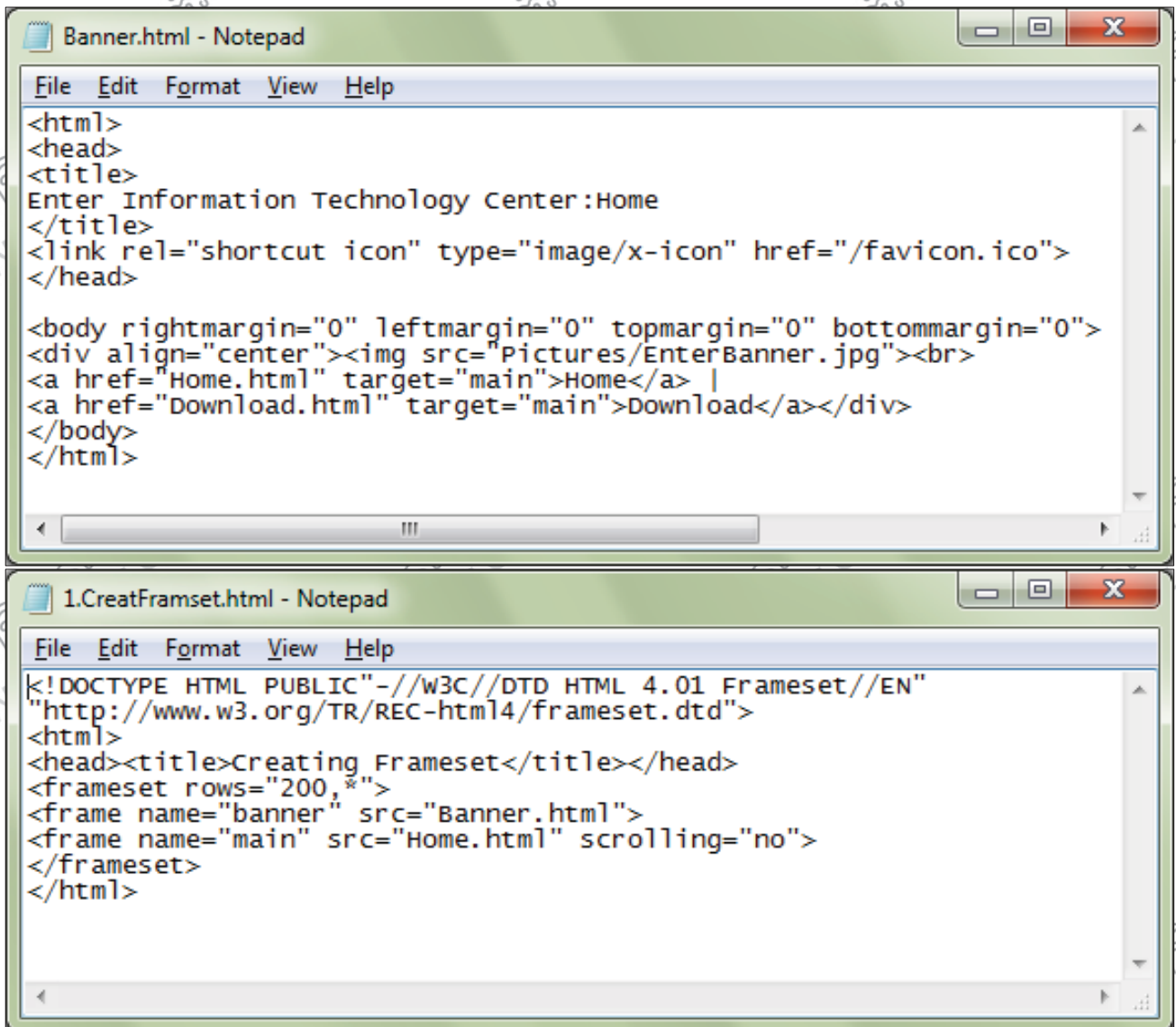
Output:



7. របៀបបើក Webpage លើ Frame ណាមួយ:

យើងអាចធ្វើការបើក Webpage ណាមួយតាមរយៈ Link ហើយឲ្យ Display នៅក្នុង Frame ណាមួយក៏បានផងដែរ។

Ex:



8. Make all the Links Open in the Same Frame:

Ex:

```

Banner.html - Notepad
File Edit Format View Help
<html>
<head>
<base target="main">
</head>
<body rightmargin="0" leftmargin="0" topmargin="0" bottommargin="0">
<div align="center"><br>
<a href="Home.html">Home</a> |
<a href="Download.html">Download</a></div>
</body>
</html>
    
```

9. ការបង្កើត Nested Frame:

Ex:

```

<!DOCTYPE HTML>
<html>
<head><title>Creating Frameset</title></head>

<frameset rows="200,*">
<frame name="banner" src="Banner.html">
    <frameset cols="100,*">
    <frame name="nav" src="Navigation.html">
    <frame name="main" src="Home.html">
    </frameset>
</frameset>

</html>
    
```

Output:



10. ការបង្កើត Inline Frame:

Inline Frame គឺជាការបង្កើត Frame មួយដែលស្ថិតនៅក្នុងតំបន់នៃ Frame មួយទៀត។

Ex:

```

<html>
<head>
<title>
Enter Information Technology Center:Home
</title>
<link rel="shortcut icon" type="image/x-icon"
href="/favicon.ico">
</head>

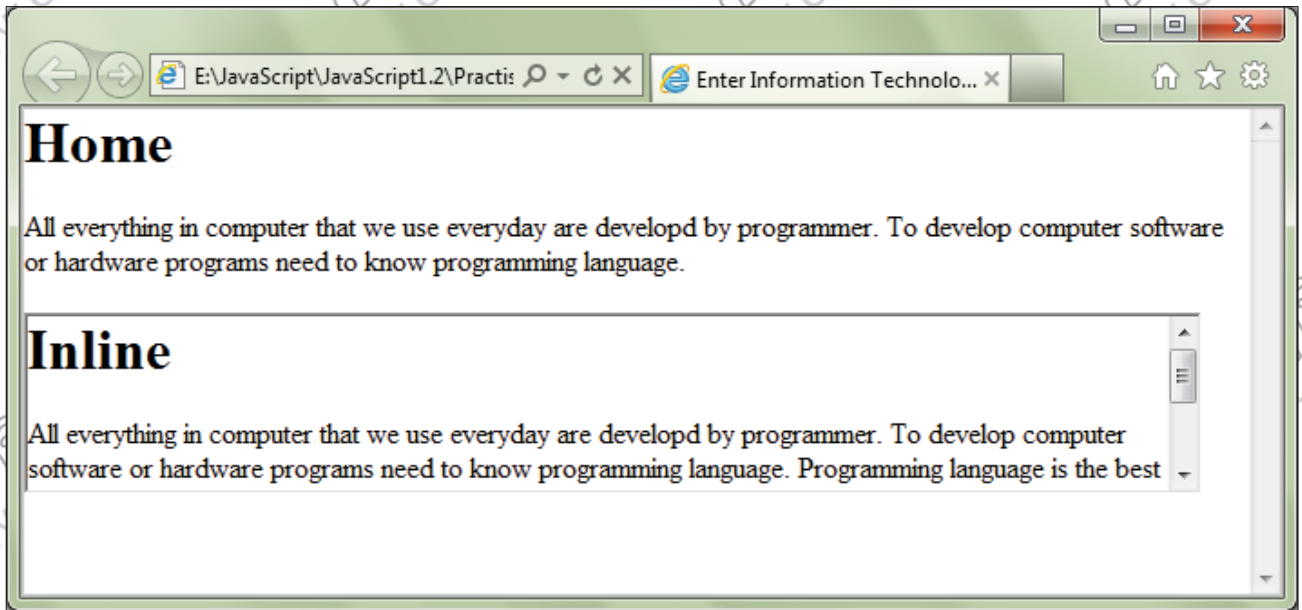
<body rightmargin="0" leftmargin="0" topmargin="0"
bottommargin="0">
<p>
<h1>Home</h1>
All everything in computer that we use everyday are developd
by programmer. To develop
computer software or hardware programs need to know
programming language.
<p>

<iframe src="Inline.html" name="dest" width="650"
height="100">
</iframe>

</body>
</html>

```

Output:



មេរៀនទី 10: ការប្រើប្រាស់ Timing Events

1. Timing Events:

Timing Events គឺជាការ execute បណ្តុំនៃ Code នៅក្នុងកំឡុងពេលចន្លោះណាមួយ។ ដើម្បីងាយស្រួលក្នុងការកំណត់ព័ត៌មានកំឡុងពេលនៃការ Execute នូវបណ្តុំនៃ Code ណាមួយនោះ គឺត្រូវប្រើប្រាស់នូវ Method ពីរដែលមានដូចជា:

- > setInterval() : គឺជា Method ដែល execute នូវបណ្តុំនៃ Code ម្តងហើយម្តងទៀតនៅក្នុងពេលជាក់លាក់ណាមួយ
- > setTimeout(): គឺជា Method ដែល execute នូវបណ្តុំនៃ Code តែម្តងបណ្តោះ ក្រោយពេលរងចាំពេលវេលាដែលបានកំណត់គិតជា milliseconds ។

Table ខ្នាតរបស់ពេលវេលា:

ខ្នាត	ស្មើនឹង
1 h	60 minute
1 minute	60 second
1 second	1000 milliseconds

2. setInterval Method:

setInterval() Method គឺជា Method ដែល execute នូវបណ្តុំនៃ Code ម្តងហើយម្តងទៀតនៅក្នុងពេលជាក់លាក់ណាមួយ ដោយរយៈពេលដែលជាក់លាក់ត្រូវបានគិតជា milliseconds ជានិច្ច។

តាមឧទាហរណ៍ខាងក្រោមយើងនឹងបង្ហាញពីការប្រើប្រាស់ setInterval Method ដើម្បីបង្ហាញពាក្យ Enter រាល់ 3 វិនាទីម្តង។

Ex:

```

<!DOCTYPE html>
<html>
<body>
<p>Click the button to wait 3 seconds, then alert "Hello".</p>
<p>After clicking away the alert box, an new alert box will appear in 3 seconds. This goes on forever...</p>
<button onclick="myFunction()">Try it</button>
<script type="text/javascript">
function myFunction()
{
setInterval(function(){alert("Hello")}, 3000);
}
</script>
</body>
</html>

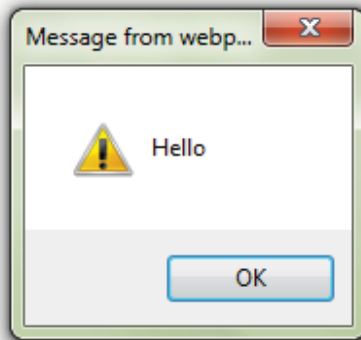
```

Output:

Click the button to wait 3 seconds, then alert "Hello".

After clicking away the alert box, an new alert box will appear in 3 seconds. This goes on forever...

Try it



3. របៀប clearInterval:

យើងត្រូវប្រើប្រាស់ clearInterval() Method ដើម្បីធ្វើការបញ្ឈប់ដំនើការកូដទាំងឡាយណាដែលត្រូវបាន execute ដោយ setInterval() Method ។

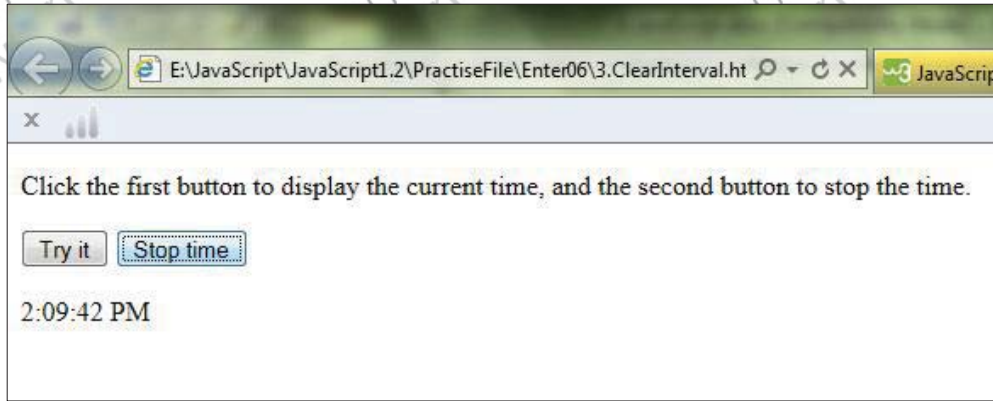
Ex:

```

<!DOCTYPE html>
<html>
<body>
<p>Click the first button to display the current time,
and the second button to stop the time.</p>
<button onclick="myFunction()">Try it</button>
<button onclick="myStopFunction()">Stop time</button>
<script type="text/javascript">
var myVar;
function myFunction()
{
myVar=setInterval(function(){myTimer()},1000);
}
function myTimer()
{
var d=new Date();
var t=d.toLocaleTimeString();
document.getElementById("demo").innerHTML=t;
}
function myStopFunction()
{
clearInterval(myVar);
}</script>
<p id="demo"></p>
</body>
</html>

```

Output:



4. SetTimeout Method:

ជា Method ដែល execute នូវបណ្តុំនៃ Code តែម្តងបណ្តោះ ក្រោយពេលរងចាំពេលវេលាដែល បានកំណត់គឺជា milliseconds ។

Ex:

```

<!DOCTYPE html>
<html>
<body>
<p>Click the button to wait 3 seconds, then alert
"Hello".</p>
<button onclick="myFunction()">Try it</button>
<script type="text/javascript">
function myFunction()
{
setTimeout(function(){alert("Hello")},3000);
}
</script>
</body>
</html>

```

Output:



5. clearTimeout Method:

យើងត្រូវប្រើប្រាស់ clearTimeout() Method ដើម្បីធ្វើការបញ្ឈប់ដំនើការកូដទាំងឡាយណាដែលត្រូវបាន execute ដោយ setTimeout() Method ។

Ex:

```

<!DOCTYPE html>
<html>
<body>
<p>Click the first button alert "Hello" after waiting 3 seconds.</p>
<p>Click the second button to prevent the first function to execute. (You must click it before the 3 seconds are up.)</p>
<button onclick="myFunction()">Try it</button>
<button onclick="myStopFunction()">Stop the alert</button>
<script type="text/javascript">
var myVar;
function myFunction()
{
myVar=setTimeout(function(){alert("Hello")},3000);
}
function myStopFunction()
{
clearTimeout(myVar);
}
</script>
</body>
</html>

```

Output:



6. របៀបបង្ហាញម៉ោងនៅក្នុង Wabpage:

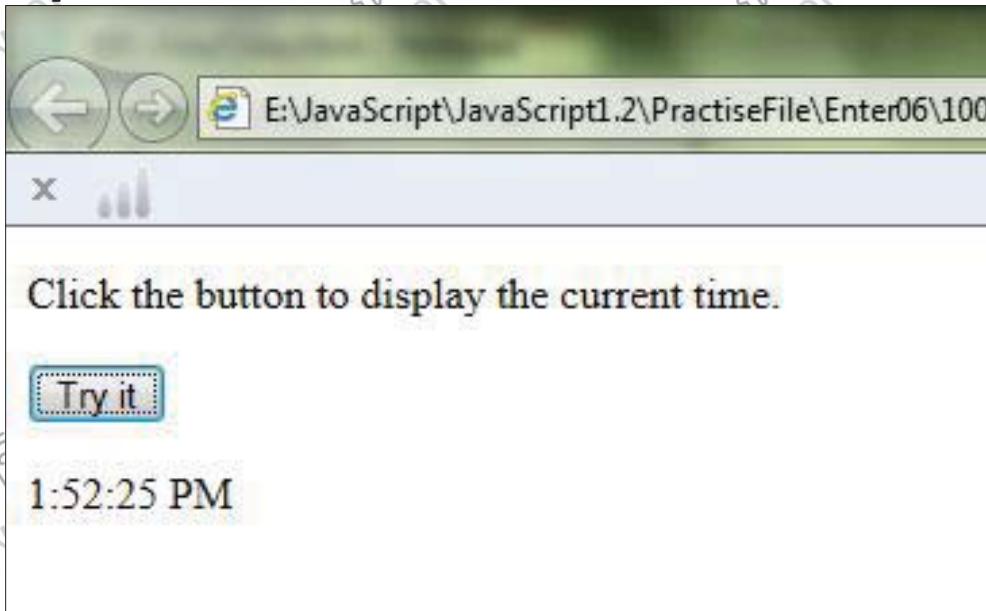
Ex:

```

<!DOCTYPE html>
<html>
<body>
<p>Click the button to display the current time.</p>
<button onclick="myFunction()">Try it</button>
<script type="text/javascript">
function myFunction()
{
setInterval(function() {myTimer() },1000);
}
function myTimer()
{
var d=new Date();
var t=d.toLocaleTimeString();
document.getElementById("demo").innerHTML=t;
}
</script>
<p id="demo"></p>
</body>
</html>

```

Output:



7. របៀបតាម SetTimeout ជាមួយ TextBox:

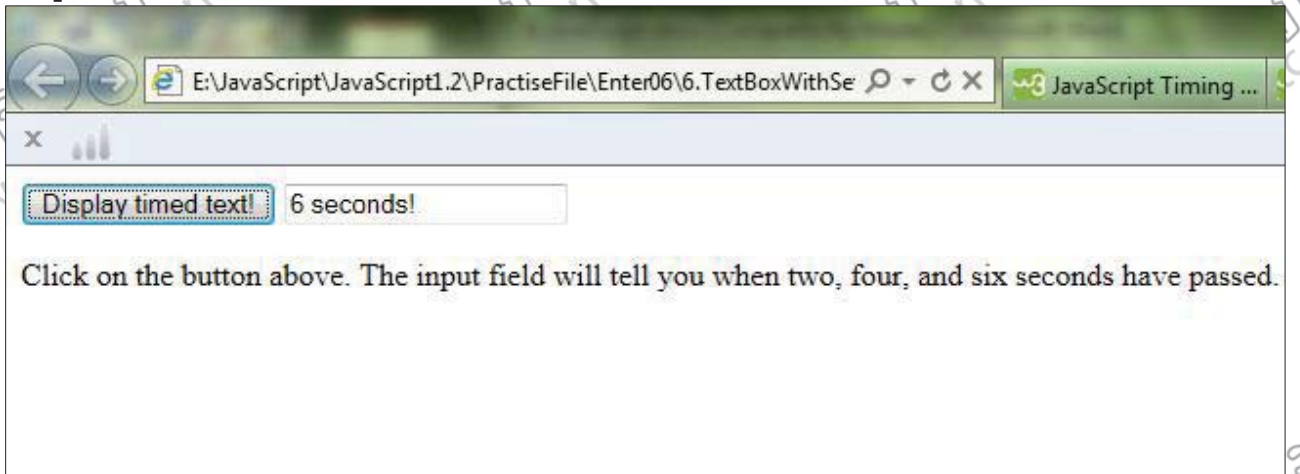
Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function timedText()
{
var
t1=setTimeout("document.getElementById('txt').value='2
seconds!'",2000);
var
t2=setTimeout("document.getElementById('txt').value='4
seconds!'",4000);
var
t3=setTimeout("document.getElementById('txt').value='6
seconds!'",6000);
}
</script>
</head>
<body>
<form>
<input type="button" value="Display timed text!"
onclick="timedText()" />
<input type="text" id="txt" />
</form>
<p>Click on the button above. The input field will tell
you when two, four, and six seconds have passed.</p>
</body>
</html>

```

Output:



8. របៀបបង្កើត Custom Time:

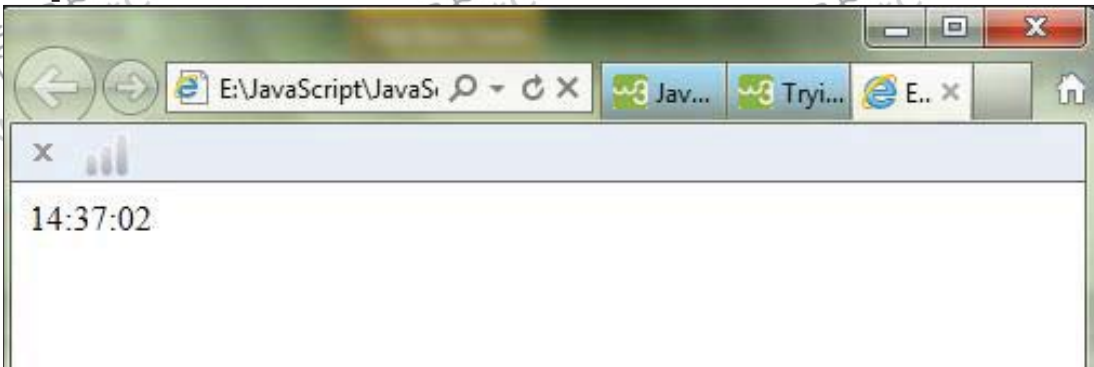
Ex:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function startTime()
{
var today=new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
// add a zero in front of numbers<10
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+":"+m+": "+s;
t=setTimeout('startTime()',100);
}
function checkTime(i)
{
if (i<10)
{
i="0" + i;
}
return i;
}
</script>
</head>
<body onload="startTime()">
<div id="txt"></div>
</body>
</html>

```

Output:



9. របៀបបង្កើត Custom Date:

Ex:

```

<HTML>
<HEAD>
<TITLE>Displaying the current date and time (formatted
example)</TITLE>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
var today = new Date();
var month = today.getMonth();
var displayMonth="";
switch (month) {
case 0 :
displayMonth = "January"
break
case 1 :
displayMonth = "February"
break
case 2 :
displayMonth = "March"
break
case 3 :
displayMonth = "April"
break
case 4 :
displayMonth = "May"
break
case 5 :
displayMonth = "June"
break
case 6 :
displayMonth = "July"
break
case 7 :
displayMonth = "August"
break
case 8 :
displayMonth = "September"
break
case 9 :
displayMonth = "October"
break
case 10 :
displayMonth = "November"
break
case 11 :
displayMonth = "December"

```

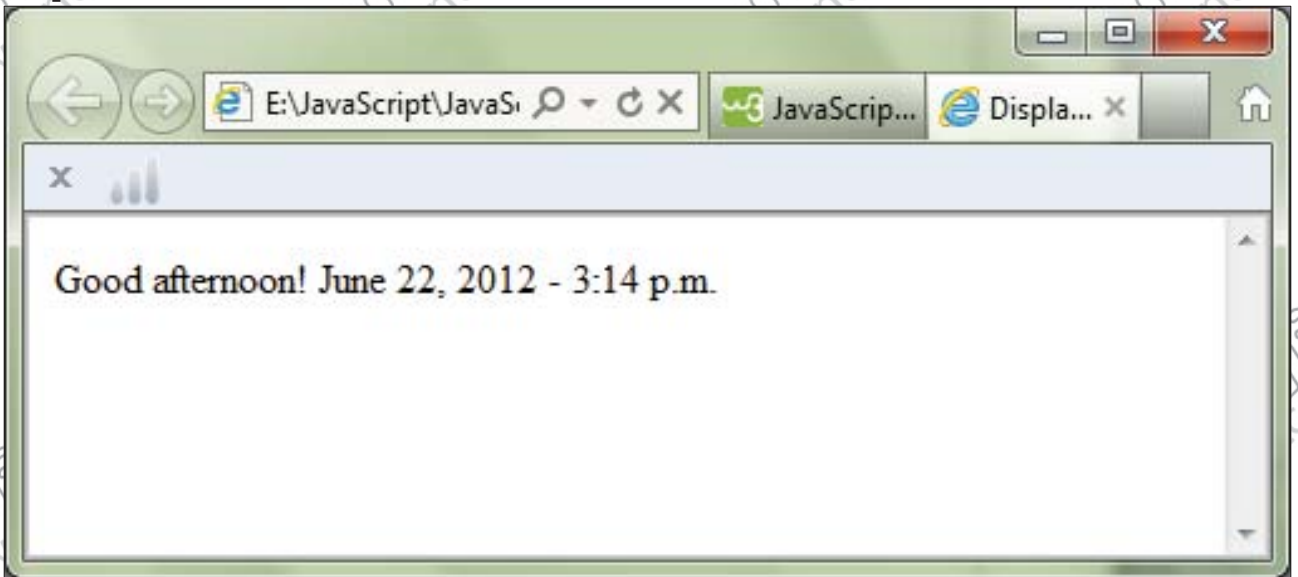


```

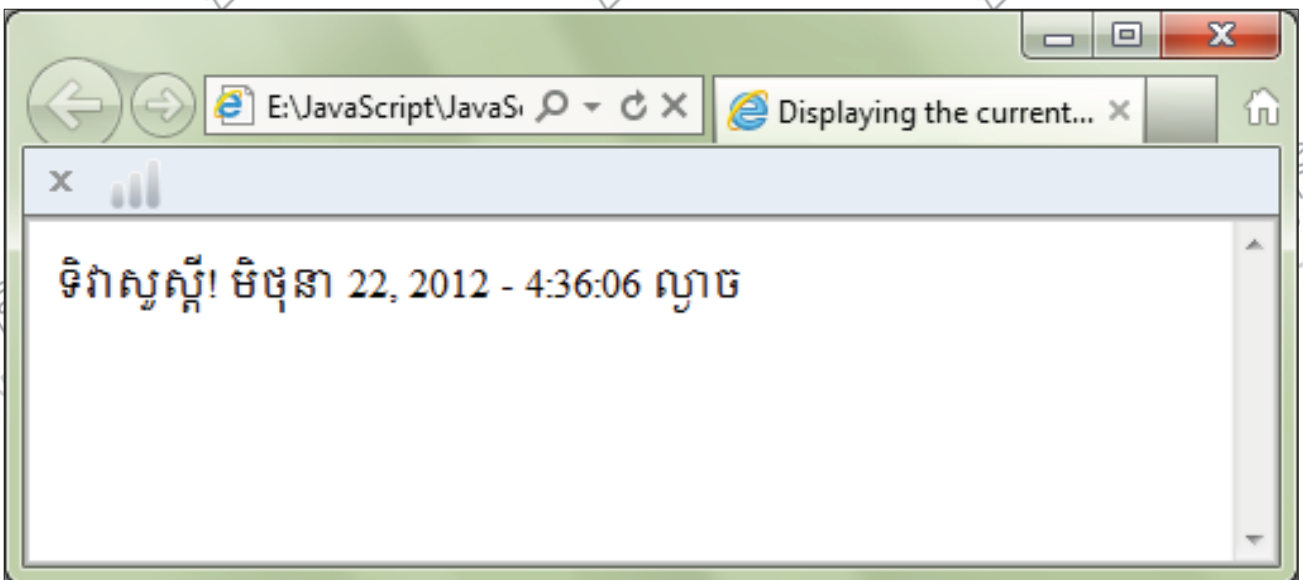
break
default: displayMonth = "INVALID"
}
var hours = today.getHours();
var minutes = today.getMinutes();
var greeting;
var ampm;
if (hours <= 11) {
greeting = "Good morning!";
ampm="a.m.";
if (hours == 0) {
hours = 12;
}
}
else if (hours > 11 && hours < 18) {
greeting = "Good afternoon!";
ampm = "p.m.";
if (hours > 12) {
hours -= 12;
}
}
else if (hours > 17 && hours < 21) {
greeting = "Good evening!";
ampm = "p.m.";
hours -= 12;
}
else if (hours > 20) {
greeting = "Good night!";
ampm = "p.m.";
hours -= 12;
}
if (minutes < 10) {
minutes = "0" + minutes;
}
var displayGreeting = greeting + " " +displayMonth + "
"
+ today.getDate() + ", "
+ today.getYear()
+ " - " + hours + ":" + minutes + " " + ampm
document.writeln(displayGreeting)
</SCRIPT>
</HEAD>
</HTML>

```

Output:



10. សំបាត់:



មេរៀនទី 11: ការបង្កើត Animation

1. ការបង្កើត Animation:

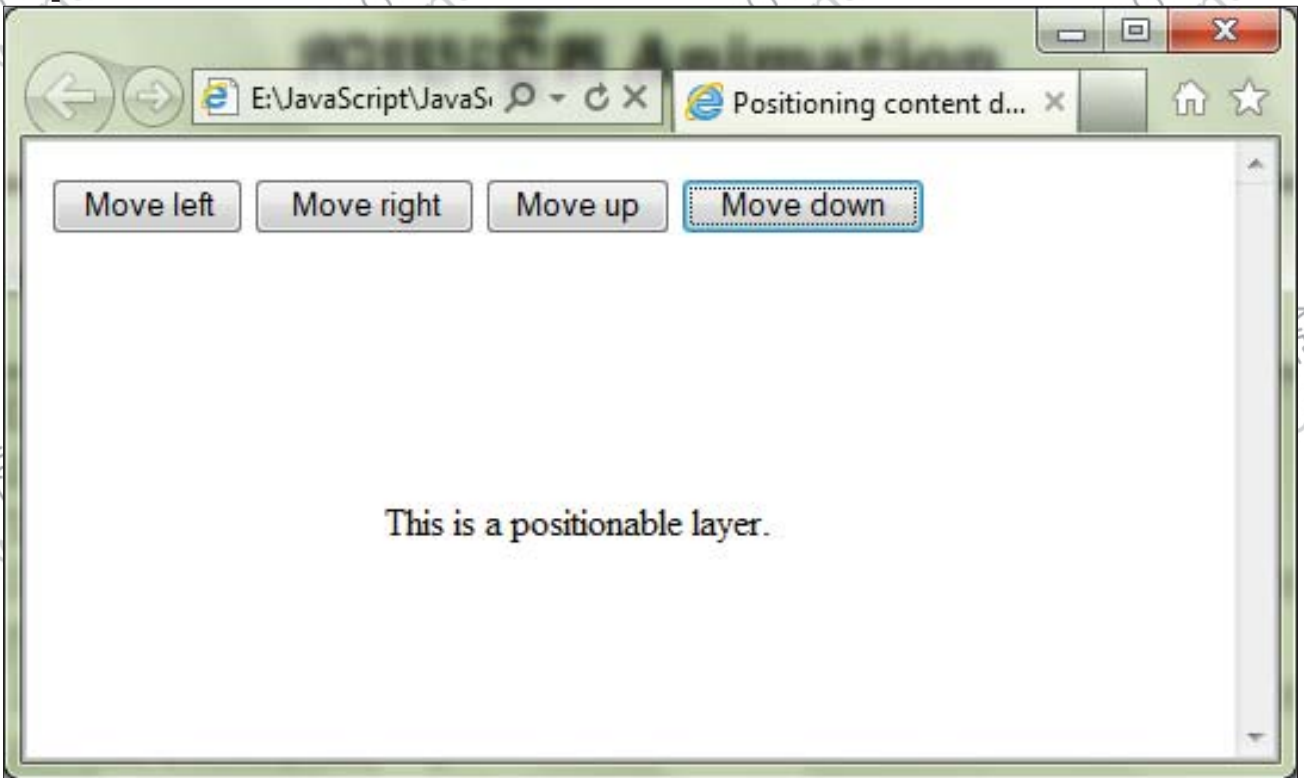
Ex:

```

<HTML>
<HEAD>
<TITLE>Positioning content dynamically</TITLE>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
function move(direction) {
var layerText = document.getElementById("myLayer");
switch(direction) {
case "left":
layerText.style.left = 50;
break;
case "right":
layerText.style.left = 150;
break;
case "up":
layerText.style.top = 50;
break;
case "down":
layerText.style.top = 150;
break;
}}
</SCRIPT></HEAD><BODY>
<DIV ID="myLayer" STYLE="position:absolute; left:100;
top:100;">
<P>This is a positionable layer.</P>
</DIV>
<FORM>
<INPUT TYPE="button" NAME="moveLayer" VALUE="Move left"
onClick="move('left');">
<INPUT TYPE="button" NAME="moveLayer" VALUE="Move
right" onClick="move('right');">
<INPUT TYPE="button" NAME="moveLayer" VALUE="Move up"
onClick="move('up');">
<INPUT TYPE="button" NAME="moveLayer" VALUE="Move down"
onClick="move('down');">
</FORM></BODY></HTML>

```

Output:



2. របៀបគ្រប់ Animation ជាមួយនឹង setInterval Method:

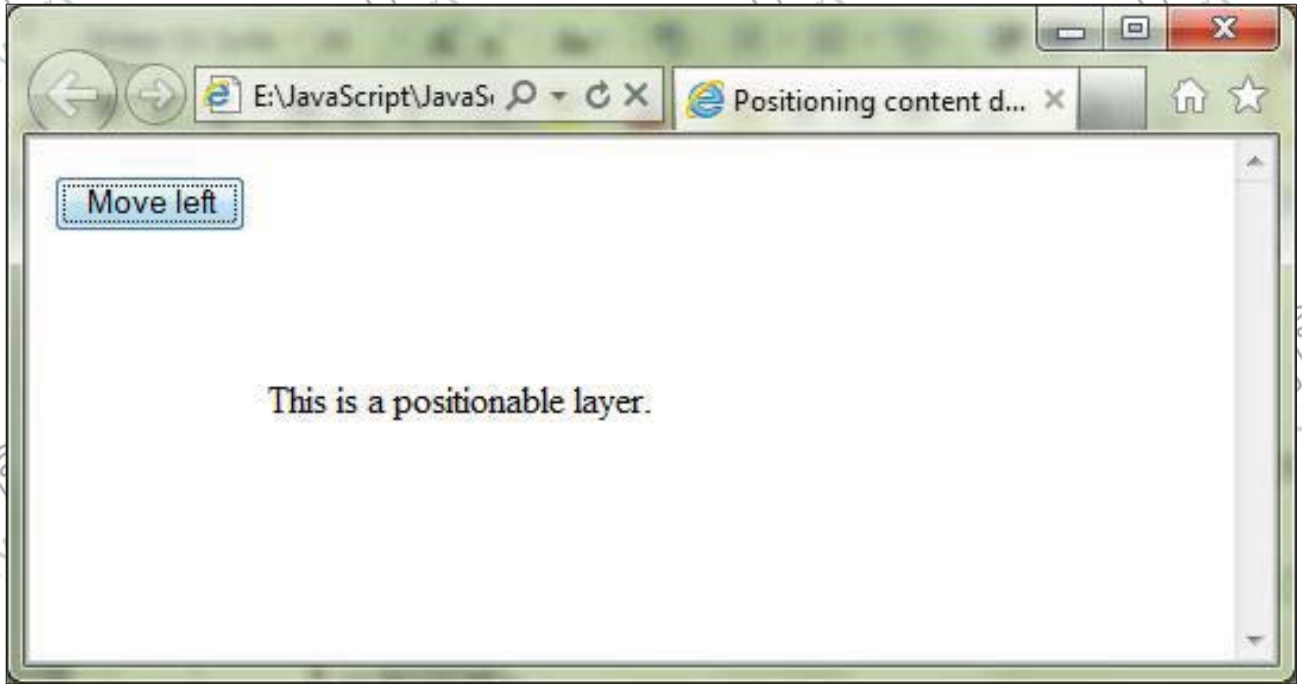
Ex:

```

<HTML>
<HEAD>
<TITLE>Positioning content dynamically</TITLE>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
var i=100;
function runDate(){
setInterval('move()',10);
}
function move() {
if (i<=800){
i =i+1;
var layerText = document.getElementById("myLayer");
layerText.style.left = i;
}}</SCRIPT></HEAD>
<BODY>
<DIV ID="myLayer" STYLE="position:absolute; left:100;
top:100;">
<P>This is a positionable layer.</P>
</DIV>
<FORM>
<INPUT TYPE="button" NAME="moveLayer" VALUE="Move left"
onClick="runDate();">
</FORM>
</BODY>
</HTML>

```

Output:



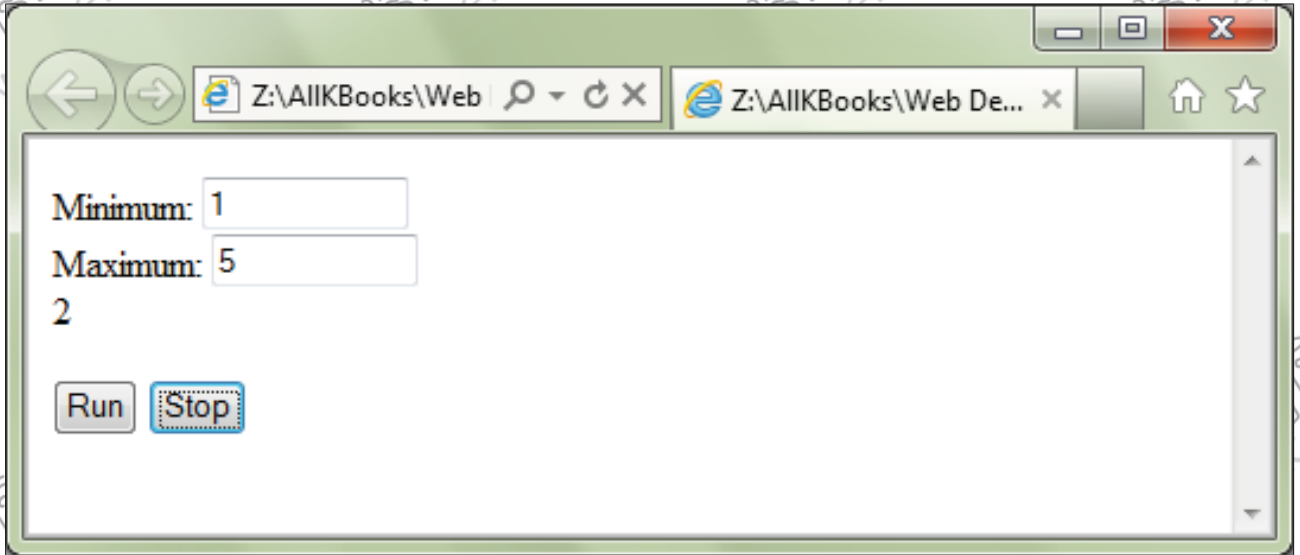
3. ប្រើប្រាស់បង្កើត Random Number:

Ex:

```

<HTML>
<BODY>
Minimum: <input type="text" size="10" id="mini"><br/>
Maximum: <input type="text" size="10" id="maxi"><br/>
<div id="abc">Run</div>
<FORM>
<INPUT TYPE="button" NAME="runNumber" VALUE="Run"
onClick="runTest();">
<INPUT TYPE="button" NAME="Clear" VALUE="Stop"
onClick="stopTest();">
</FORM>
<SCRIPT>
function runTest(){
a = setInterval(function(){runStart()},1);
}
function runStart(){
var x = document.getElementById("mini").value;
var y = document.getElementById("maxi").value;
x = parseInt(x);
y = parseInt(y)+1;
var z = parseInt(Math.random() *(y-x)+1);
document.getElementById("abc").innerHTML=z;
}
function stopTest(){
clearInterval(a);
}</SCRIPT></BODY></HTML>

```



4. របៀបបង្កើត Photo Slide Show:

Ex:
> Style.css

```

body {margin:0; background:#111; padding:5px; font:11px
Verdana,Arial}
#slideshow {list-style:none; color:#fff}
#slideshow span {display:none}
#wrapper {width:506px; margin:50px auto; display:none}
#wrapper * {margin:0; padding:0}
#fullsize {position:relative; width:500px;
height:300px; padding:2px; border:1px solid #ccc;
background:#000}
#information {position:absolute; bottom:0; width:500px;
height:0; background:#000; color:#fff; overflow:hidden;
z-index:200; opacity:.7; filter:alpha(opacity=70)}
#information h3 {padding:4px 8px 3px; font-size:14px}
#information p {padding:0 8px 8px}
#image {width:500px}
#image img {position:absolute; z-index:25; width:auto}
.imgnav {position:absolute; width:25%; height:306px;
cursor:pointer; z-index:150}
#imgprev {left:0; background:url(images/left.gif) left
center no-repeat}
#imgnext {right:0; background:url(images/right.gif)
right center no-repeat}
#imglink {position:absolute; height:306px; width:100%;
z-index:100; opacity:.4; filter:alpha(opacity=40)}
.linkhover {background:url(images/link.gif) center
center no-repeat}
#thumbnails {margin-top:15px}
#slideleft {float:left; width:20px; height:81px;
background:url(images/scroll-left.gif) center center
no-repeat; background-color:#222}

```

```
#slideleft:hover {background-color:#333}
#slideright {float:right; width:20px; height:81px;
background:#222 url(images/scroll-right.gif) center
center no-repeat}
#slideright:hover {background-color:#333}
#slidearea {float:left; position:relative; width:456px;
margin-left:5px; height:81px; overflow:hidden}
#slider {position:absolute; left:0; height:81px}
#slider img {cursor:pointer; border:1px solid #666;
padding:2px}
```

> index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>JavaScript Slideshow - TinySlideshow</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
  <ul id="slideshow">
    <li>
      <h3>TinySlideshow v1</h3>
      <span>photos/orange-fish.jpg</span>
      <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
      <a href="#"></a>
    </li>
    <li>
      <h3>Sea Turtle</h3>
      <span>photos/sea-turtle.jpg</span>
      <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
      
    </li>
  </ul>
</body>
</html>
```

```

</li>
<li>
  <h3>Red Coral</h3>
  <span>photos/red-coral.jpg</span>
  <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
  <a href="#"></a>
</li>
<li>
  <h3>Coral Reef</h3>
  <span>photos/coral-reef.jpg</span>
  <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
  <a href="#"></a>
</li>
<li>
  <h3>Blue Fish</h3>
  <span>photos/blue-fish.jpg</span>
  <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
  
</li>
<li>
  <h3>TinySlideshow v.2</h3>
  <span>photos/yellow-fish.jpg</span>
  <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
  <a href="#"></a>
</li>
<li>
  <h3>Squid</h3>
  <span>photos/squid.jpg</span>

```



```

        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
        <a href="#"></a>
    </li>
    <li>
        <h3>Small Fish</h3>
        <span>photos/small-fish.jpg</span>
        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Aliquam ut urna. Mauris nulla. Donec
nec mauris. Proin nulla dolor, bibendum et, dapibus in,
euismod ut, felis.</p>
        <a href="#"></a>
    </li>
</ul>
<div id="wrapper">
    <div id="fullsize">
        <div id="imgprev" class="imgnav"
title="Previous Image"></div>
        <div id="imglink"></div>
        <div id="imgnext" class="imgnav"
title="Next Image"></div>
        <div id="image"></div>
        <div id="information">
            <h3></h3>
            <p></p>
        </div>
    </div>
    <div id="thumbnails">
        <div id="slideleft" title="Slide
Left"></div>
        <div id="slidearea">
            <div id="slider"></div>
        </div>
        <div id="sliderright" title="Slide
Right"></div>
    </div>
</div>
<script type="text/javascript"
src="compressed.js"></script>
<script type="text/javascript">
    $('slideshow').style.display='none';
    $('wrapper').style.display='block';

```

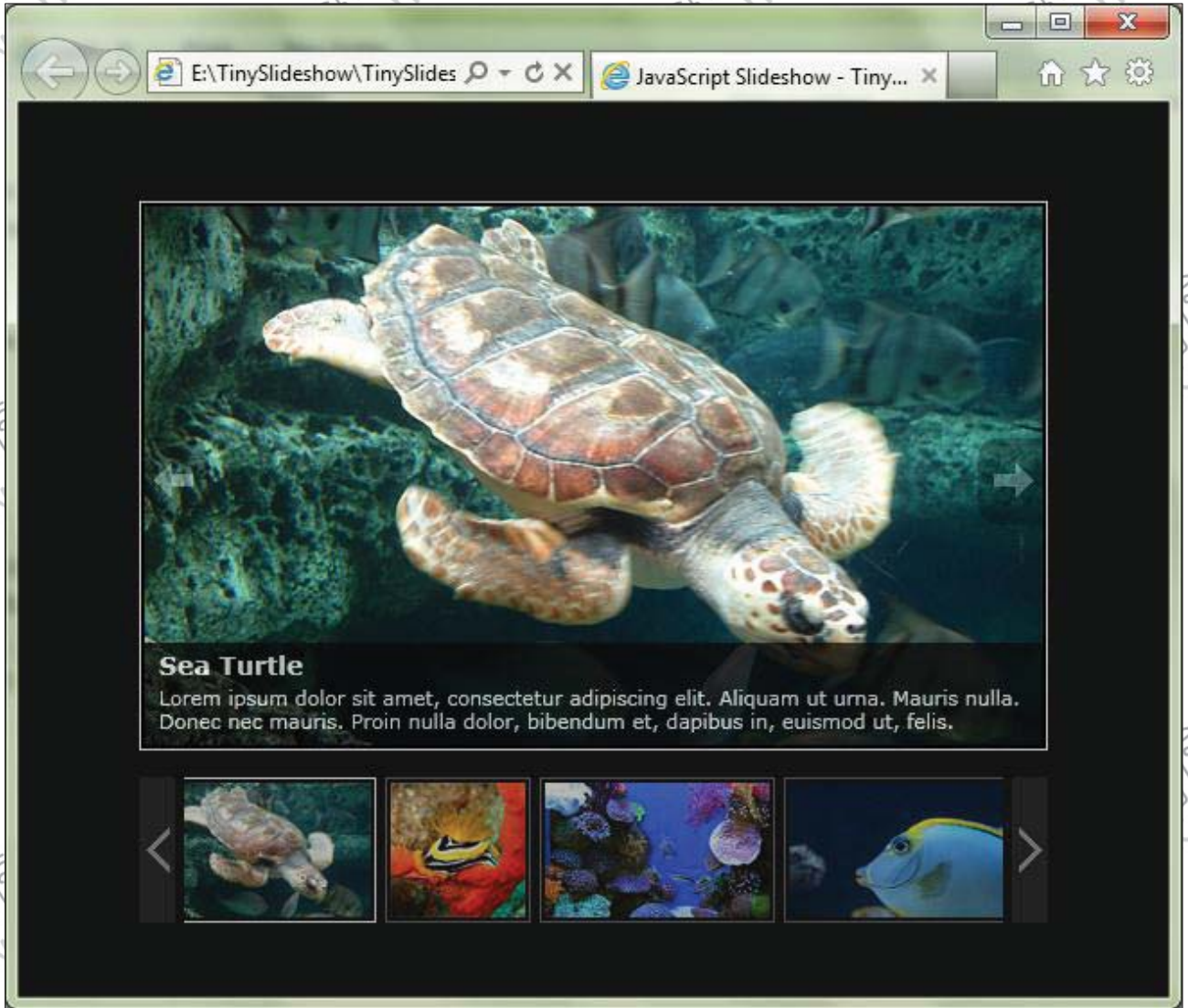
```

var slideshow=new TINY.slideshow("slideshow");
window.onload=function(){
  slideshow.auto=true;
  slideshow.speed=5;
  slideshow.link="linkhover";
  slideshow.info="information";
  slideshow.thumbs="slider";
  slideshow.left="slideleft";
  slideshow.right="sliderright";
  slideshow.scrollSpeed=4;
  slideshow.spacing=5;
  slideshow.active="#fff";

  slideshow.init("slideshow","image","imgprev","imgnext",
"imglink");
}
</script>
</body>
</html>

```

Output:



5. របៀបបង្កើត Dropdown Menu:

Ex:

> bluetabs.css

```
.bluetabs{
border-bottom: 1px solid gray;
}
.bluetabs ul{padding: 3px 0;margin-left: 0; margin-top:
1px;margin-bottom: 0;font: bold 12px Verdana;
list-style-type: none; text-align: left; /*set to left,
center, or right to align the menu as desired*/
}
.bluetabs li{display: inline;margin: 0;}
.bluetabs li a{text-decoration: none;padding: 3px 7px;
margin-right: 3px;border: 1px solid #778;
color: #2d2b2b;background: white url(media/bluetab.gif)
top left repeat-x; /*THEME CHANGE HERE*/ }

.bluetabs li a:visited{color: #2d2b2b;}

.bluetabs li a:hover{text-decoration: underline;
color: #2d2b2b;}

.bluetabs li.selected{
}

.bluetabs li.selected a{ /*selected main tab style */
background-image: url(media/bluetabactive.gif); /*THEME
CHANGE HERE*/border-bottom-color: white;}
```

```

.bluetabs li.selected a:hover{ /*selected main tab
style */
text-decoration: none;
}

/* ##### Style for Drop Down Menu ##### */

.dropmenudiv_b{
position:absolute;
top: 0;
border: 1px solid #918d8d; /*THEME CHANGE HERE*/
border-width: 0 1px;
font:normal 12px Verdana;
line-height:18px;
z-index:100;
background-color: white;
width: 200px;
visibility: hidden;
}
.dropmenudiv_b a{
width: auto;
display: block;
text-indent: 5px;
border: 0 solid #918d8d; /*THEME CHANGE HERE*/
border-bottom-width: 1px;
padding: 2px 0;
text-decoration: none;
color: black;
}

* html .dropmenudiv_b a{ /*IE only hack*/
width: 100%;
}

.dropmenudiv_b a:hover{ /*THEME CHANGE HERE*/
background-color: #edf3f3;
}

```

> dropdowntabsdemo.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en"
xml:lang="en">
<head>
<script type="text/javascript"
src="dropdowntabfiles/dropdowntabs.js">
</script>
<!-- CSS for Drop Down Tabs Menu -->
<link rel="stylesheet" type="text/css"
href="dropdowntabfiles/bluetabs.css" />
<body>

<h2><!--1st drop down menu --><!--2nd drop down menu --
>
  <script type="text/javascript">
//SYNTAX: tabdropdown.init("menu_id", [integer OR
"auto"])
tabdropdown.init("colortab", 3)
  </script>
</h2>
<h2>JavaScript Dropdown Menu</h2>

<div id="bluemenu" class="bluetabs">
<ul>
<li><a href="http://www.dynamicdrive.com">Home</a></li>
<li><a href="http://www.dynamicdrive.com/style/"
rel="dropmenu1_b">CSS</a></li>
<li><a href="http://www.dynamicdrive.com/resources/"
rel="dropmenu2_b">Partners</a></li>
<li><a
href="http://tools.dynamicdrive.com">Tools</a></li>
</ul>
</div>

<!--1st drop down menu -->
<div id="dropmenu1_b" class="dropmenudiv_b">
<a
href="http://www.dynamicdrive.com/style/csslibrary/cate
gory/C1/">Horizontal CSS Menus</a>
<a
href="http://www.dynamicdrive.com/style/csslibrary/cate
gory/C2/">Vertical CSS Menus</a>

```

```

<a
href="http://www.dynamicdrive.com/style/csslibrary/cate
gory/C4/">Image CSS</a>
<a
href="http://www.dynamicdrive.com/style/csslibrary/cate
gory/C6/">Form CSS</a>
<a
href="http://www.dynamicdrive.com/style/csslibrary/cate
gory/C5/">DIVs and containers</a>
<a
href="http://www.dynamicdrive.com/style/csslibrary/cate
gory/C7/">Links and Buttons</a>
<a
href="http://www.dynamicdrive.com/style/csslibrary/cate
gory/C8/">Other</a>
</div>

<!--2nd drop down menu -->
<div id="dropmenu2_b" class="dropmenudiv_b"
style="width: 150px;">
<a href="http://www.cssdrive.com">CSS Drive</a>
<a href="http://www.javascriptkit.com">JavaScript
Kit</a>
<a href="http://www.codingforums.com">Coding Forums</a>
<a
href="http://www.javascriptkit.com/jsref/">JavaScript
Reference</a>
</div>

<script type="text/javascript">
//SYNTAX: tabdropdown.init("menu_id", [integer OR
"auto"])
tabdropdown.init("bluemenu")
</script>
<h2>Enter Information Technology Center</h2>

<p>&nbsp;</p>
<br style="clear: left;" />
<br class="IEonlybr" />
<!--1st drop down menu --><!--2nd drop down menu --
><script type="text/javascript">
//SYNTAX: tabdropdown.init("menu_id", [integer OR
"auto"])
tabdropdown.init("slidemenu")
</script>

```

```

<h2>&nbsp;</h2>
<br style="clear: left;" />
<br class="IEonlybr" />

<!--1st drop down menu --><!--2nd drop down menu --
><script type="text/javascript">
//SYNTAX: tabdropdown.init("menu_id", [integer OR
"auto"])
tabdropdown.init("glowmenu", "auto")
</script>

<h2>&nbsp;</h2>
<br style="clear: left;" />

<!--1st drop down menu --><!--2nd drop down menu --
><script type="text/javascript">
//SYNTAX: tabdropdown.init("menu_id", [integer OR
"auto"])
tabdropdown.init("moonmenu", 3)
</script>
</body>
</html>

```


Output:



មេរៀនទី 12: របៀបដាក់ Multimedia

1. និយមន័យ:

នៅលើ Webpage យើងអាចធ្វើការដាក់នូវ Multimedia មួយចំនួនដូចជា sound, music, videos និង Animations ផ្សេងៗទៀត។ ក្នុងនោះ Webpage អាច Support ជាមួយនឹង Video File ជាច្រើនដូចជា: *.avi, *.wmv, *.mpeg, *.mov, *.flv, *.mp4,... ចំនែក Sound វិញវា Support ជាមួយនឹង *.midi, *.wma, *.mp3,... ។ ចំនែក wav និង mp3 គឺពេញនិយមសំរាប់ Sound Format ចំនែក flv គឺពេញនិយមសំរាប់ Video File Format ។

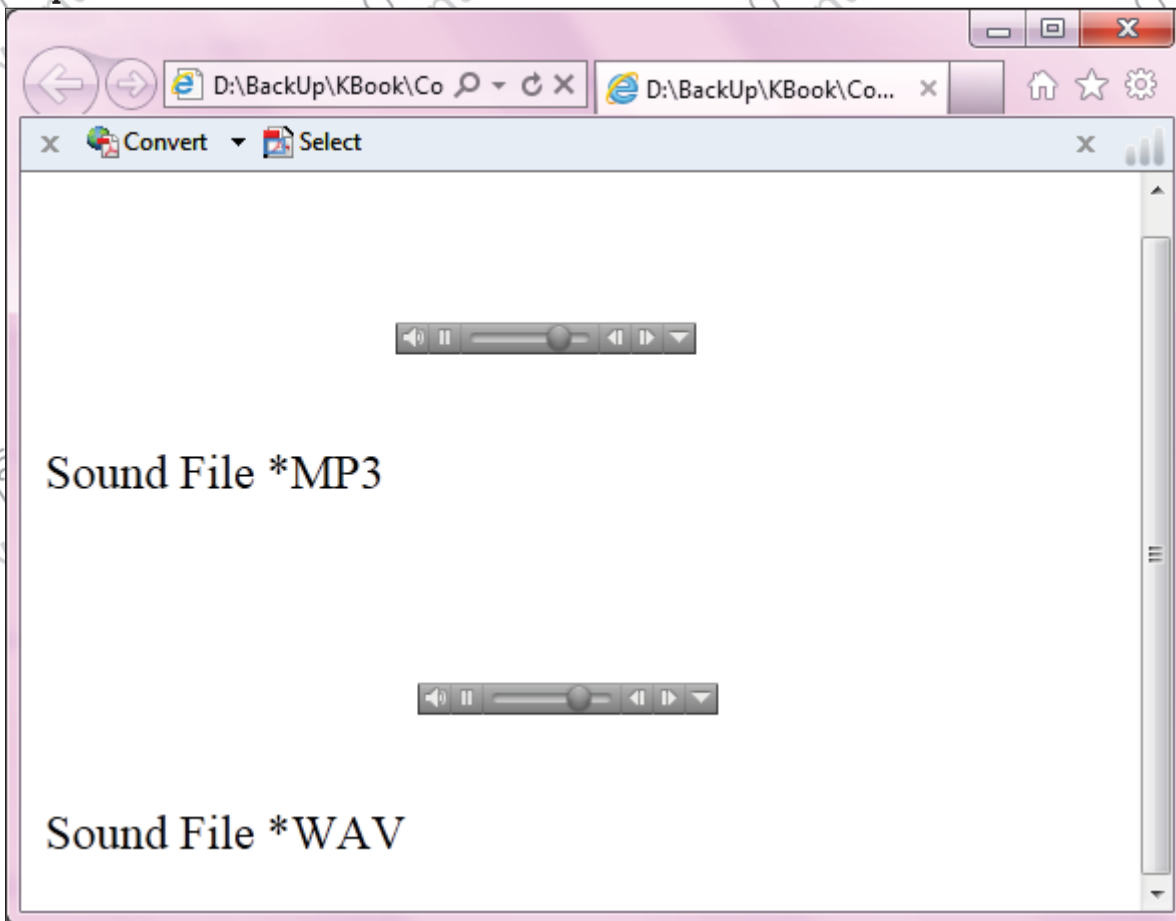
2. របៀបដាក់ Sound ចូល Webpage:

ដើម្បីដាក់ Sound ចូលទៅក្នុង Webpage នោះយើងអាចប្រើប្រាស់ Tag មួយឈ្មោះ <embed> ដូចខាងក្រោម: Tag ដែលត្រូវប្រើប្រាស់គឺ <embed src=?> ដែលក្នុងនោះ ? គឺជាឈ្មោះរបស់ Sound File ដែលត្រូវដាក់។

Ex:

```
<!DOCTYPE html>
<html>
<body>
<p>Sound File *MP3
<embed height="100" width="100"
src="sound\\Wildlife.mp3"></embed>
</p>
<p>Sound File *WAV
<embed height="100" width="100"
src="sound\\Wildlife.wav"></embed>
</p>
</body>
</html>
```

Output:



3. ប្រើប្រាស់កំណត់ Option បន្ថែម២ Sound:

យើងអាចធ្វើការបន្ថែមជាមួយនឹង Attribute ដូចជា autostart និង loop ដើម្បីកំណត់ Option បន្ថែមទៀតទៅឲ្យ ដែលក្នុងនោះ:

> autostart = "?1" សំរាប់កំណត់ Play Auto នៅពេល Webpage ដែល ?1 អាចជា true ឬ false

> loop = "?2" សំរាប់កំណត់ Replay ឡើងវិញដែល ក្នុងនោះ ?2 អាចជា true ឬ false

Ex:

```

<!DOCTYPE html>
<html>
<body>

<p>Sound File *MP3
<embed height="100" width="100"
src="sound\Wildlife.mp3" autostart=false
loop=true</embed>
</p>

</body>
</html>

```

4. របៀបដាក់ Video ចូលក្នុង Webpage:

Ex:

```

<!DOCTYPE html>
<html>
<body>
<p>Video File *.3GP
<embed height="100" width="100"
src="video\\Wildlife.3gp"></embed>
</p>
<p>Video File *.AVI
<embed height="100" width="100"
src="video\\Wildlife.avi"></embed>
</p>
<p>Video File *.MP4
<embed height="100" width="100"
src="video\\Wildlife.mp4"></embed>
</p>
<p>Video File *.SWF
<embed height="100" width="100"
src="video\\Wildlife.swf"></embed>
</p>
<p>Video File *.WMV
<embed height="100" width="100"
src="video\\Wildlife.wmv"></embed>
</p></body>
</html>

```