

# CHƯƠNG XVII


## MÔ HÌNH ĐỐI TƯỢNG

### DOM

---



# Object

Object	Properties	Methods
	<ul style="list-style-type: none"><li>• xe tên vespa</li><li>• Hãng Piaggio</li><li>• Xe màu đỏ</li><li>• Xe sản xuất năm 1954</li><li>• Xe có cân nặng 110kg</li></ul>	<ul style="list-style-type: none"><li>• Khởi động()</li><li>• Chạy xe()</li><li>• Dừng xe()</li><li>• Thắt xe lại()</li></ul>

# Đối tượng Xe Vespa

---

```
<xemay ten=vespa mau=đỏ hang=piaggio  
namsx=1980 kannang=130kg>
```

```
<xemay ten=SH mau=trắng hang=Honda  
namsx=2010 kannang=120kg>
```

Cách truy cập các thuộc tính của xe:

```
vespa.hang="piaggio";
```

```
vespa.namsx="1980";
```

```
vespa.kannang="130kg";
```

```
vespa.mau="đỏ";
```

# I. MÔ HÌNH DOM (Document Object Model)

Đối tượng – Mô hình đối tượng:

❑ **object** :

---

- properties (thuộc tính),
- methods (phương thức)
- events (sự kiện).

❑ Object **[id]**: Định danh cho objects *duy nhất*

❑ Ví dụ: `<img id="h1" ...>`

`<table id="tb1" ...>`

`<tr id="ro1" ...>`

`<td id="c11" ...>`

`<form id="frmDK" ...>`

`<input type=text id="txtName">`

❑ Toán tử dấu chấm (.) để phân cấp giữa các đối tượng và truy cập thuộc tính của mỗi đối tượng

---

❑ Mỗi đối tượng đều có *properties (thuộc tính)*, *events (sự kiện)* và *methods (phương thức)*, nhờ các thành phần này mà có thể truy cập và thay đổi nội dung của chúng.

➤ Properties: mô tả thông tin của đối tượng. (How?)

```

```

```
[idh1.height, idh1.width, idh1.src]
```

# JavaScript Events

---

- `onClick`
- `onChange`
- `onFocus`
- `onBlur`
- `onMouseOver`
- `onMouseOut`
- `onLoad`
- `onSubmit`
- `onMouseDown`
- `onMouseUp`

# Một số đối tượng

---

## ❏ **Array () :**

- lưu trữ nhiều giá trị với cùng một tên gọi.
- chứa các thành phần mang kiểu giá trị **khác nhau**
- chỉ số từ 0 đến n-1.
- Khởi tạo một mảng:

Dùng từ khóa **new** để khởi tạo một mảng

```
var   variable_arr = new Array() ;
```

- **.length**: số phần tử của Array

Ví dụ:

```
<script>
```

```
var arr= new Array() ;
```

---

```
arr[0]= "thu hai";
```

```
arr[1]= "Thu ba";
```

```
arr[2]= "Thu tu";
```

```
arr[3]= "Thu nam";
```

```
arr[4]= "Thu sau";
```

```
arr[5]= "Thu bay";
```

```
for(i=0; i<=5; i++)
```

```
document.write(arr[i]+ "<br>");
```

```
document.write(arr.length+ "<br>");//6
```

```
</script>
```



```
<script language="JavaScript" type="text/JavaScript">
function changecolor()
{
  var arrcolor= new Array();
  arrcolor[0]="#ff99ff";
  arrcolor[1]="#ffff99";
  arrcolor[2]="#99ffff";
  arrcolor[3]="#ff6633";
  arrcolor[4]="#990033";
  var m= Math.round(Math.random()*5);
  document.bgColor=arrcolor[m];
  idq=setTimeout("changecolor()",1000);
}
function stop()
{
  clearTimeout(idq);
}
</script>
```

## ➤ Các phương thức của đối tượng Array()

Phương thức	Ý nghĩa	Ví dụ
concat()	Dùng để nối 2 mảng	a=a.concat(b)
join(separator)	để ghép các phần tử trong mảng lại với nhau cách nhau bởi dấu separator	a=a.join("+")
slice(start,end)	Dùng tách một mảng bắt đầu từ vtrí start đến vtrí end-1.	str=a.slice(i,j)
reverse()	Dùng để đảo ngược chuỗi	a.reverse()
valueOf()	Dùng để lấy tất cả các đối tượng trong chuỗi	a.valueOf()
pop()	Lấy phần tử cuối của mảng	
push()	Thêm 1 hoặc nhiều phần tử vào cuối mảng	
Shift()	lấy phần tử và trả về phần tử đầu tiên của mảng	
Sort()	sắp xếp các phần tử của mảng	
valueOf()	Trả về tất cả các giá trị ban đầu của mảng	

## Ví dụ 2

```
<script type="text/javascript">
var arrName = new Array(3)
arrName [0] = "Jani"
arrName [1] = "Tove"
arrName [2] = "Hege"
document.write(arrName.length + "<br>")
document.write(arrName.join(".") + "<br>")
document.write(arrName.reverse() + "<br>")
document.write(arrName.sort() + "<br>")
document.write(arrName.push("Ola", "Jon") + "<br>")
document.write(arrName.pop() + "<br>")
document.write(arrName.shift() + "<br>")
</script>
```

## □ Date ()

Cung cấp thông tin về ngày, giờ trên môi trường client.

Thiết lập ngày tháng năm và giờ hiện hành trên trang web.

---

### Cách khai báo:

#### Cách 1: Khai báo và khởi tạo

```
var variableName= new Date ("month, day,  
    year , hours : minutes : seconds");
```

hoặc:

```
var variableName= new  
    Date (year, month, day, hours, minutes, secon  
d) ;
```

#### Cách 2: Khai báo ngày hiện hành ( Không khởi tạo)

```
var variableName=new Date () ;
```

➤ Các phương thức của đối tượng **Date ()** :

Phương thức	Mô tả
Date()	trả về đối tượng date
getDate()	Trả về giá trị ngày ( số nguyên từ 1-31) trong tháng
getDay()	Trả về giá trị ngày trong tuần ( số nguyên từ 0-7 Sunday=0)
getMonth()	Trả về tháng trong năm (from 0-11. 0=January, 1=February)
getFullYear()	Trả về giá trị năm (bốn số )
getYear()	Trả về giá trị năm (hai số )
getHours()	Trả về giờ của hệ thống (từ 0-23)
getMinutes()	Trả về phút của hệ thống (từ 0-59)
getSeconds()	Trả về giây của hệ thống (từ 0-59)
getMilliseconds()	Trả về giá trị millisecond from 0-999)

Ví dụ: Hiển thị giờ trên status bar

## ❏ String:

Mỗi chuỗi trong JavaScript là một đối tượng, gồm các thuộc tính và phương thức thực hiện trên chuỗi, đó là các phương thức tìm kiếm chuỗi, trích chuỗi con và áp dụng các thẻ HTML vào nội dung của chuỗi.

### Cách khai báo đối tượng String:

```
var str=new String();
```

### Thuộc tính của String():

- **Length:** xác định chiều dài của chuỗi.
- Các ký tự trong chuỗi được đánh chỉ số từ 0 đến Length-1. Tất cả các thành phần có giá trị chuỗi đều dùng được thuộc tính length.

## Một số phương thức của string

<b>Phương thức</b>	<b>Mô tả</b>	<b>Ví dụ</b>
charAt(n)	Trả về ký tự thứ n trong chuỗi	s= "Hello" s.charAt(2)= "l"
indexOf(str, n)	Trả về vị trí đầu tiên của chuỗi cần tìm str bắt đầu từ n, [-1: không có]	Vt=s.indexOf("ll",0) //2
lastIndexOf(str)	Trả về vị trí cuối cùng của chuỗi cần tìm str từ cuối chuỗi tới đầu chuỗi [-1: không có]	Vt=s.lastIndexOf("ll") //3
Substr(n, len)	Trả về 1 chuỗi con được ngắt từ vị trí thứ n và có len số ký tự	Str=s.substr(0,2) // He
Substring(n, m)	Trả về chuỗi con được ngắt từ vị trí n đến m trong chuỗi cha.	Str=s.substring(2,4) // llo

## ❑ Math ()

### Các phương thức:

**Math.method([value])**

Phương thức	Mô tả
abs(x)	Trả về giá trị tuyệt đối của biến x
acos(x)	Trả về giá trị arccosine của x
ceil(x)	Trả về số nguyên lớn hơn hoặc bằng x
floor(x)	Trả về số nguyên nhỏ hơn hoặc bằng x
log(x)	Trả về giá trị log của x
max(x, y)	Trả về giá trị lớn nhất trong hai số x và y
min(x, y)	Trả về giá trị nhỏ nhất trong hai số x và y
pow(x, y)	Trả về giá trị x lũy thừa y
random()	Trả về giá trị một số ngẫu nhiên từ 0 đến 1
round(x)	Làm tròn số x
sqrt(x)	Trả về giá trị căn bậc 2 của x



# Windows

---

## □ Các thuộc tính:

- Status: status mặc định cho windows
- Location: xác định vị trí của windows
- ...

## □ Các phương thức:

- Open(); mở windows mới
- Close(): đóng windows;

<b>Thuộc tính</b>	<b>Mô tả</b>	<b>Giá trị</b>
defaultStatus	thiết lập chuỗi t/báo trên thanh trạng thái	Text
status	thiết lập thông báo tại thời điểm hiện hành	Text
location	Xác định vị trí trang hiện tại trong cửa sổ	URL
history	Xác định các phần tử trong history	
alwaysLowered	hiển thị cửa sổ bên dưới các cửa sổ khác	Yes/no
alwaysRaised	hiển thị cửa sổ trên tất cả các cửa sổ khác	Yes/no
Dependent	Cửa sổ này sẽ đóng khi cửa sổ cha bị đóng	Yes/no
directories	Hiển thị Button thư mục	Yes/no
fullscreen	hiển thị chế độ đầy màn hình	Yes/no
height	thiết lập chiều cao của cửa sổ	số nguyên
hotkeys	Cho phép dùng phím nóng	Yes/no
left	Thiết lập k/cách từ văn bản đến cạnh cửa sổ	số nguyên
location	hiển thị hộp location	Yes/no
menubar	hiển thị thanh menu bar	Yes/no
resizable	Cho phép thay đổi kích thước cửa sổ	Yes/no
scrollbars	xuất hiện /không xuất hiện thanh cuộn	Yes/no
status	Hiển thị thanh trạng thái	Yes/no
titlebar	hiển thị thanh tiêu đề	Yes/no
toolbar	hiển thị thanh công cụ	Yes/no
width	Xác định độ rộng của cửa sổ	số nguyên
closed	trả về giá trị true, false. True khi cửa sổ đóng	true, false

---

### □ Ví dụ:

Tạo 2 liên kết trên trang, khi user di chuyển mouse qua liên kết nào thì hiển thị trên status thông báo về liên kết đó

□ Events: Sự kiện của windows chính là sự kiện của document gồm 2 sự kiện: **onload và unonLoad**

## □ Phương thức:

Phương Thức	Mô tả
<code>alert("msg")</code>	Hiển Thị hộp thoại thông báo
<code>blur()</code>	Di chuyển con trỏ đến cửa sổ hiện hành
<code>clearinterval(ID)</code>	Hủy thời gian đã thiết lập bằng <code>setinterval()</code>
<code>clearTimeout(ID)</code>	Hủy thời gian đã thiết lập bằng <code>setTimeout()</code>
<code>close()</code>	Đóng cửa sổ hiện hành
<code>confirm("msg")</code>	Hiển thị hộp thoại xác nhận với hai nút Cancel và OK
<code>focus()</code>	Đưa con trỏ về cửa sổ hiện hành
<code>MoveBy(x,y)</code>	Di chuyển cửa sổ đến một vị trí mới một đoạn pixel so với cửa sổ hiện hành
<code>MoveTo(x,y)</code>	Di chuyển cửa sổ qua trái và lên trên một đoạn pixel cụ thể so với cửa sổ hiện hành.

<code>open(URL,"windowname","FeatureList")</code>	<p>URL : là địa chỉ trang web muốn nạp vào cửa sổ.</p> <p>WindowName: là tên cửa sổ .</p> <p>FeatureList : là danh sách các thuộc tính của cửa sổ: thanh công cụ, thanh menu, thanh status ....</p>
<code>print()</code>	in nội dung trong cửa sổ hiện hành.
<code>prompt("msg","reply")</code>	Hiển thị hộp thoại nhập liệu
<code>setTimeout(func,millisec)</code>	Thiết lập thời gian mili giây để gọi một hàm
<code>stop()</code>	Hủy việc download một cửa sổ. Tương tự như việc đóng một cửa sổ trình duyệt.
<code>resizeBy(dx,dy)</code>	Thay đổi kích thước cửa sổ sang phải dx, dưới dy pixel
<code>resizeTo(x,y)</code>	Thay đổi kích thước x, y pixel
<code>scrollBy(dx,dy)</code>	cuộn nội dung sang phải dx, xuống dưới dy pixel
<code>scrollTo(x,y)</code>	cuộn nội dung trên trang đến vị trí x,y

Ví dụ: Tạo đoạn script gồm 3 button như hình bên

Open New Window

Close New Window

Close Main Window

```
<script type="text/JavaScript">
var win;
function mo()
{
    win=window.open('hinhInOut.html','NewWin',
        'toolbar=yes,status=yes,
        width=500,height=500');
}
function dong()
{
    win.close();
}
</script>
```

## □ document :

document cung cấp các thuộc tính và phương thức để làm việc với toàn bộ tài liệu hiện hành gồm: <form>, <a>, <img>, <title>, <table>, ...

document được định nghĩa khi <body> được xử lý trong trang HTML và nó vẫn tồn tại nếu trang được load lên. Các thuộc tính của document được thiết lập trong tag **<body>**. Trong <body> có 2 sự kiện **onLoad** và **onunload**.

```
<body onLoad="alert('Welcome to my web site');"  
onunload="alert('Goodbye, see you again');">
```

---

□ Ví dụ:

Viết đoạn script thực hiện khi load webpage cho phép user nhập tên, và hiện thị 1 window “Chào bạn X đến với web site”. Khi kết thúc đóng window hiển thị “Tạm biệt bạn X, hẹn gặp bạn lần sau.”



---

## ❑ Các thuộc tính của đối tượng **document**

- **alink**: màu cho liên kết
- **bgcolor**: màu nền cho trang
- **fgcolor**: màu chữ

Ví dụ:

```
document.bgcolor="red";  
document.fgcolor="yellow";
```

Thuộc tính	Mô tả
alinkcolor	Thiết lập hoặc trả về giá trị màu của liên kết đang xem của tài liệu
bgcolor	Thiết lập hoặc trả về giá trị màu nền của tài liệu
cookie	Chứa giá trị các cookies dành cho tài liệu hiện hành
domain	Trả về giá trị tên miền máy chủ chứa document
fgcolor	Thiết lập hoặc trả về giá trị màu chữ của tài liệu
lastmodified	Trả về giá trị ngày giờ cuối cùng mà tài liệu được cập nhật
linkcolor	Thiết lập hoặc trả về giá trị màu của liên kết trong tài liệu
location	mở một trang web mới
referrer	Returns the URL of the document that loaded the current document
title	Trả về giá trị của tựa đề của tài liệu
url	Trả về đường dẫn củ tài liệu hiện hành
vlinkcolor	Thiết lập hoặc trả về giá trị màu của liên kết đã xem của tài liệu

## ➤ Phương thức:

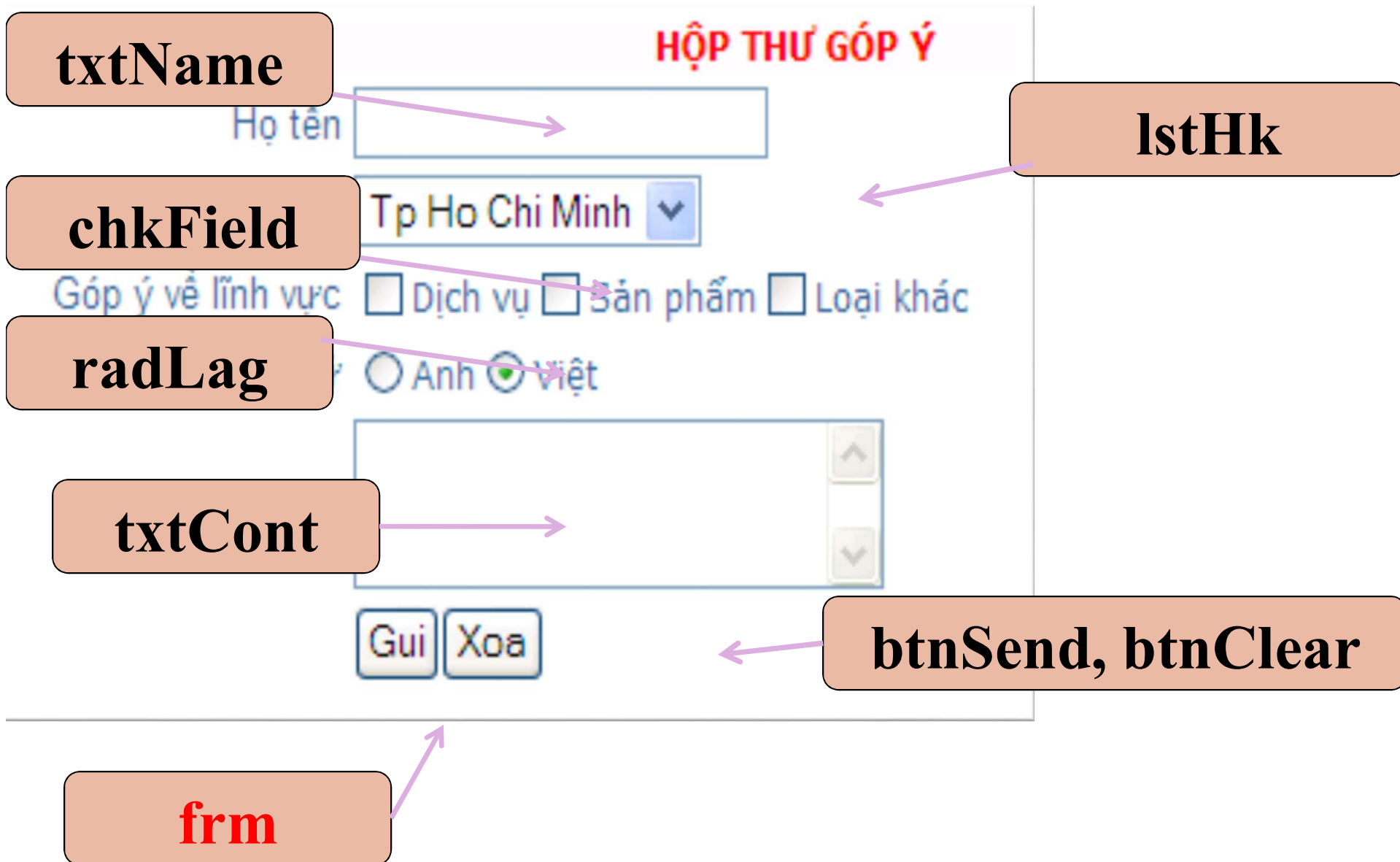
Thuộc tính	Mô tả
<code>clear()</code>	Xóa tài liệu
<code>close()</code>	Đóng một tài liệu
<code>focus()</code>	Đưa trở về một đối tượng trong trang
<code>open("mimetype",replace)</code>	
<code>write("str")</code>	viết một chuỗi vào một tài liệu
<code>writeln("str")</code>	viết một chuỗi vào một tài liệu và xuống dòng

## □ Events

---

Ví dụ: Viết đoạn mã Script thực hiện việc thay đổi màu nền của Webpage với giá trị màu do user nhập vào bởi hàm prompt khi trang Web được load lên

Form và cách làm việc trên form



# Form và cách làm việc trên form

## Các sự kiện của các phần tử trên form

Phần tử	Tên sự kiện
Button	onClick
Checkbox	onClick
Form	OnSubmit, onReset
Textbox	OnBlur,OnChange,OnFocus,Onselect
Radio	OnClick
Reset button	OnClick
Select	OnBlur,onChange,onFocus,onSelect
Submit button	OnClick
Textarea	OnBlur,OnChange,OnFocus,Onselect

# Cách đặt tên cho các phần tử trên form

---

☐ Textbox → txt

☐ Checkbox → chk

☐ Radio → rad

☐ Listbox → lst

☐ Button → btn

# Form

`<form id="name form" action="url file"  
method=[post/get]>`

---

Thuộc tính	Mô tả	Ví dụ
Action	Trả về đường dẫn (URL) đến tập tin xử lý của form thứ i	Document.forms[i].action
Length	Trả về số form trên trang web Hoặc trả về số phần tử trên form thứ i	Countform=document.forms.length Countfield=document.forms[i].length
Name	Trả về giá trị tên của form thứ i	Nameform=document.forms[i].name
Method	Các định phương thức của form thứ i	Methodform=document.forms[i].method
elements	mảng element chứa các phần tử trên form	document.form[i].elements[j].value



---

## ➤ Sự kiện form

- OnSubmit
- OnReset

```
<form id="form1" name="form1" method="post" action="frmsubmit.html">  
  <label>  
    <input type="submit" name="ok" id="ok" value="Submit" />  
  </label>  
  <label>  
    <input type="reset" name="cancel" id="cancel" value="Reset" />  
  </label>  
</form>
```

# Truy cập các phần tử trên form

## Cú pháp:

```
formName.formelement.properties
```

```
formName.formelement.method
```

## Ví dụ:

```
frm.txtName.value
```

```
frm.chkField.value
```

```
frm.chkField.checked
```

```
<form id="frm">
```

```
  <input id="txtName" name="txtName"  
    type="text" />
```

```
  <input type="Checkbox" id="chkField"  
    name="chkField" value="Ly thuyet" />
```

```
</form>
```

# Truy cập các phần tử trên form

```
document.getElementById("id").property
```

Ví dụ:

```
<form id="frm">  
  <input id="txtName" name="txtName" type="text" />  
  <input type="Checkbox" id="chkField"  
    name="chkField" value="Ly thuyet" />  
</form>
```

Truy cập

```
document.getElementById("txtName").value
```

```
document.getElementById("chkField").value
```

# Textbox

---

`<input type="text" name="name" value="buttonname" name_events="function_events()">`

Cách thức và thuộc tính	Mô tả
defaultValue	Chỉ ra giá trị mặc định của phần tử được chỉ ra trong thẻ INPUT
name	Tên của đối tượng được chỉ ra trong thẻ INPUT
value	Giá trị hiện thời của phần tử
readonly	[true , false]:
focus()	Mô tả việc con trỏ tới trường text.

**Events: onblur, onfocus**

---

```
<form id="frm">  
  <input id="txtso1" name="txtso1" type="text" />  
  <input id="txtkq" name="txtkq" type="text"  
    readonly=true />  
  ...  
</form>
```

```
var a=document.getElementById("txtso1").value;  
document.getElementById("txtso1").value="1900";  
document.getElementById("txtkq").value= "2000";  
frm.txtso1.focus();
```

# Regular Expression

## Biểu thức chính qui

---

Cú pháp: `/pattern/modifiers`

Trong đó:

`pattern`: chuỗi Regular Expression

`modifiers`: thông số cấu hình cho chuỗi với các giá trị:

`i`: so khớp chữ hoa và thường

`g`: so khớp toàn chuỗi

`m`: so khớp cả dữ liệu xuống dòng

Sử dụng hàm `test()` [`true/false`]: so khớp biểu thức chính qui

---

Partern chứa 2 loại ký tự:

❑ Ký tự thường bao gồm các ký tự trong bảng chữ cái: a,b,c,d,.. và các chuỗi thông thường được ghép từ các ký tự đó.

❑ Ký tự đặc biệt:

^ : Trả về chuỗi kết quả phù hợp với vị trí đầu của chuỗi gốc.

\$ : Trả về chuỗi kết quả phù hợp với vị trí cuối của chuỗi gốc.

: Ký tự Escape.

+ : Lặp lại 1 hay nhiều lần.

Expression	Mô tả
[abc]	Tìm các ký tự a, b, c.
[^abc]	Tìm các ký tự không phải a, b, c.
[0-9]	Tìm các ký tự là chữ số từ 0-9
[^0-9]	Tìm các ký tự là không phải chữ số từ 0-9
(x y)	Tìm ký tự x hoặc y
\d	Tìm các ký tự chữ số
\D	Tìm các ký tự không phải chữ số
\s	Tìm các khoảng trắng
\w	Tìm ký tự chữ cái
\W	Tìm ký tự không phải chữ cái
+	Kiểm tra ký tự xuất hiện một hoặc nhiều lần
*	Kiểm tra ký tự xuất hiện không hoặc nhiều lần
?	Kiểm tra ký tự xuất hiện không hoặc một lần
{X}	Kiểm tra ký tự xuất hiện đúng X lần
{X, Y}	Kiểm tra ký tự xuất hiện ít nhất X lần và tối đa Y lần
{X, }	Kiểm tra ký tự xuất hiện ít nhất X lần
^	Kiểm tra ký tự bắt đầu chuỗi
\$	Kiểm tra ký tự kết thúc chuỗi



Ví dụ: Biểu thức chính quy cho số CMND

```
pattern = "/^[0-9]{9}$/"
```

Ví dụ: Xây dựng biểu thức chính quy cho password từ 6-20 ký tự, con số hoặc các ký tự đặc biệt.

```
pattern = "/^[A-Za-z0-9(!@#$%^&*()_ ]{6,20} $/"
```

Ví dụ: Xây dựng biểu thức chính quy cho địa chỉ mail.

```
pattern = "/^[a-zA-Z0-9]+\@[a-zA-Z]{4,7}+\.[a-zA-Z]{3}/";
```

Ví dụ: Xây dựng biểu thức chính quy cho số điện thoại di động 10 hoặc 11 số.

```
pattern = "/^(01\d{9})|(09\d{8})$/";
```

Ví dụ: Xây dựng biểu thức chính quy cho mã thẻ cào.

```
pattern = "/^[0-9]{4}-[0-9]{4}-[0-9]{4}-[0-9]{4}$/";
```

## Ví dụ:

❑ Viết biểu thức chính quy cho các ví dụ sau:

- Kiểm tra tên đăng nhập chỉ chứa các ký tự
- Mật khẩu chứa các ký tự, con số, và các ký tự đặc biệt, có chiều dài từ 8-20
- Kiểm tra email hợp lệ  
[abc123abc@gmail.com]
- Email nhập lại trùng email đã nhập
- Nhập ngày sinh tính tuổi hợp lệ trên 16
- Kiểm tra lương là 6 con số [hàng trăm nghìn]

ĐĂNG KÝ THÀNH VIÊN

UserName  (\*)

Password  (\*)

Email  (\*)

Nhập lại Email  (\*)

Ngày sinh  (\*)

Lương  (\*)

# Button

```
<input type="button" name="name" value="buttonname"  
events_name="func_events()" disabled=true>
```

---

```
<script>
```

```
function pheptoan()
```

```
{
```

```
//các phép tính toán;
```

```
}
```

```
</script>
```

```
<input type="button" name="Button1"  
value="Cong" onclick="pheptoan()" />
```

# Checkbox

---

`<input type="checkbox" id="name" value="value"  
events_name="func_events()" checked=true>`

Phương thức và thuộc tính	Mô tả
checked	[true/false] Cho biết trạng thái hiện thời của checkbox
defaultChecked	Cho biết trạng thái mặc định của phần tử checkbox
name	Cho biết tên của phần tử checkbox
value	Cho biết giá trị hiện thời của phần tử checkbox

# Cách thức duyệt checkbox:

---

- ❑ Form luôn chứa nhiều thành phần checkbox → mảng checkbox  
→ sử dụng vòng lặp for duyệt qua các checkbox có được chọn hay không thông qua thuộc tính **checked**

```
for(var i=0; i<form1.length;i++)  
  
if(form1.elements[i].checked &&  
form1.elements[i].type=="checkbox")  
  
    var lo+=form1.elements[i].value
```

```

<form id="frm" name="frm">
  <p>Chon loai cafe : <br>
    <input name="chkcf" type="checkbox" id="chkcf" value="Trung Nguyen">Trung Nguyen
    <br>
    <input name="chkcf" type="checkbox" id="chkcf" value="MeKong">MeKong
    <br>
    <input name="chkcf" type="checkbox" id="chkcf" value="New Life">New Life
    <br>
    <input name="chkcf" type="checkbox" id="chkcf" value="Ocean Blue">Ocean Blue
    <br>
    <input name="chk5" type="checkbox" id="chkcf" value="Blue Space"> Blue Space
    <br>
    <input name="chkcf" type="checkbox" id="chkcf" value="Windows"> Windows<br>
    <input type="button" name="Button" value="Chọn" onClick="choncafe(frm.chkcf)">
  </form>
<p>Bạn chọn cafe: <font color="#FF0000"><span id='cafe'></span></font></p>

```

```

<script language="javascript" type="text/javascript">
function choncafe(field)
{
var cfcoun="";
for(var i=0; i<field.length;i++)

    if(field[i].checked)
        cfcoun+= field[i].value + ", ";

cafe.innerText=cfcoun;
}
</script>

```

Chon loai cafe :

☐ Trung Nguyen  
☒ Me Kong  
☐ New life  
☒ Ocean Blue  
☐ Blue Space  
☐ Windows

Bạn chọn cafe: **MeKong, Ocean Blue,**

# Radio

---

`<input type="radio" id="NAME" value= "value1"  
events_name="func_events()" checked=true>`

`<input type="radio" id="NAME" value= "value2"  
events_name="func_events()" checked=true>`

Phương thức và thuộc tính	Mô tả
checked	[true/false] Cho biết trạng thái hiện thời của checkbox
defaultChecked	Cho biết trạng thái mặc định của phân tử checkbox
name	Cho biết tên của phân tử checkbox
value	Cho biết giá trị hiện thời của phân tử checkbox

## Cách thức duyệt radio (group):

---

Muốn lấy giá trị của trường radio ta phải sử dụng đến mảng elements. Duyệt qua tất cả các phần tử và kiểm tra phần tử đó có được checked không?

Ví dụ:

```
for (var i=0; i<nameform.length; i++)  
if (nameform.elements[i].type == "radio"  
    && nameform.elements[i].checked == true)  
    Giatri = nameform.elements[i].value
```



Ví dụ:

```
<script>
```

```
function choncafe()  
{  
var m=frm.length;  
for(var i=0;i<m;i++)  
if(frm.elements[i].type=="radio" &&  
   frm.elements[i].checked==true)  
alert("Bạn đang chọn  
   cafe"+frm.elements[i].value);  
}
```

```
</script>
```

Chon loai cafe :

☐ Trung Nguyen

☐ Me Kong

☐ New life

☐ Ocean Blue

Click me!

## Selection list <select>

---

- ❑ DEFAULTSELECTED: cho biết option có được mặc định là lựa chọn trong thẻ OPTION
- ❑ INDEX: chứa giá trị số thứ tự của option hiện thời trong mảng option.
- ❑ SELECTED: cho biết trạng thái hiện thời của option
- ❑ TEXT: có chứa giá trị của dòng text hiển thị trên menu cho mỗi option,
- ❑ VALUED: giá trị chỉ ra trong thẻ OPTION.
- ❑ Sự kiện onChange

□ Ví dụ:

**<script>**

```
function change()  
{  
    var m=frm.xe.value;  
    hd.src=m;  
}
```

**</script>**



Chon loai xe :

Chon loai xe:

```
<select id="xe" name="xe" onChange="change () ">
```

---

## Bài tập (6 tiết)