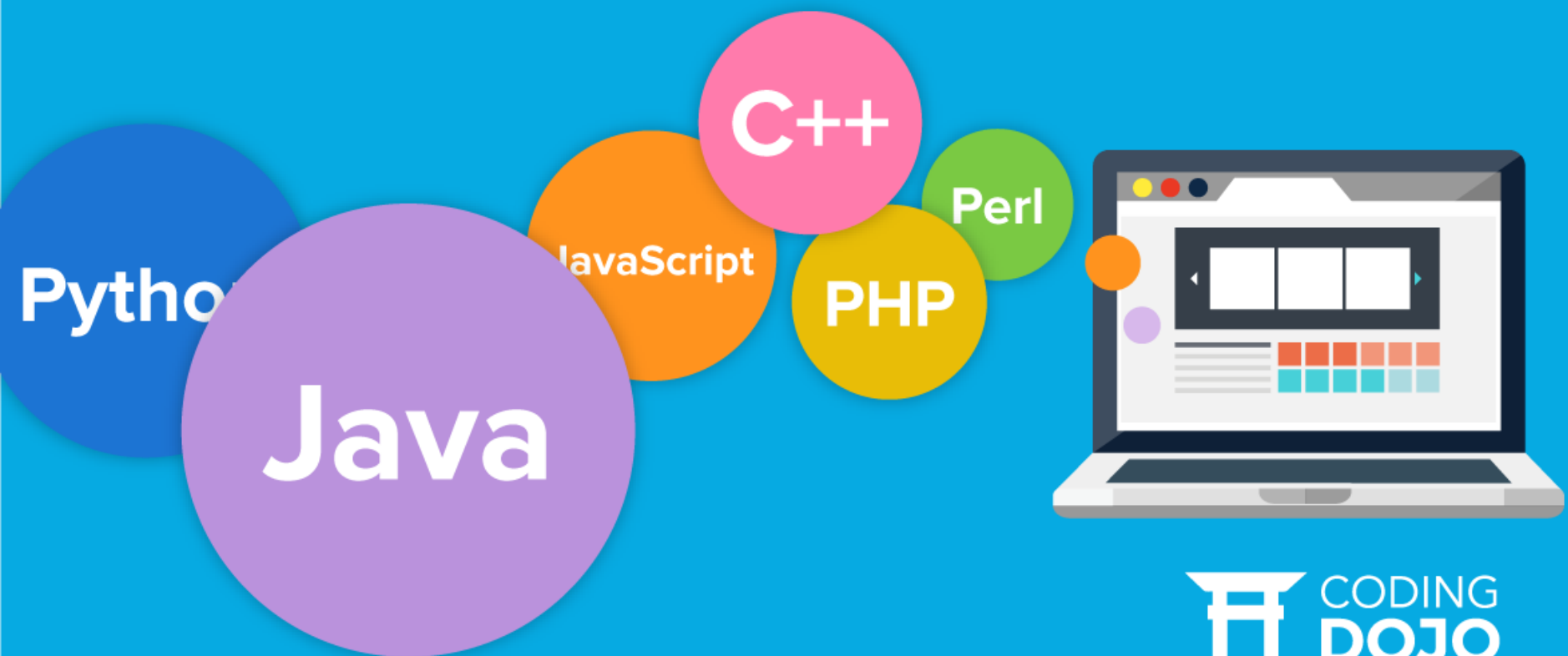


Ngôn ngữ JavaScript





Tổng quan Internet & Web





Giới thiệu JavaScript



Ngôn ngữ JavaScript



£ Client-Side Script

- ❑ Script được thực thi tại *Client-Side* (trình duyệt): Thực hiện các tương tác với người dùng (tạo menu chuyển động, ...) , kiểm tra dữ liệu nhập,...

£ Server-Side Script

- ❑ Script được xử lý tại *Server-Side*, nhằm tạo các trang web có khả năng phát sinh nội dung động. Một số xử lý chính: kết nối CSDL, truy cập hệ thống file trên server, phát sinh nội dung html trả về người dùng...



- £ **Client-side JavaScript** Là ngôn ngữ hoạt động trên trình duyệt của người dùng (*client*)
- £ **Chia sẻ xử lý** trong ứng dụng web. Giảm các xử lý không cần thiết trên server.
- £ Giúp **tạo các hiệu ứng, tương tác cho** trang web.



- £ Khi trình duyệt (*Client browser*) truy cập trang web có chứa các đoạn mã xử lý tại server-side. Server (run-time engine) sẽ thực hiện các lệnh *Server-side Scripts* và trả về nội dung HTML cho trình duyệt.
- £ Nội dung HTML trả về chủ yếu bao gồm: *mã html, client-script*.





Nhúng JavaScript vào trang HTML



Ngôn ngữ JavaScript



1. Định nghĩa Script trực tiếp trong trang HTML:

```
<script type="text/javascript">  
  <!--  
    // Lệnh Javascript  
  -->  
</script>
```

2. Nhúng sử dụng script cài đặt từ 1 file .js khác:

```
<script src="abc.js"></script>
```

Ghi chú: Không chèn lệnh javascript vào giữa thẻ script trong trường hợp liên kết đến file js



- £ Web Browser sẽ thực thi **<script>** khi load trang web theo thứ tự từ *trên xuống* dưới.
- £ Đối với *Source code JavaScript* có thể đặt trong các file *.js* sẽ được *nhúng* vào file HTML trước khi hoạt động.
- £ Các đoạn *code JavaScript* được Browser xử lý *cùng thứ tự* với các thẻ *HTML*. Trừ các *hàm* (*function*) chỉ được thực thi khi *có lời gọi hàm*.

```
<html>
  <head>
    <script type="text/javascript">
      some statements
    </script>
  </head>
  <body>
    <script type="text/javascript">
      some statements
    </script>

    <script src="Tên_file_script.js">method()</script>

    <script type="text/javascript">
      // gọi thực hiện các phương thức được định nghĩa
      // trong "Tên_file_script.js"
    </script>
  </body>
</html>
```

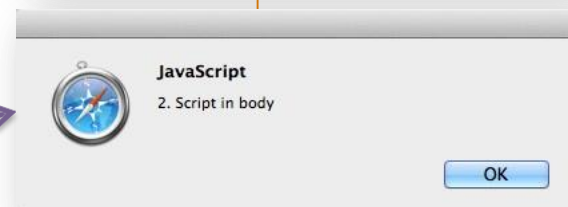
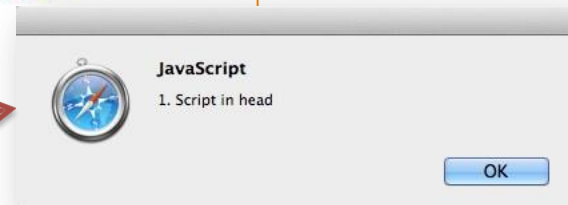
- £ Đặt giữa tag `<head>` và `</head>`: script sẽ thực thi ngay khi trang web được mở.
- £ Đặt giữa tag `<body>` và `</body>`: script trong phần body được thực thi khi trang web đang mở (sau khi thực thi các đoạn script có trong phần `<head>`).
- £ Số lượng đoạn client-script chèn vào trang không hạn chế.



```
<!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=UTF-8" />
    <title>Title of website</title>
    <script type="text/javascript">
      alert("1. Script in head");
    </script>
  </head>
  <body>
    <script type="text/javascript">
      alert("2. Script in body");
    </script>

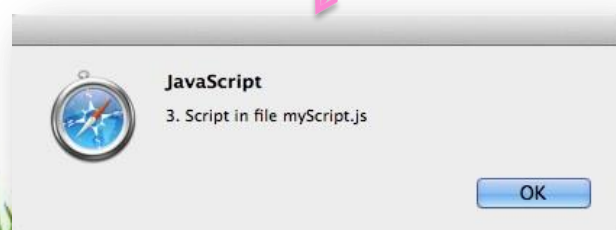
    <script type="text/javascript" src="myScript.js" ></script>

    <script type="text/javascript">
      showMessage();
    </script>
  </body>
</html>
```



File myScript.js

```
function showMessage()
{
  alert("3. Script in file myScript.js");
}
```





Kiểu dữ liệu và cú pháp

Ngôn ngữ JavaScript

Ngôn ngữ JavaScript



£ Cách đặt tên biến

- ❑ Bắt đầu bằng một chữ cái hoặc dấu _ hoặc \$
- ❑ A..Z,a..z,0..9,_ : **phân biệt HOA, Thường**

£ Khai báo biến

- ❑ Sử dụng từ khóa **var**

Ví dụ: **var count = 10, amount;**

- ❑ *Không cần khai báo biến trước khi sử dụng*, biến thật sự tồn tại khi bắt đầu sử dụng lần đầu tiên
- ❑ Biến *không cần khai báo kiểu dữ* liệu vì biến trong javascript *không có kiểu dữ liệu nhất định*



£ Phạm vi tồn tại của biến

- ❑ Khai báo với từ khoá var trong hàm => phạm vi cục bộ trong hàm (local)
- ❑ Không khai báo với từ khoá var hoặc khai báo ngoài hàm => phạm vi toàn cục (global)

```
var globalVar = "Global Variable";  
  
function foo() {  
    var localVar = "Local Variable";  
    newGlobalVar = "New Global Variable";  
    ...  
}
```


Kiểu dữ liệu	Ví dụ	Mô tả
Object	<pre>var myObject = {} ; myObject.attribute = "hello"; var myObject = new Object(); var myArray = []; myArray[0] = 1; var myArray = new Array();</pre>	Khởi tạo đối tượng bằng từ khoá new
String	<pre>"The cow jumped over the moon." "40"</pre>	Chứa được chuỗi unicode Chuỗi rỗng ""
Number	<pre>0.066218 12</pre>	Theo chuẩn IEEE 754
boolean	true / false	
undefined	var myVariable ;	myVariable = undefined
null	connection.Close();	connection = null

Một biến trong JavaScript có thể lưu bất kỳ kiểu nào

- £ Biến tự đổi kiểu dữ liệu khi giá trị mà nó lưu trữ thay đổi

Ví dụ:

```
var x = 10;           // x kiểu Number  
x = "hello world !"; // x kiểu String
```

- £ Có thể cộng 2 biến khác kiểu dữ liệu

Ví dụ:

```
var x;  
x = "12" + 34.5;      // KQ: x = "1234.5"
```

- £ Hàm **parseInt(...)**, **parseFloat(...)** : Đổi KDL từ chuỗi sang số.
- £ Hàm **toString()**: Đổi KDL từ số sang chuỗi

- £ **isNaN(...)** : kiểm tra dữ liệu không là số (Not a Number)
- £ **typeof ...** : trả về kiểu dữ liệu của biến “number, string, boolean, object, undefined, null”

```
var myNumberType = 45;  
var myNumberType = myNumberType.toString();  
// confirm the type is indeed string but the content is still a number  
console.log(typeof myNumberType); // string  
console.log(isNaN(myNumberType)); // false
```

```
var myArray = new Array();  
// or  
var myArray = [];  
    myArray[0] = "Andrew";  
    myArray[1] = "Monica";  
    myArray[2] = "Catie";  
    myArray[3] = "Jenna";  
    myArray[4] = "Christopher";  
  
// you can achieve the same result doing this:  
  
var myArray = ["Andrew", "Monica", "Catie", "Jenna", "Christopher"];
```

```
// Example 1
var myFirstObject = {};
myFirstObject.firstName = "Andrew";
myFirstObject.lastName = "Grant";
console.log(myFirstObject.firstName);
// Example 2
var mySecondObject = {
  firstName: "Andrew",
  lastName: "Grant"
};
console.log(mySecondObject.firstName);
// Example 3
var myThirdObject = new Object();
myThirdObject.firstName = "Andrew";
myThirdObject.lastName = "Grant";
console.log(myThirdObject.firstName);
```

- £ JavaScript Object Notation
- £ Định dạng trao đổi dữ liệu độc lập, tương tự XML
- £ Cấu trúc dựa trên các cặp Key/Value
- £ Có thể được biểu diễn và lưu trữ trong JavaScript dưới dạng Object



```
var jsonObject = {  
  "firstName": "John",  
  "lastName": "Smith",  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": 10021  
  },  
  "phoneNumbers": [  
    "212 555-1234",  
    "646 555-4567"  
  ]  
};
```

Toán tử	Mô tả
++, --	
+, -, *, /, %	
<, <=, >, >=	
==, !=	So sánh giá trị
===, !==	So sánh giá trị và kiểu dữ liệu
&&,	AND và OR
=	Gán
+	Nối chuỗi



- £ Khối lệnh được bao trong dấu {}
- £ Mỗi lệnh nên kết thúc bằng dấu ;
- £ Cách ghi chú thích:
 - // *Chú thích 1 dòng*
 - /* *Chú thích
nhiều dòng */*




```
if (condition)  
{  
    statement[s] if true  
}  
else  
{  
    statement[s] if false  
}
```

Ví dụ:

```
var x = 5, y = 6, z;  
if (x == 5)  
{  
    if (y == 6)  
        z = 17;  
}  
else  
    z = 20;
```

```
switch (expression)  
{  
    case label1 :  
        statementlist  
    case label2 :  
        statementlist  
    ...  
    default :  
        statement list  
}
```

Ví dụ :

```
var diem = "G";  
switch (diem) {  
    case "Y":  
        document.write("Yếu");  
        break;  
    case "TB":  
        document.write("Trung bình");  
        break;  
    case "K":  
        document.write("Khá");  
        // break;  
    case "G":  
        document.write("Giỏi");  
        break;  
    default:  
        document.write("Xuất sắc");
```

```
}
```

```
for ([initial expression]; [condition]; [update expression])  
{  
    statement[s] inside loop  
}
```

Ví dụ:

```
var myarray = new Array();  
for (i = 0; i < 10; i++)  
{  
    myarray[i] = i;  
}
```

while (*expression*)

```
{  
    statements  
}
```

Ví dụ:

```
var i = 9, total = 0;  
while (i < 10)  
{  
    total += i * 3 + 5;  
    i = i + 5;  
}
```

do

```
{  
    statements  
} while (expression);
```

Ví dụ:

```
var i = 9, total = 0;  
do  
{  
    total += i * 3 + 5;  
    i = i + 5;  
} while (i < 10);
```

£ Dạng thức khai báo chung:

```
function Tên_hàm(thamso1, thamso2, ...)  
{  
    ...  
}
```

£ Hàm có giá trị trả về:

```
function Tên_hàm(thamso1, thamso2, ...)  
{  
    ...  
    return (value);  
}
```



£ Hàm không định nghĩa tham số:

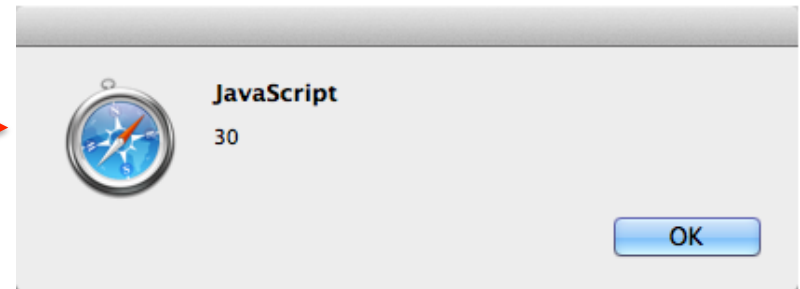
```
function Tên_hàm()  
{  
    ...  
    for (var i=0; i< arguments.length; i++) {  
        // do sth with arguments[i];  
    }  
}
```

£ Ví dụ:

```
var Sum = function (x, y){  
    tong = x + y;  
    return tong;  
}
```

£ Gọi hàm:

```
var x = Sum(10, 20);  
alert(x);
```



```
var myCleverObject = {  
  firstName: "Andrew",  
  age: 21,  
  myInfo: function () {  
    console.log("My name is " + this.firstName + ".");  
    console.log("My age is " + this.age + ".");  
  }  
};  
myCleverObject.myInfo();|
```



```
var actionsToTakeWhenServerHasResponded = function () {  
    console.log('The server just responded!')  
}  
function communicateWithServer(callback) {  
    callback();  
}  
communicateWithServer(actionsToTakeWhenServerHasResponded);
```



```
function person(first, last) {  
    this.firstName = first;  
    this.lastName = last;  
    this.fullname = function() {  
        return this.firstName + " " + this.lastName;  
    };  
}  
person.prototype.nationality = "English";  
  
var myFather = new person("John", "Doe");  
console.log(myFather.fullname());
```



Xử lý sự kiện với JavaScript



Ngôn ngữ JavaScript



£ Các sự kiện được hỗ trợ bởi hầu hết các đối tượng

- ☐ onClick
- ☐ onFocus
- ☐ onChange
- ☐ onBlur
- ☐ onMouseOver
- ☐ onMouseOut
- ☐ onMouseDown
- ☐ onMouseUp
- ☐ onLoad
- ☐ onSubmit
- ☐ onResize
- ☐ ...

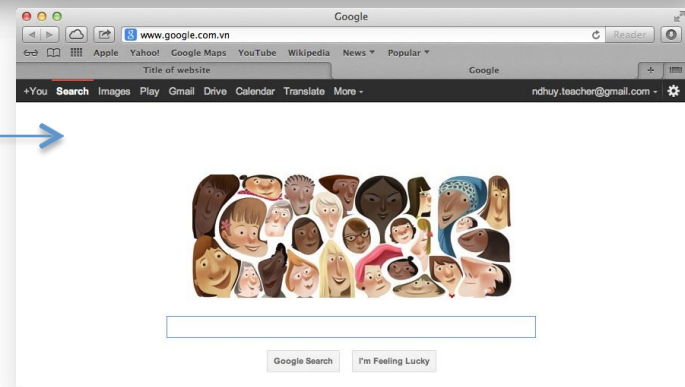
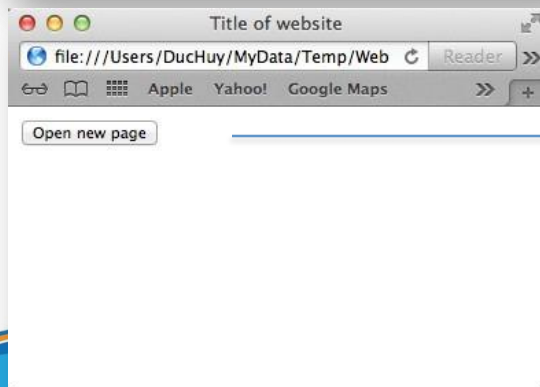


£ Cú pháp:

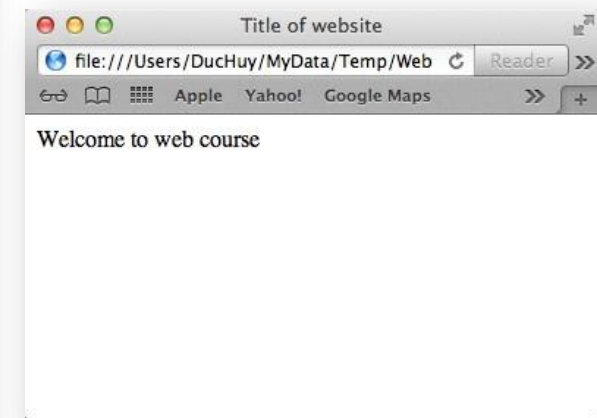
<TAG eventHandler = "JavaScript Code">

£ Lưu ý: Dấu “...” và ‘...’

```
<body>
  <input type="button"
        name="btnClickMe"
        value="Open new page"
        onclick="window.open('http://www.google.com');" />
</body>
```



```
<!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=UTF-8" />  
    <title>Title of website</title>  
    <script type="text/javascript">  
      function ShowMessage() {  
        alert("hello world");  
      }  
    </script>  
  </head>  
  <body onload="ShowMessage()">  
    Welcome to web course  
  </body>  
</html>
```



- £ Gán tên hàm xử lý cho 1 object event
object.eventhandler = function_name;

```
<html>
  <head>
    <script language="Javascript">
      function GreetingMessage()
      {
        window.alert("Welcome to my world");
      }
      window.onload = GreetingMessage();
    </script>
  </head>
  <body>
  </body>
</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=UTF-8" />
    <title>Title of website</title>
    <script type="text/javascript">

```

```

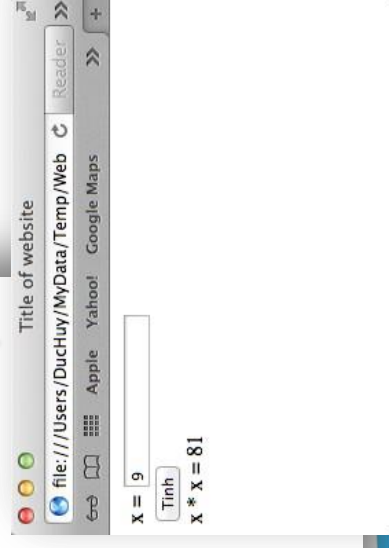
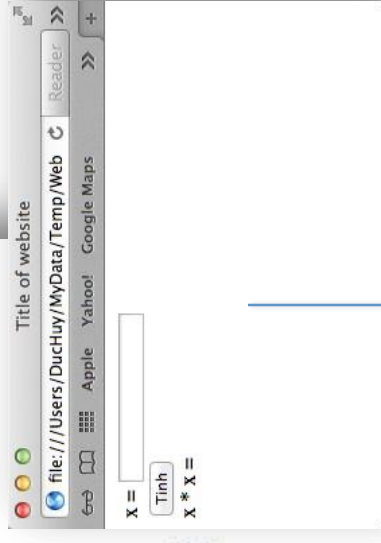
      function TinhToan() {
        var x = frmMain.txtGiaTri.value;
        ketQua.innerHTML = x * x;
      }
    </script>
  </head>

```

```

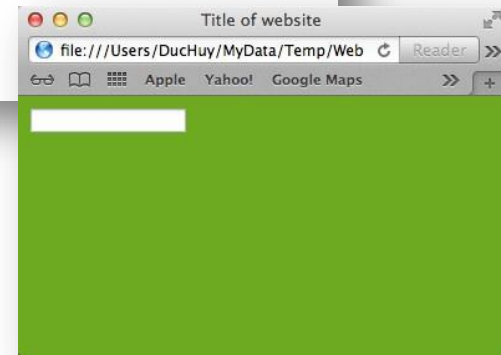
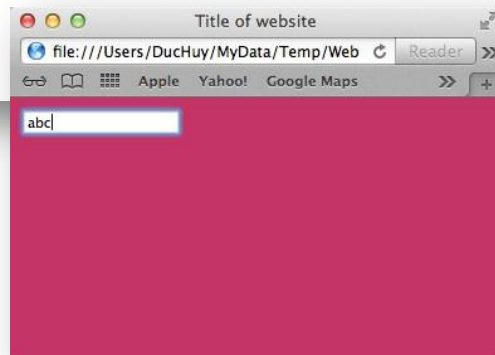
  <body>
    <form name="frmMain">
      x = <input type="text" name="txtGiaTri" />
      <br />
      <input type="button" value="Tinh"
        onclick="TinhToan()" />
      <br />
      x * x = <span id="ketQua"></span>
    </form>
  </body>
</html>

```



- £ Xảy ra khi một thành phần HTML bị focus (**onFocus**) và mất focus (**onBlur**)

```
<body>  
  <form name="frmMain">  
    <input type="text"  
      onfocus="document.bgColor='#cc3366';"  
      onBlur="document.bgColor='#66aa33';"  
    />  
  </form>  
</body>
```



- £ Nhúng JavaScript vào tài liệu HTML
- £ Cú pháp
- £ Xử lý sự kiện

