

Summary HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
</body>
</html>
```

```
<a href="https://www.w3schools.com">This is a link</a>

<button>Click me</button>
```

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

```
<pre> viết như thế nào thì nó sẽ như vậy
```

- Use the **style** attribute for styling HTML elements
 - Use **background-color** for background color
 - Use **color** for text colors
 - Use **font-family** for text fonts
 - Use **font-size** for text sizes
 - Use **text-align** for text alignment
-
- **** - Bold text
 - **** - Important text – quan trọng
 - **<i>** - Italic text
 - **** - Emphasized text – nhấn mạnh , in nghiêng
 - **<mark>** - Marked text – đánh dấu – high light
 - **<small>** - Small text – làm chữ nhỏ
 - **** - Deleted text – gạch ngang
 - **<ins>** - Inserted text – gạch dưới

- `<sub>` - Subscript text – bên dưới
 - `<sup>` - Superscript text – bên trên
-

`<q>` quote

Phần `<blockquote>` tử HTML xác định một phần được trích dẫn từ một nguồn khác.

`<abbr>` viết tắt

`<p>`The `<abbr title="World Health Organization">`WHO`</abbr>` was founded in 1948.`</p>`

`<address>` địa chỉ in nghiêng

```
<!DOCTYPE html>
<html>
<body>

<p>The HTML address element defines
contact information (author/owner) of
a document or article.</p>

<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>

</body>
</html>
```

The HTML address element defines contact information (author/owner) of a document or article.

*Written by John Doe.
Visit us at:
Example.com
Box 564, Disneyland
USA*

`<cite>` xác định tiêu đề tác phẩm

```
<!DOCTYPE html>
<html>
<body>

<p>If your browser supports bi-directional
override (bdo), the next line will be
written from right to left (rtl):</p>

<bdo dir="rtl">This line will be written
from right to left</bdo>

</body>
</html>
```

If your browser supports bi-directional override (bdo), the next line will be written from right to left (rtl):

tfel ot thgir morf nettirw eb lliw enil sihT

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>

<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

To define a style for special types of elements, add a `class` attribute to the element:

```
<p class="error">I am different</p>
```

then define a style for the elements with the specific class:

Example

```
p.error {
  color: red;
}
```

Try it Yourself »

```
<link rel="stylesheet" href="https://www.w3schools.com/html/styles.css">
```

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}

a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}
```

```
a:hover {  
  color: red;  
  background-color: transparent;  
  text-decoration: underline;  
}
```

```
a:active {  
  color: yellow;  
  background-color: transparent;  
  text-decoration: underline;  
}
```

```
</style>
```

- `_blank` - Mở tài liệu được liên kết trong một cửa sổ hoặc tab mới
 - `_elf` - Mở tài liệu được liên kết trong cùng một cửa sổ / tab khi được nhấp (đây là mặc định)
 - `_parent` - Mở tài liệu được liên kết trong khung cha
 - `_top` - Mở tài liệu được liên kết trong toàn bộ cửa sổ
 - `framename` - Mở tài liệu được liên kết trong một khung có tên
-
- Sử dụng `` phần tử HTML để xác định hình ảnh
 - Sử dụng `src` thuộc tính HTML để xác định URL của hình ảnh
 - Sử dụng `alt` thuộc tính HTML để xác định văn bản thay thế cho hình ảnh, nếu nó không thể được hiển thị
 - Sử dụng HTML `width` và `height` các thuộc tính để xác định kích thước của hình ảnh
 - Sử dụng CSS `width` và `height` các thuộc tính để xác định kích thước của hình ảnh (cách khác)
 - Sử dụng thuộc tính CSS `float` để cho hình ảnh nổi
 - Sử dụng `<map>` phần tử HTML để xác định bản đồ hình ảnh
 - Sử dụng `<area>` phần tử HTML để xác định các khu vực có thể nhấp trong bản đồ hình ảnh
 - Sử dụng thuộc tính thành `` phần của HTML `usemap` để trỏ đến bản đồ hình ảnh
 - Sử dụng `<picture>` phần tử HTML để hiển thị các hình ảnh khác nhau cho các thiết bị khác nhau
-

```

<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
  text-align: left;
}
</style>
</head>
<body>

<h2>Cell that spans two columns</h2>
<p>To make a cell span more than one
column, use the colspan attribute.</p>

<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>

</body>
</html>

```

Cell that spans two columns

To make a cell span more than one column, use the colspan attribute.

Name	Telephone	
Bill Gates	55577854	55577855

```

<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
  text-align: left;
}
</style>
</head>
<body>

<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one row,
use the rowspan attribute.</p>

<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>

</body>
</html>

```

Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name:	Bill Gates
Telephone:	55577854
	55577855

- Sử dụng `<table>` phần tử HTML để xác định bảng
- Sử dụng `<tr>` phần tử HTML để xác định một hàng của bảng
- Sử dụng `<td>` phần tử HTML để xác định dữ liệu bảng
- Sử dụng `<th>` phần tử HTML để xác định tiêu đề bảng
- Sử dụng `<caption>` phần tử HTML để xác định chú thích bảng
- Sử dụng thuộc tính CSS `border` để xác định đường viền
- Sử dụng thuộc tính CSS `border-collapse` để thu gọn viền ô
- Sử dụng thuộc tính CSS `padding` để thêm phần đệm vào các ô
- Sử dụng thuộc tính CSS `text-align` để căn chỉnh văn bản ô
- Sử dụng thuộc tính CSS `border-spacing` để đặt khoảng cách giữa các ô
- Sử dụng `colspan` thuộc tính để tạo một ô kéo dài nhiều cột
- Sử dụng `rowspan` thuộc tính để tạo một ô kéo dài nhiều hàng
- Sử dụng `id` thuộc tính để xác định duy nhất một bảng
- Sử dụng `` phần tử HTML để xác định danh sách không có thứ tự

- Sử dụng thuộc tính CSS `list-style-type` để xác định điểm đánh dấu mục danh sách
- Sử dụng `` phần tử HTML để xác định danh sách theo thứ tự
- Sử dụng `type` thuộc tính HTML để xác định loại đánh số
- Sử dụng `` phần tử HTML để xác định mục danh sách
- Sử dụng `<dl>` phần tử HTML để xác định danh sách mô tả
- Sử dụng `<dt>` phần tử HTML để xác định thuật ngữ mô tả
- Sử dụng `<dd>` phần tử HTML để mô tả thuật ngữ trong danh sách mô tả
- Danh sách có thể được lồng trong danh sách
- Các mục danh sách có thể chứa các thành phần HTML khác
- Sử dụng thuộc tính CSS `float:left` hoặc `display:inline` để hiển thị danh sách theo chiều ngang

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

HTML Input Types

Input Type Text

`<input type="text">` defines a **one-line text input field**:

Example

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

[Try it Yourself »](#)

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

```
<form>
  User name:<br>
  <input type="text" name="username"><br>
  User password:<br>
  <input type="password" name="psw">
</form>
```

The characters in a password field are masked (shown as asterisks or circles).

```
<input type="submit" value="Submit">
```

```
<input type="reset">
```

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

Input Type Radio

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONLY ONE of a limited number of choices:

Example

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

- ☒ Male
 - ☐ Female
 - ☐ Other
-

Input Type Checkbox

`<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

- ☐ I have a bike
- ☐ I have a car

Input Type Button

`<input type="button">` defines a **button**:

Example

```
<input type="button" onclick="alert('Hello World!')" value="Click Me">
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Click Me!

HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

New input types that are not supported by older web browsers, will behave as `<input type="text">`.

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```

```
<!DOCTYPE html>
<html>
<body>

<h2>Numeric Steps</h2>
<p>Depending on browser support: Fixed
steps will apply in the input field.</p>

<form action="/action_page.php">
  Quantity:
  <input type="number" name="quantity"
    min="0" max="100" step="10" value="30">
  <input type="submit">
</form>

<p><b>Note:</b>type="number" is not
supported in IE9 and earlier.
</p>

</body>
</html>
```

Numeric Steps

Depending on browser support:
Fixed steps will apply in the input field.

Quantity:

Note:type="number" is not supported in IE9 and earlier.

1. Các **value** thuộc tính xác định giá trị ban đầu cho một lĩnh vực đầu vào:
2. Các **readonly** thuộc tính xác định rằng lĩnh vực đầu vào là chỉ đọc (không thể thay đổi):
3. Các **disabled** thuộc tính xác định rằng lĩnh vực đầu vào bị vô hiệu hóa.

4. Trường nhập bị vô hiệu hóa không thể sử dụng và không thể nhấp và giá trị của nó sẽ không được gửi khi gửi biểu mẫu:
 5. Các **size** thuộc tính xác định kích thước (bằng ký tự) cho lĩnh vực đầu vào:
 6. Các **maxlength** thuộc tính xác định tối đa cho phép chiều dài cho lĩnh vực đầu vào:
 7. Các **autofocus** thuộc tính xác định rằng trường nhập tự động nên được tập trung khi tải trang: chọn trường cần điền vào khi vào trang web
-

For example HTML FORM

Diện tích hcn

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Diện tích hình chữ nhật</title>
  <script>
    function tinhDienTich() {
      cdai=document.getElementById("txtchieudai").value;
      crong=document.getElementById("txtchieurong").value;
      dtich=cdai*crong*1;
      document.getElementById("txtdientich").value=dtich;
    }
  </script>
</head>
<body>
  <form action="" method="get">
    <table width="500px" align="center" cellpadding="0" cellspacing="0" border="1"
    style="background-color:cornsilk;color:midnightblue;">
      <caption style="background-color:cadetblue; font-size: 20px; color: #000;">Diện
      tích HCN</caption>
      <tr>
        <td width="200px">Chiều dài: </td>
        <td><input style="width:99%;" type="number" id="txtchieudai"
        required="required"></td>
      </tr>
      <tr>
        <td width="200px">Chiều rộng: </td>
        <td><input style="width:99%;" type="number" name="chieurong"
        id="txtchieurong"></td>
      </tr>
      <tr>
        <td width="200px">Diện tích: </td>
```

```

        <td><input style="width:99%;" type="number" name="dientich"
id="txtdientich"></td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <input type="button" value="Tính" onclick="tinhDienTich();">
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Đọc số

```

<form action="">
    <table width="500px" align="center" border="1" cellpadding="0">
        <caption>Đọc Số</caption>
        <tr>
            <td>Nhập số từ (0-999):</td>
            <td><input type="number" style="width:97%;" id="txtnumber"
required="required"></td>
            <td><input type="button" value="Đọc Số" onclick="docSo();"></td>
        </tr>
        <tr>
            <td>Bảng chữ: </td>
            <td colspan="2"><input style="width:98%;" type="text" id="txtchu"
readonly="readonly"></td>
        </tr>
    </table>
</form>

```

Checked box

```

    <tr>
        <td width="23%">
            <input type="checkbox" id="dulich"> Du lịch
        </td>
        <td width="23%">
            <input type="checkbox" name="v2" id="nghehac" value="2">Nghe Nhạc
        </td>
        <td width="23%">
            <input type="checkbox" name="v3" id="trang" value="3">Trắng
        </td>
        <td width="23%">
            <input type="checkbox" name="v4" id="do" value="4">Đỏ
        </td>
    </tr>

```

```

var cb_dl = document.getElementById("dulich");

var cb_nc = document.getElementById("nghehac");
var cb_trang = document.getElementById("trang");
if (cb_dl.checked == true) {
    st+="Du lịch, ";
}

```

```

    }
    if (cb_nc.checked == true) {
        st+="Nghe nhạc, ";
    }
    if (cb_trang.checked == true) {
        mausac+="Trắng, ";
    }
}

```

```

<select id="day">
    <option value="">Chọn Ngày</option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>

```

Khi get sẽ lấy giá trị value

Tính năm

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Tính năm âm lịch</title>
    <script>
        function doiNam() {
            var l_can=["Canh","Tân","Nhâm","Quý","Giáp","Ất","Bính","Đinh","Mậu","Kỷ"];
            var
l_chi=["Thân","Dậu","Tuất","Hợi","Tý","Sử","Dần","Mão","Thìn","Tỵ","Ngọ","Mùi"];
            var images=[
                "than.jpg",
                "dau.jpg",
                "tuat.jpg",
                "hoi.jpg",
                "ty.jpg",
                "suu.png",
                "dan.jpg",
                "mao.jpg",
                "thin.jpg",
                "ty1.jpg",
                "ngo.jpg",
                "mui.JPG"
            ];
            var namduonglich=document.getElementById("txtduonglich").value;
            var can = namduonglich % 10;
            var chi = namduonglich % 12;
            var namamlich = l_can[can] + ' ' + l_chi[chi];
            var hinh = images[chi];
            document.getElementById("txtamlich").value= namamlich;
            document.getElementById("img_hinh").src="images/"+hinh;
        }
    </script>

```

```

    </script>
</head>
<body>
    <form>
        <table width="500px" align="center" border="1" cellpadding="0">
            <caption><b>Tính năm âm lịch</b></caption>

            <tr>
                <td width="200px">Năm dương lịch</td>
                <td width="100px"></td>
                <td width="200px">Năm âm lịch</td>
            </tr>

            <tr>
                <td><input type="number" id="txtduonglich" required="required"></td>
                <td><input type="button" value="=" onclick="doiNam();"></td>
                <td><input type="text" id="txtamlich"></td>
            </tr>

            <tr>
                <td colspan="3" align="center"><img id="imghinh"></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

Đăng ký mua hoa

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>ĐĂNG KÝ MUA HOA</title>
    <script>
        function xuly() {
            var hten = document.getElementById("hoten").value;
            var dchi = document.getElementById("diachi").value;
            var sdt = document.getElementById("sdt").value;
            var email = document.getElementById("email").value;
            var lhoa = document.getElementById("loaihoa").value;
            var msac = document.getElementById("mausac").value;
            var sluong = document.getElementById("soluong").value;
            //start thanh toan
            var tmat = document.getElementById("tienmat");
            var ckhoan = document.getElementById("chuyenkhoan");
            var mcard = document.getElementById("mcard");
            //end thanh toan
            //start don vi tinh
            var nhanh = document.getElementById("nhanh");
            var so = document.getElementById("so");

```

```

var chau = document.getElementById("chau");
//end đơn vị tính
var ycau = document.getElementById("yeucau").value;

var str_thanhtoan = "Phương thức thanh toán: ";
var str_donvitinh = "Đơn vị tính: ";

if (tmat.checked == true) {
    str_thanhtoan += "Tiền mặt. ";
}
if (ckhoan.checked == true) {
    str_thanhtoan += "Chuyển khoản. ";
}
if (mcard.checked == true) {
    str_thanhtoan += "Master card. ";
}
if (nhanh.checked == true) {
    str_donvitinh += "Nhánh. ";
}
if (so.checked == true) {
    str_donvitinh += "Số. ";
}
if (chau.checked == true) {
    str_donvitinh += "Chậu. ";
}

var str_result = "Họ tên: " + hten + "\n" + "Địa Chỉ: " + dchi + "\n" +
    "SDT: " + sdt + "\n" + "Email: " + email + "\n" + "Loại hoa: " + lhoa
    + "\n" + "Màu sắc: " + msac + "\n" + "Số lượng: " + sluong + "\n"
    + str_donvitinh + "\n" + str_thanhtoan + "\n" + "Yêu cầu: " + ycau;

alert(str_result);
}
</script>
</head>
<body>
<form action="">
    <table width="600px" align="center" cellpadding="0" cellspacing="0" border="0">
        <tr>
            <td width="100px">Họ Tên:</td>
            <td colspan="3"><input type="text" style="width:99%" id="hoten" required></td>
        </tr>
        <tr>
            <td width="100px">Địa Chỉ:</td>
            <td colspan="3"><input type="text" style="width:99%" id="diachi" required></td>
        </tr>
        <tr>
            <td width="100px">SDT:</td>
            <td colspan="3"><input type="tel" style="width:99%" id="sdt" required></td>
        </tr>
        <tr>
            <td width="100px">Email:</td>
            <td colspan="3"><input type="email" style="width:99%" id="email" required></td>
        </tr>
    </table>
</form>

```

```

</tr>
<tr>
  <td width="100px">Loại hoa:</td>
  <td width="300px">
    <select name="loaihoa" id="loaihoa">
      <option value="Hoa Hồng">Hoa Hồng</option>
      <option value="Hoa Lay Ơn">Hoa Lay Ơn</option>
      <option value="Hoa Cúc">Hoa Cúc</option>
      <option value="Hoa Tulip">Hoa Tulip</option>
      <option value="Hoa Thược Dược">Hoa Thược Dược</option>
      <option value="Hoa Vạn Thọ">Hoa Vạn Thọ</option>
      <option value="Mai Vàng">Mai Vàng</option>
    </select>
  </td>
  <td width="100px">
    Màu sắc:
  </td>
  <td width="300px">
    <select name="mausac" id="mausac">
      <option value="Đỏ">Đỏ</option>
      <option value="Vàng">Vàng</option>
    </select>
  </td>
</tr>
<tr>
  <td width="100px">Số lượng:</td>
  <td colspan="3"><input type="number" style="width:99%" id="soluong" required></td>
</tr>
<tr>
  <td width="100px">Đơn vị tính:</td>
  <td width="233px"><input type="checkbox" name="nhanh" id="nhanh">Nhánh</td>
  <td width="233px"><input type="checkbox" name="so" id="so">Số</td>
  <td width="233px"><input type="checkbox" name="chau" id="chau">Chậu</td>
</tr>
<tr>
  <td width="100px">Thanh toán:</td>
  <td width="233px"><input type="checkbox" name="tienmat" id="tienmat">Tiền mặt</td>
  <td width="233px"><input type="checkbox" name="chuyenkhoan" id="chuyenkhoan">Chuyển khoản</td>
  <td width="233px"><input type="checkbox" name="mcard" id="mcard">Master Card</td>
</tr>
<tr>
  <td width="100px">Yêu cầu:</td>
  <td colspan="3">
    <textarea name="yeucau" id="yeucau" style="height:100px;width:
99%;"></textarea>
  </td>
</tr>
<tr>
  <td></td>
  <td><input type="button" value="Đăng ký" onclick="xuLy();"></td>
  <td colspan="2"><input type="reset" name="reset" id="reset" value="Xóa Thông
Tin"></td>

```



```

        </tr>
    </table>
</form>
</body>
</html>

```

Cột “CSS” chỉ ra phiên bản CSS mà thuộc tính đó được sử dụng(CSS1, CSS2 hoặc CSS3).

Các thuộc tính Color

Chức năng để thiết lập màu sắc và độ trong suốt của văn bản

Thuộc tính	Mô tả	CSS
color	Đặt màu sắc cho đoạn văn bản	1
opacity	Đặt mức độ trong suốt cho một phần tử nào đó	3

Các thuộc tính Background và Border

Chức năng thiết lập các thuộc tính màu nền và đường biên cho các đối tượng trên trang.

Thuộc tính	Mô tả	CSS
background	Một thuộc tính viết tắt để đặt thuộc tính nền trong lời khai báo	1
background-attachment	Đặt một ảnh nền để cố định hoặc cuộn với phần còn lại của trang	1
background-blend-mode	Chỉ định các chế độ trộn của mỗi lớp nền(màu sắc/hình ảnh)	3
background-color	Chỉ định màu nền của một phần tử	1
background-image	Chỉ định một hoặc nhiều hình ảnh nền cho một phần tử	1
background-position	Chỉ định vị trí của hình ảnh nền	1
background-repeat	Thiết lập như thế nào một ảnh nền sẽ được lặp đi lặp lại	1

background-clip	Chỉ định khu vực để sơn màu nền	3
background-origin	Chỉ định nơi các hình nền được định vị	3
background-size	Chỉ định kích thước của ảnh nền	3
border	Thiết lập tất cả các thuộc tính đường viền trong lời khai báo	1
border-bottom	Thiết lập tất cả các thuộc tính đường viền dưới trong lời khai báo	1
border-bottom-color	Thiết lập màu của đường viền dưới	1
border-bottom-left-radius	Xác định hình dạng của các đường viền góc dưới bên trái	3
border-bottom-right-radius	Xác định hình dạng của các đường viền góc dưới bên phải	3
border-bottom-style	Thiết lập kiểu (style) của đường viền dưới	1
border-bottom-width	Thiết lập độ rộng của đường viền dưới	1
border-color	Thiết lập màu sắc của bốn đường viền	1
border-image	Một thuộc tính viết tắt để thiết lập tất cả các thuộc tính border-image-*	3
border-image-outset	Chỉ định giá trị khu vực ảnh viền vượt ra ngoài vùng giới hạn của vùng bao	3
border-image-repeat	Chỉ định ảnh viền nên được lặp đi lặp lại, tròn hoặc kéo dài	3
border-image-slice	Chỉ định cụ thể như thế nào để cắt ảnh viền	3
border-image-source	Chỉ định đường dẫn tới hình ảnh để sử dụng như đường viền	3
border-image-width	Chỉ định độ rộng của ảnh-viền	3
border-left	Thiết lập tất cả thuộc tính đường viền trái trong lời khai báo	1
border-left-color	Thiết lập màu sắc của đường viền trái	1

border-left-style	Thiết lập kiểu (style) của đường viền trái	1
border-left-width	Thiết lập độ rộng của đường viền trái	1
border-radius	Thuộc tính viết tắt để thiết lập tất cả bốn đường viền-*-thuộc tính radius(bán kính)	3
border-right	Thiết lập thuộc tính đường viền phải trong lời khai báo	1
border-right-color	Thiết lập màu sắc cho đường viền phải	1
border-right-style	Thiết lập kiểu (style) cho đường viền phải	1
border-right-width	Thiết lập độ rộng cho đường viền phải	1
border-style	Thiết lập kiểu (style) cho bốn đường viền	1
border-top	Thiết lập thuộc tính đường viền trên trong lời khai báo	1
border-top-color	Thiết lập màu sắc cho đường viền trên	1
border-top-left-radius	Xác định hình dạng của đường viền góc trên bên trái	3
border-top-right-radius	Xác định hình dạng của đường viền góc trên bên phải	3
border-top-style	Thiết lập kiểu (style) cho đường viền trên	1
border-top-width	Thiết lập độ rộng của đường viền trên	1
border-width	Thiết lập độ rộng của bốn đường viền	1
box-decoration-break	Thiết lập dáng vẽ của hình nền và đường viền của một phần tử tại trang ngắt, hoặc đối với phần tử in-line , tại dòng ngắt.	3
box-shadow	Gắn một hoặc nhiều đổ bóng (drop-shadows) vào hộp	3

Các thuộc tính Basic Box

Thuộc tính	Mô tả	CSS
------------	-------	-----

bottom	Xác định vị trí cuối của phần tử position (vị trí)	2
clear	Xác định hai bên của phần tử (left,right) nơi mà phần tử float không được cho phép (ngăn cản thành phần không được float trái, phải hay cả hai)	1
clip	Xác định đoạn cho phần tử khi sử dụng thuộc tính position có giá trị " <i>absolute</i> "	2
display	Chỉ định làm thế nào một phần tử HTML nào đó sẽ được hiển thị	1
float	Xác định có hay không một box được float(trôi nổi)	1
height	Thiết lập chiều cao của thành phần	1
left	Xác định vị trí bên trái của phần tử position	2
margin	Thiết lập thuộc tính margin (căn lề cho phần tử) trong một thông báo	1
margin-bottom	Thiết lập lề dưới của một phần tử	1
margin-left	Thiết lập lề trái của một phần tử	1
margin-right	Thiết lập lề phải của một phần tử	1
margin-top	Thiết lập lề trên của một phần tử	1
max-height	Thiết lập chiều cao tối đa của một phần tử	2
max-width	Thiết lập chiều rộng tối đa của một phần tử	2
min-height	Thiết lập chiều cao tối thiểu của một phần tử	2
min-width	Thiết lập chiều rộng tối thiểu của một phần tử	2
overflow	Chỉ định những gì sẽ xảy ra nếu nội dung tràn ra ngoài phần tử hộp	2
overflow-x	Xác định có hay không để cắt cạnh trái/phải của đoạn văn bản, nếu nó tràn ra khỏi khu vực nội dung của phần tử	3
overflow-y	Xác định có hay không để cắt cạnh trên /dưới của đoạn văn bản, nếu nó tràn ra khỏi khu vực nội dung của phần tử	3

padding	Thiết lập lại thuộc tính padding (vùng đệm) trong lời khai báo	1
padding-bottom	Thiết lập vùng đệm dưới của phần tử	1
padding-left	Thiết lập vùng đệm trái của một phần tử	1
padding-right	Thiết lập vùng đệm phải của một phần tử	1
padding-top	Thiết lập vùng đệm trên của một phần tử	1
position	Xác định kiểu của phương thức định vị được sử dụng cho một phần tử (static, relative, absolute or fixed)	2
right	Xác định vị trí phải của phần tử position	2
top	Xác định vị trí trên của phần tử position	2
visibility	Xác định có hay không một phần tử có thể nhìn thấy được	2
width	Thiết lập độ rộng của phần tử	1
vertical-align	Sắp xếp nội dung theo chiều dọc của phần tử	1
z-index	Thiết lập thứ tự xếp chồng nhau của phần tử position	2

Thuộc tính cho bố cục Flexible Box

Thuộc tính	Mô tả	CSS
align-content	Xác định sự căn chỉnh giữa các dòng bên trong một khối linh hoạt (flexible container) khi các mục không sử dụng các khoảng cách cố sẵn	3
align-items	Xác định sự căn chỉnh cho các mục bên trong một khối linh hoạt (flexible container)	3
align-self	Xác định sự căn chỉnh cho các mục được chọn bên trong một khối linh hoạt (flexible container)	3

flex	Xác định độ dài của các mục, tương đối với phần còn lại	3
flex-basis	Xác định độ dài ban đầu của một mục linh hoạt (flexible item)	3
flex-direction	Xác định hướng của các mục linh hoạt (flexible items)	3
flex-flow	Một thuộc tính viết tắt cho thuộc tính flex-direction và flex-wrap	3
flex-grow	Chỉ định bao nhiêu mục sẽ tăng tương đối so với phần còn lại	3
flex-shrink	Chỉ định bao nhiêu mục sẽ giảm tương đối so với phần còn lại	3
flex-wrap	Chỉ định liệu các mục linh hoạt (flexible items) nên bọc hay không	3
justify-content	Xác định sự căn chỉnh giữa các mục bên trong một khối linh hoạt (flexible container) khi các mục không sử dụng tất cả các khoảng cách có sẵn	3
order	Thiết lập thứ tự của các mục linh hoạt (flexible items), tương đối so với phần còn lại	3

Các thuộc tính Text

Thuộc tính	Mô tả	CSS
hanging-punctuation	Xác định ký tự chấm câu có thể được đặt ở ngoài đường hộp	3
hyphens	Thiết lập làm thế nào tách từ để cải thiện cách bố trí của đoạn văn	3
letter-spacing	Tăng hoặc giảm khoảng cách giữa các ký tự trong một văn bản	1
line-break	Xác định cách ngắt dòng	3
line-height	Thiết lập chiều cao của dòng	1
overflow-wrap	Xác định có hay không trình duyệt có thể ngắt dòng trong các từ để ngăn cản việc tràn (Khi một chuỗi quá dài để phù hợp với hộp chứa nó)	3
tab-size	Xác định chiều dài của các ký tự tab	3

text-align	Xác định căn chỉnh nội dung theo chiều ngang	1
text-align-last	Mô tả cách dòng cuối cùng của một đoạn hoặc một dòng ngay trước khi một ngắt dòng là căn chỉnh khi text-align là "justify"	3
text-combine-upright	Xác định sự kết hợp của đa ký tự vào khoảng cách của một ký tự đơn	3
text-indent	Xác định rõ sự thụt dòng đầu tiên trong một khối văn bản	1
text-justify	Xác định phương pháp căn lề thẳng hàng hai bên sử dụng khi text-align là "justify"	3
text-transform	Điều khiển các chữ in hoa	1
white-space	Xác định cách mà khoảng trắng bên trong một phần tử được xử lý	1
word-break	Xác định quy tắc ngắt dòng đối với scripts non-CJK	3
word-spacing	Tăng hoặc giảm khoảng cách giữa các từ trong đoạn văn	1
word-wrap	Cho phép dài, những từ dài được xuống hàng mà không làm vỡ layout	3

Các thuộc tính Text Decoration

Thuộc tính	Mô tả	CSS
text-decoration	Xác định các trang trí thêm vào văn bản	1
text-decoration-color	Xác định màu của văn bản trang trí (text-decoration)	3
text-decoration-line	Xác định loại của dòng trong text-decoration	3
text-decoration-style	Xác định kiểu của các dòng trong một văn bản trang trí	3
text-shadow	Thêm đổ bóng cho văn bản	3
text-underline-position	Xác định vị trí của các gạch dưới được thiết lập bằng cách sử dụng thuộc tính text-decoration	3

Các thuộc tính Font

Thuộc tính	Mô tả	CSS
@font-face	Một quy tắc cho phép các trang web tải và sử dụng các phông chữ khác với phông chữ "web-safe"	3
@font-feature-values	Cho phép tác giả sử dụng một tên chung trong font-variant-alternate đối với tính năng kích hoạt khác nhau trong OpenType	3
font	Đặt tất cả các thuộc tính font trong lời khai báo	1
font-family	Xác định họ phông chữ cho văn bản	1
font-feature-settings	Cho phép kiểm soát các tính năng về in tiên tiến trong OpenType fonts	3
font-kerning	Kiểm soát việc sử dụng các thông tin kerning (cách các từ cách nhau)	3
font-language-override	Kiểm soát việc sử dụng của ngôn ngữ cụ thể nét trạm (language-specific glyphs) trong kiểu chữ	3
font-size	Xác định kích thước phông chữ của đoạn văn	1
font-size-adjust	Duy trì khả năng đọc văn bản khi phông chữ dự phòng xuất hiện	3
font-stretch	Chọn kiểu normal, condensed, hoặc expanded từ họ phông chữ	3
font-style	Xác định kiểu chữ cho văn bản	1
font-synthesis	Quản lý các kiểu chữ (đậm hoặc nghiêng) có thể được tổng hợp bởi trình duyệt	3
font-variant	Xác định có hay không một văn bản được hiển thị với phông chữ small-caps	1
font-variant-alternates	Kiểm soát việc sử dụng các nét chạm thay thế liên quan đến việc thay tên được định nghĩa trong @font-feature-values	3
font-variant-caps	Kiểm soát việc sử dụng nét chạm (glyph) thay thế cho chữ in hoa	3

font-variant-east-asian	Kiểm soát việc sử dụng nét chạm (glyph) thay thế cho các kịch bản Đông Á (ví dụ như Trung Quốc và Nhật Bản)	3
font-variant-ligatures	Kiểm soát các chữ ghép và các biểu mẫu theo ngữ cảnh được sử dụng trong các nội dung văn bản của phần tử mà nó được áp dụng	3
font-variant-numeric	Kiểm soát việc sử dụng các họa tiết (glyph) thay thế cho số, phân số và đánh dấu thứ tự	3
font-variant-position	Kiểm soát việc sử dụng các họa tiết thay thế kích thước nhỏ hơn vị trí như là chỉ số trên hoặc chỉ số dưới về đường cơ sở của các phong chữ	3
font-weight	Xác định độ đậm của phong chữ	1

Các thuộc tính Writing Modes

Thuộc tính	Mô tả	CSS
direction	Xác định hướng văn bản/ hướng bài viết	2
text-orientation	Xác định hướng của văn bản trong một dòng	3
text-combine-upright	Xác định sự kết hợp của đa ký tự vào khoảng cách của ký tự đơn	3
unicode-bidi	Được sử dụng cùng với thuộc tính direction để đặt và trả về cho dù văn bản nên được ghi đề để hỗ trợ nhiều ngôn ngữ trong cùng một tài liệu	2
writing-mode	Xác định liệu dòng văn bản được đặt ra theo chiều ngang hay chiều dọc hay hướng mà khối tiến tới	3

Các thuộc tính Table

Thuộc tính	Mô tả	CSS
border-collapse	Xác định liệu có hay không đường viền của bảng nên tách biệt	2
border-spacing	Xác định khoảng cách giữa đường viền của các ô lân cận	2
caption-side	Xác định vị trí của một chú thích bảng	2

empty-cells	Xác định có hay không để hiển thị đường viền và nền trên ô trống trong một bảng	2
table-layout	Thiết lập thuật toán bố trí (layout algorithm) được sử dụng cho bảng	2

Các thuộc tính Lists và Counters

Thuộc tính	Mô tả	CSS
counter-increment	Gia tăng một hoặc nhiều counters	2
counter-reset	Tạo hoặc thiết lập lại một hay nhiều counters	2
list-style	Thiết lập tất cả các thuộc tính cho danh sách trong lời khai báo	1
list-style-image	Xác định một hình ảnh như đánh dấu mục danh sách	1
list-style-position	Xác định nếu đánh danh mục sẽ xuất hiện bên trong hoặc bên ngoài nội dung	1
list-style-type	Xác định loại của đánh dấu danh mục	1

Các thuộc tính Animation

Thuộc tính	Mô tả	CSS
@keyframes	Xác định mã animation	3
animation	Một thuộc tính viết tắt cho các thuộc tính animation (ngoại trừ animation-play-state và animation-fill-mode)	3
animation-delay	Xác định sự chậm trễ đối với sự bắt đầu của một chuyển động của hình ảnh hoặc tag (animation)	3
animation-direction	Xác định có hay không các chuyển động nên chạy ngược lại trên chu kỳ thay thế	3
animation-duration	Xác định có bao nhiêu giây hoặc mili giây một chuyển động cần để hoàn thành một chu kỳ	3

animation-fill-mode	Xác định kiểu cho các phần tử khi các chuyển động không chạy (Khi nó dừng lại , hoặc khi nó bị trễ)	3
animation-iteration-count	Xác định số lần một chuyển động được thực hiện	3
animation-name	Xác định tên của @keyframes animation	3
animation-play-state	Xác định xem các chuyển động đang được chạy hay tạm dừng	3
animation-timing-function	Xác định tốc độ cong của một chuyển động của hình ảnh hoặc tag	3

Các thuộc tính Transform

Thuộc tính	Mô tả	CSS
backface-visibility	Xác định có hay không một phần tử nên được nhìn thấy khi không ở chế độ toàn màn hình	3
perspective	Xác định chiều sâu	3
perspective-origin	Xác định vị trí dưới của phần tử 3D	3
transform	Áp dụng một chuyển đổi 2D hoặc 3D đến một phần tử	3
transform-origin	Cho phép bạn thay đổi vị trí trên phần tử transform	3
transform-style	Xác định các phần tử lồng nhau sẽ thế nào trong không gian 3D	3

Các thuộc tính Transitions

Thuộc tính	Mô tả	CSS
transition	Một thuộc tính viết tắt để thiết lập cho 4 thuộc tính chuyển đổi	3
transition-property	Xác định tên của thuộc tính CSS trong hiệu ứng của quá trình chuyển đổi(none, width, height, all)	3

transition-duration	Xác định bao nhiêu giây hoặc mili giây một hiệu ứng chuyển đổi hoàn thành	3
transition-timing-function	Xác định đường cong tốc độ của hiệu ứng chuyển đổi	3
transition-delay	Xác định khi nào hiệu ứng chuyển đổi sẽ bắt đầu	3

Các thuộc tính Basic User Interface

Thuộc tính	Mô tả	CSS
box-sizing	Bảo cho trình duyệt các thuộc tính sizing (độ rộng và độ cao) nên có	3
content	Sử dụng với :before và :after pseudo-elements, để chèn nội dung được tạo ra	2
cursor	Xác định kiểu con trỏ chuột sẽ được hiển thị khi di chuyển vào phần tử nào đó	2
ime-mode	Điều khiển trạng thái của trình soạn thảo phương thức nhập cho trường văn bản	3
nav-down	Xác định nơi để di chuyển khi sử dụng phím mũi tên xuống	3
nav-index	Xác định thứ tự tab của phần tử	3
nav-left	Xác định nơi để di chuyển khi sử dụng phím mũi tên sang trái	3
nav-right	Xác định nơi để di chuyển khi sử dụng phím mũi tên sang phải	3
nav-up	Xác định nơi để di chuyển khi sử dụng phím mũi tên lên	3
outline	Thiết lập thuộc tính đường viền trong lời khai báo	2
outline-color	Đặt màu sắc của đường viền bao ngoài	2
outline-offset	Đệm đường viền bao ngoài, và rút nó ra bên ngoài cạnh đường viền (border)	3
outline-style	Thiết lập kiểu của đường viền bao ngoài	2
outline-width	Thiết lập độ rộng của đường viền bao ngoài	2
resize	Xác định có hay không một phần tử là thay đổi kích thước bởi người dùng	3

text-overflow	Xác định các vấn đề khi văn bản tràn khỏi các phần tử container	3
---------------	---	---

Các thuộc tính Multi-column Layout

Thuộc tính	Mô tả	CSS
break-after	Xác định cách xử lý ngắt trang, ngắt cột hoặc ngắt đoạn sau khi tạo hộp	3
break-before	Xác định cách xử lý ngắt trang, ngắt cột hoặc ngắt đoạn trước khi tạo hộp	3
break-inside	Xác định cách xử lý ngắt trang, ngắt cột hoặc ngắt đoạn bên trong hộp được tạo ra	3
column-count	Xác định số cột phần tử nên được chia thành	3
column-fill	Xác định như thế nào để điền vào cột	3
column-gap	Xác định khoảng cách giữa các cột	3
column-rule	Một thuộc tính viết tắt để thiết lập thuộc tính column-rule-* cho các đường kẻ giữa các cột	3
column-rule-color	Xác định màu sắc của các đường kẻ giữa các cột	3
column-rule-style	Xác định kiểu của các đường kẻ giữa các cột	3
column-rule-width	Xác định độ rộng của các đường kẻ giữa các cột	3
column-span	Xác định có bao nhiêu cột chứa phần tử được trải dài	3
column-width	Xác định độ rộng của cột	3
columns	Một thuộc tính viết tắt để thiết lập độ rộng cột (column-width) và số cột (column-count)	3
widows	Xác định số dòng tối thiểu phải còn lại trên một trang khi ngắt trang xảy ra bên trong phần tử	2

Paged Media

Thuộc tính	Mô tả	CSS
orphans	Thiết lập số dòng tối thiểu phải còn lại ở cuối của một trang khi ngắt trang xảy ra bên trong phần tử	2
page-break-after	Thiết lập chế độ ngắt trang sau một phần tử	2
page-break-before	Thiết lập chế độ ngắt trang trước một phần tử	2
page-break-inside	Thiết lập chế độ ngắt trang bên trong một phần tử	2

Generated Content cho trang đa phương tiện

Thuộc tính	Mô tả	CSS
marks	Thêm đoạn văn bản được đánh dấu và/hoặc sử dụng cho tài liệu	3
quotes	Thiết lập các loại dấu bao ngoài khi nhúng một trích dẫn	2

Các thuộc tính Filter Effects

Thuộc tính	Mô tả	CSS
filter	Xác định hiệu ứng (vd: làm mờ hoặc chuyển đổi màu sắc) trên một phần tử trước khi nó được hiển thị	3

Image Values và Replaced Content

Thuộc tính	Mô tả	CSS

image-orientation	Xác định sự quay theo bên phải hoặc theo chiều kim đồng hồ do người dùng áp dụng cho một ảnh (Thuộc tính này có khả năng bị phản đối và chức năng của nó chuyển sang cho HTML)	3
image-rendering	Gợi ý cho các trình duyệt về các khía cạnh của một ảnh là rất quan trọng để bảo vệ khi hình ảnh được thu nhỏ lại	3
image-resolution	Xác định độ phân giải nội tại của các ảnh raster được sử dụng trong/ trên phần tử	3
object-fit	Xác định làm thế nào nội dung của một phần tử thay thế nên được trang bị hộp được tạo bởi độ cao và độ rộng	3
object-position	Xác định căn chỉnh của phần tử thay thế bên trong hộp của nó	3

Các thuộc tính Masking

Thuộc tính	Mô tả	CSS
mask		3
mask-type		3

Các thuộc tính Speech

Thuộc tính	Mô tả	CSS
mark	Một thuộc tính viết tắt để thiết lập thuộc tính mark-before và mark-after	3
mark-after	Cho phép đánh dấu tên được gắn liền với những dòng âm thanh	3
mark-before	Cho phép đánh dấu tên được gắn liền với những dòng âm thanh	3
phonemes	xác định cách phát âm cho các văn bản chứa các phần tử tương ứng	3

rest	Một thuộc tính viết tắt để thiết lập thuộc tính the rest-before và rest-after	3
rest-after	Xác định phần còn lại hoặc ranh giới điều kiện tính được quan sát sau khi nói phần tử nội dung	3
rest-before	Xác định phần còn lại hoặc ranh giới điều kiện để quan sát trước khi nói phần tử nội dung	3
voice-balance	Xác định sự cân bằng giữa các kênh trái và phải	3
voice-duration	Xác định nó phải mất bao lâu để trả lại các lựa chọn phần tử nội dung	3
voice-pitch	Xác định mức độ trung bình (tần số) của giọng nói	3
voice-pitch-range	Xác định sự thay đổi của tần số	3
voice-rate	Điều khiển tốc độ nói	3
voice-stress	Chỉ ra cường độ của sự nhấn mạnh được áp dụng	3
voice-volume	Đề cập đến biên độ của dạng sóng đầu ra của các bài phát biểu	3

Các thuộc tính Marquee

Chức năng tạo chữ chạy trong HTML

Thuộc tính	Mô tả	CSS
marquee-direction	Thiết lập hướng di chuyển nội dung	3
marquee-play-count	Thiết lập bao nhiêu lần di chuyển nội dung	3
marquee-speed	Thiết lập cách nhanh chóng cuộn nội dung	3
marquee-style	Thiết lập kiểu di chuyển nội dung	3

Summary JS


```
document.getElementById("txtkq").innerHTML=kq3; //for tag normal
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

```
document.getElementById("txtkq").value=kq;
```

```
var sodau = document.getElementById("txtbatdau").value;
```

Thay đổi giá trị HTML

```
document.getElementById("demo").style.fontSize = "35px";
or
document.getElementById('demo').style.fontSize = '35px';
```

Thay đổi font size của chữ

```
document.getElementById("demo").style.display = "block";
or
document.getElementById('demo').style.display = 'block';
```

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can show hidden HTML elements.</p>

<p id="demo" style="display:none">Hello JavaScript!
</p>

<button type="button"
onclick="document.getElementById('demo').style.display
='block'">Click Me!</button>

</body>
</html>
```

What Can JavaScript Do?

JavaScript can show hidden HTML elements.

Hello JavaScript!

Click Me!

Tìm kiếm chuỗi con

Chúng ta có ba hàm thường dùng để tìm kiếm chuỗi con trong Javascript như sau:

- indexOf()
- lastIndexOf()
- search()

Hàm indexOf()

Để tìm kiếm chuỗi con thì ta sử dụng hàm `String.indexOf(str)`, trong đó `str` là chuỗi con và `String` là chuỗi cha. Hàm này sẽ trả kết quả về kết quả là vị trí xuất hiện đầu tiên của chuỗi (bắt đầu là vị trí 0), nếu không tìm thấy chuỗi con thì nó sẽ trả về -1.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Chào mừng bạn đến với freetuts.net";
2 document.write("Vị trí xuất hiện chuỗi freetuts.net là: " + string.indexOf("freetuts.net"));
```

Hàm `lastIndexOf()`

Trường hợp nếu chuỗi con xuất hiện nhiều lần trong chuỗi cha thì kết quả cũng trả về vị trí xuất hiện của chuỗi con đầu tiên. Vậy làm thế nào để lấy vị trí của chuỗi con cuối cùng trong chuỗi cha? Ta sẽ sử dụng hàm `String.lastIndexOf(str)`, hàm này sẽ trả về vị trí xuất hiện của chuỗi con cuối cùng và trả về -1 nếu không tìm thấy.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Website freetuts.net - học lập trình miễn phí tại freetuts.net";
2 document.write("Vị trí xuất hiện chuỗi freetuts.net là: " +
  string.lastIndexOf("freetuts.net"));
```

Hàm `search()`

Ngoài hai hàm trên bạn có thể sử dụng hàm `string.search(str)` để tìm kiếm, tác dụng của nó cũng giống như hàm `string.indexOf(str)`.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Chào mừng bạn đến với freetuts.net";
2 document.write("Vị trí xuất hiện chuỗi freetuts.net là: " +
  string.search("freetuts.net")); <br>
```

Cắt chuỗi con

Nếu bạn muốn cắt một chuỗi con từ chuỗi cha thì bạn có thể sử dụng ba hàm sau:

- `slice(start, end)`
- `substring(start, end)`
- `substr(start, length)`

Note: tất cả các vị trí của chuỗi đều bắt đầu từ 0, vì vậy khi tính toán vị trí coi chừng bị nhầm lẫn nhé.

Hàm `slice()`

Hàm `slice` có hai tham số truyền vào:

- **start**: vị trí bắt đầu
- **end**: vị trí kết thúc

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome to freetuts.net";  
2 document.write("Chuỗi cần lấy là: " + string.slice(11, 23));<br>
```

Nếu tham số truyền vào là số âm thì nó sẽ tính ngược lại, nghĩa là nó sẽ đếm từ cuối lên.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome to freetuts.net";  
2 document.write("Chuỗi cần lấy là: " + string.slice(-12, 23));
```

Nếu bạn chỉ truyền một tham số đầu tiên thì nó sẽ tự hiểu vị trí **end** là vị trí cuối cùng.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome to freetuts.net";  
2 document.write("Chuỗi cần lấy là: " + string.slice(5));
```

Hàm `substring()`

Hàm `substring()` có cách sử dụng giống với hàm `slice()`, tuy nhiên tham số truyền vào hàm `substring()` phải luôn luôn lớn hơn 0.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome to freetuts.net";  
2 document.write("Chuỗi cần lấy là: " + string.substring(11, 23));
```

Hàm `substr()`

Hàm `substr()` có hai tham số là **start** và **length**, trong đó **start** là vị trí bắt đầu và **length** là số ký tự muốn lấy bắt đầu từ vị trí **start**. Nếu bạn truyền tham số **start** là số âm thì nó sẽ tính từ cuối trở lên, còn tham số **length** phải luôn luôn là số dương.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome to freetuts.net";  
2 document.write("Chuỗi cần lấy là: " + string.substr(11, 12));
```

Tìm kiếm và lặp chuỗi

Để tìm kiếm và lặp một chuỗi con nào đó thì bạn sử dụng hàm `replace(str_find, str_replace)`, trong đó `str_find` là chuỗi cần tìm và `str_replace` là chuỗi sẽ được thay thế chuỗi `str_find`.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome to freetuts.net";
2 document.write(string.replace("freetuts.net", "freetuts.net"));
```

Chuyển thành chữ hoa và chữ thường

Để chuyển chuỗi thành chữ hoa ta dùng hàm `toUpperCase()` và chuyển thành chữ thường ta dùng hàm `toLowerCase()`.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome to freetuts.net";
2 document.write(string.toUpperCase() + "<br/>");
3 document.write(string.toLowerCase());
```

Nối thêm chuỗi

Để nối thêm chuỗi thông thường ta dùng toán tử `+`, ngoài ra bạn có thể dùng hàm `concat()` để thực hiện nối chuỗi.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome " + "to" + " freetuts.net";
2 document.write(string + "<br/>");
3
4 // hoặc
5 var string = "Welcome ";
6 string = string.concat("to ", "freetuts.net");
7 document.write(string + "<br/>");
```

Tìm ký tự hoặc mã ASCII của một ký tự

Để xem ký tự của một vị trí nào đó thì dùng hàm `charAt()`, còn xem [mã ASCII](#) thì dùng hàm `charCodeAt()`. Cả hai hàm này đều có tham số truyền vào là vị trí muốn xem.

Ví dụ: [XEM DEMO](#)

```
1 var string = "Welcome freetuts.net";
2
3 document.write(string.charAt(1) + "<br/>");
4 document.write(string.charCodeAt(1) + "<br/>");
```

Chuyển đổi chuỗi sang mảng

Để chuyển một chuỗi sang mảng thì ta sử dụng hàm `split()` với tham số truyền vào là ký tự ngăn cách giữa các phần tử.

Hàm xử lý chuỗi

Nhóm toàn cục:

- `Number(type)` trả về một number và định dạng cơ số `type` = (*nhị phân, thập phân, thập lục phân*)
- `parseFloat()` chuyển sang một số float
- `parseInt()` chuyển sang một số integer

Nhóm cục bộ:

- `toString()` chuyển sang kiểu `string`
- `toFixed(n)` chuyển sang số có `n` số lẻ ở đằng sau
- `toPrecision(n)` chuyển sang số có chiều dài là `n`
- `valueOf()` lấy giá trị của biến hoặc một giá trị nào đó

Number()

Dùng để chuyển đổi một biến hoặc một giá trị nào đó sang kiểu number, nó chuyển tất cả các định dạng như Boolean, Date, String. Nếu trường hợp giá trị cần chuyển đổi không thể chuyển sang Number được thì nó sẽ chuyển sang giá trị mặc định là **NaN**.

Ví dụ: [XEM DEMO](#)

```
1  var boolean_true = true;
2  Number(boolean_true);           // returns 1
3
4  var boolean_false = false;
5  Number(boolean_false);           // returns 0
6
7  var string_str = 'freetuts.net';
8  Number(string_str);              // returns NaN
9
10 var string_num = '100';
11
12 var date = new Date();
```

```
13  Number(boolean_true);           // returns 1
14
```

parseInt()

Hàm này có tác dụng giống như hàm **Number()**, tuy nhiên có một số điểm khác biệt như sau:

- Nếu chuỗi có các ký tự đầu tiên là các con số và ở đằng sau là chữ cái thì nó sẽ lấy các số đầu tiên đó và chuyển thành kiểu number. Trường hợp này nếu dùng hàm **Number()** thì nó sẽ chuyển thành **NaN**.
- Nếu dữ liệu ở các định dạng khác string thì nó sẽ chuyển thành **NaN**

Ví dụ: [XEM DEMO](#)

```
1      var boolean_true = true;
2      parseInt(boolean_true); // returns NaN
3
4      var boolean_false = false;
5      parseInt(boolean_false); // returns NaN
6
7      var string_str = '10 freetuts.net';
8      parseInt(string_str);    // returns 10
9
10     var string_num = '100';
11     parseInt(string_num);    // returns 100
12
13     var date = new Date();
14     parseInt(boolean_true); // returns NaN
```

parseFloat()

Hàm này chuyển dữ liệu sang định dạng float, về cách sử dụng nó giống với hàm **parseInt()**.

Ví dụ: [XEM DEMO](#)

```
1      var boolean_true = true;
```

```

2  parseFloat(boolean_true);    // returns NaN
3
4  var boolean_false = false;
5  parseFloat(boolean_false);    // returns NaN
6
7  var string_str = '10.2 freetuts.net';
8  parseFloat(string_str);       // returns 10.2
9
10 var string_num = '100';
11 parseFloat(string_num);       // returns 100
12
13 var date = new Date();
14 parseFloat(boolean_true);     // returns NaN

```

2. Hàm xử lý Number cục bộ

Những hàm cục bộ phải gắn liền với đối tượng Number cụ thể, ví dụ khi bạn khởi tạo một biến `var x = 12` thì lúc này `x` có tất cả các hàm cục bộ đó. Ngoài ra bạn có thể sử dụng cặp mở đóng `()` để bao quanh một biểu thức hoặc một giá trị thì vẫn sử dụng bình thường.

Ví dụ:

```

1  var x = 12;
2  x.toString();
3  (12).toString();
4  (12 + 12).toString();

```

toString()

Hàm `toString()` có tác dụng chuyển đổi Number sang kiểu String.

Ví dụ: [XEM DEMO](#)

```

1  var x = 123;

```

```

2
3     typeof x;    // number
4
5     x = x.toString();
6     typeof x;    // string
7
8     typeof 12;   // number
9
10    typeof (12).toString(); // string

```

toFixed(n)

Hàm này có tác dụng chuyển một số sang một số có **n** số lẻ ở sau nó và có làm tròn.

Ví dụ: [XEM DEMO](#)

```

1     var x = 5.656;
2     x.toFixed(0); // returns 6
3     x.toFixed(2); // returns 5.66
4     x.toFixed(4); // returns 5.6560
5     x.toFixed(6); // returns 5.656000

```

toPrecision(n)

Hàm này có tác dụng chuyển một số thành số có chiều dài là **n**, hàm này khác với hàm **toFixed()** ở chỗ hàm **toFixed()** chuyển thành số có **n** số lẻ ở đằng sau. Có một điều lưu ý là tham số **n** phải luôn luôn lớn hơn 0 và nếu bạn không truyền tham số vào thì mặc định nó lấy luôn chiều dài ban đầu.

Ví dụ: [XEM DEMO](#)

```

1     var x = 5.656;
2     x.toPrecision(); // returns 5.656
3     x.toPrecision(2); // returns 5.6
4     x.toPrecision(4); // returns 5.656
5     x.toPrecision(6); // returns 5.65600

```


5

valueOf()

Hàm `valueOf()` có tác dụng lấy giá trị của một biến hoặc một giá trị khác - ý nghĩa là lấy giá trị của (cái gì đó).

Hàm này thực chất không sử dụng nhiều vì thông thường ta lấy giá trị trực tiếp luôn.

Ví dụ: [XEM DEMO](#)

```
1  var x = 123 + 12;
2
3  x.valueOf(); // returns 135
4
5  (2 + 3).valueOf(); // returns 5
```

Ngoài kiểu number ra thì hàm `valueOf()` còn có thể sử dụng được với bất kì các kiểu dữ liệu khác.

Hàm xử lý mảng

1. Danh sách các hàm xử lý mảng trong javascript

Phần này chúng ta sẽ tìm hiểu một số hàm có sẵn trong javascript được tích hợp vào array object, chính vì nó chỉ dùng trong array object nên nếu bạn dùng với kiểu dữ liệu khác sẽ bị sai đấy nhé.

Hàm `array.valueOf()`

Hàm này có tác dụng tương tự như hàm `array.join()` mà ta đã học ở bài trước, có nghĩa là nó sẽ nối các phần tử với nhau vào một chuỗi cách nhau bởi dấu phẩy.

Ví dụ: [XEM DEMO](#)

```
1  var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2  document.write(mang.valueOf());
```

Hàm `array.push()`

Hàm thêm một phần tử vào cuối mảng.

Ví dụ: [XEM DEMO](#)

```
1  var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2
3  // in mảng
4  document.write(mang.valueOf());
5  document.write('<br/>');
6
7  // Thêm và in
8  mang.push("Miễn phí");
9  document.write(mang.valueOf());
```

Hàm `array.pop()`

Ngược với hàm `array.push()`, hàm này có tác dụng xóa đi phần tử cuối cùng trong mảng.

Ví dụ: [XEM DEMO](#)

```
1  var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2
3  // in mảng
4  document.write(mang.valueOf());
5  document.write('<br/>');
6
7  // Thêm và in
8  mang.pop();
9  document.write(mang.valueOf());
```

Hàm `array.shift()`

Hàm xóa phần tử đầu tiên của mảng, sau đó dồn các phần tử phía sau xuống một bậc.

Ví dụ: [XEM DEMO](#)

```

1   var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2
3   // in mảng
4   document.write(mang.valueOf());
5   document.write('<br/>');
6
7   // Thêm và in
8   mang.shift();
9   document.write(mang.valueOf());

```

Hàm array.unshift()

Thêm một phần tử vào vị trí đầu tiên của mảng, đồng thời đẩy các phần tử phía sau lên một bậc.

Ví dụ: [XEM DEMO](#)

```

1   var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2
3   // in mảng
4   document.write(mang.valueOf());
5   document.write('<br/>');
6
7   // Thêm và in
8   mang.unshift("Chào Mừng");
9   document.write(mang.valueOf());

```

Hàm array.splice()

Hàm `splice()` có ba tham số truyền vào như sau: `splice(position_add, num_element_remove, value1, value2, ...)`.

Trong đó:

- **position_add** là vị trí sẽ thêm (vị trí đầu tiên là 0)
- **num_element_remove** là số phần tử sẽ xóa (bắt đầu từ **position_add**)

- **value1, value2, ..** là danh sách các phần tử sẽ được thêm vào sau khi tại vị trí **position_add** và sau khi remove **num_element_remove** phần tử.

Ví dụ 1: [XEM DEMO](#)

```
1  var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2
3  mang.splice(1, 2, 'PHP', 'căn bản ');
4  document.write(mang.valueOf());
```

Trong ví dụ này thì:

- Vị trí thêm là số 1 (phần tử có giá trị là "lập")
- Xóa 2 phần tử liên tiếp từ vị trí 1 (xóa phần tử "lập" và "trình")
- Thêm hai phần tử "php" và "căn bản" vào

Cuối cùng ta có được một mảng gồm ["Học", "php", "căn bản", "tại", "freetuts.net"].

Ví dụ 2: [XEM DEMO](#)

Trong ví dụ này ta sẽ không thêm phần tử nữa mà sẽ lợi dụng hàm này để xóa đi một số phần tử.

```
1  var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2  // xóa phần tử "lập" và "trình"
3  mang.splice(1, 2);
4  document.write(mang.valueOf());
```

Hàm **array.sort()**

Hàm này dùng để sắp xếp các phần tử trong mảng theo thứ tự chữ cái alpha.

Ví dụ: [XEM DEMO](#)

```
1  var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2  document.write(mang.valueOf());
3  document.write('<br/>');
4
5  // Sắp xếp lại
```

```

6   mang.sort();
7   document.write(mang.valueOf());

```

Hàm `array.reverse()`

Hàm đảo ngược các phần tử lại. Vị trí đầu sẽ được chuyển xuống cuối mảng và vị trí cuối mảng sẽ được chuyển lên đầu mảng.

Ví dụ: [XEM DEMO](#)

```

1   var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2   document.write(mang.valueOf());
3   document.write('<br/>');
4
5   // Đảo ngược vị trí của các phần tử
6   mang.reverse();
7   document.write(mang.valueOf());

```

Hàm `array.concat()`

Hàm dùng để nối hai mảng với nhau và trả về một mảng gồm tổng số phần tử của hai mảng đó.

Ví dụ: [XEM DEMO](#)

```
1var mang1 = ["Học", "lập", "trình"];<br>var mang2 = ["tại", "freetuts.net"];<br><br>// Nối
```

Hàm `array.slice()`

Hàm dùng để lấy một số phần tử con trong mảng. Có hai tham số truyền vào như sau: `slice(start, end)`.

Trong đó:

- **start**: là vị trí bắt đầu
- **end**: là vị trí kết thúc

Lưu ý: Để dễ hiểu thì start sẽ phần tử đầu tiên là 0 và end sẽ tính phần tử đầu tiên là 1. Chính vì vậy bạn sẽ phải cộng thêm 1 ở end thì mới lấy đúng phần tử mong muốn

Ví dụ: [XEM DEMO](#)

```

1   var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
2
3   // Lấy phần tử "tại" và "freetuts.net"
4   var mang_moi = mang.slice(3, 5);
5
6   // In ra thử
7   document.write(mang_moi.valueOf());

```

Trường hợp bạn muốn lấy từ vị trí nào đó đến cuối mảng thì bạn sẽ truyền một tham số thôi.

Ví dụ: [XEM DEMO](#)

```

1   <!DOCTYPE html>
2   <html>
3       <body>
4           <script language="javascript">
5               var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];
6
7               // Lấy phần tử "tại" và "freetuts.net"
8               var mang_moi = mang.slice(3);
9
10              // In ra thử
11              document.write(mang_moi.valueOf());
12          </script>
13      </body>
14  </html>

```

Math.nameObject

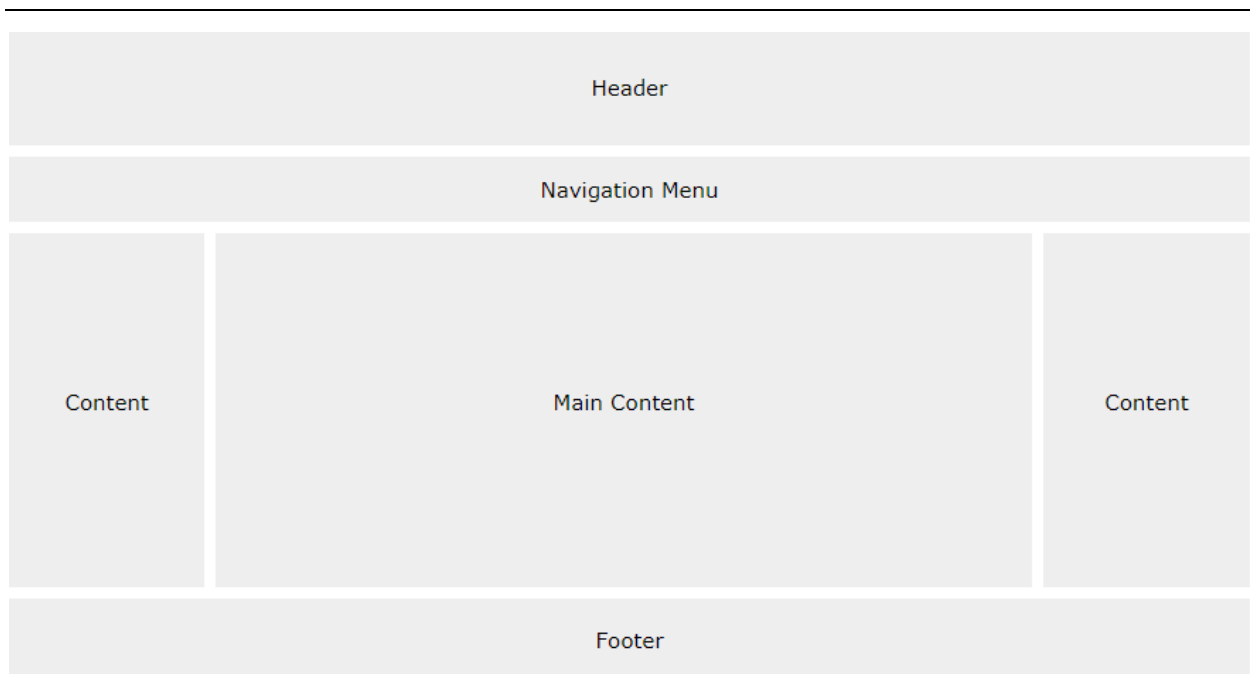
-Sau đây mình xin được liệt kê ra các thuộc tính trong đối tượng **Math** và chức năng của nó.

Thuộc tính	Chức năng
E	Trả về hằng số toán học hay còn gọi là số Euler ($e \sim 2.718$)
LN2	Trả về logarit tự nhiên của 2 (0.693)
LN10	Trả về logarit tự nhiên của 10 (2.302)
LOG2E	Trả về logarit cơ số 2 của E (1.442)
LOG10E	Trả về logarit cơ số 10 của E (0.434)
PI	Trả về hệ số PI (3.14)
SQRT1_2	Trả về căn bậc 2 của 1/2 (0.707)
SQRT2	Trả về căn bậc 2 của 2 (1.414)

-Trong đối tượng Math có hỗ trợ chúng ta 18 phương thức để xử lý tính toán như sau:

Phương Thức	Chức năng
abs(x)	Trả về giá trị tuyệt đối của một số x.
acos(x)	Trả về arccosine của một số x ($-1 \leq x \leq 1$).
asin(x)	Trả về arcsine của một số x ($-1 \leq x \leq 1$).
atan(x)	Trả về arctangent của x.
atan2(x, y)	Trả về hệ số góc của điểm a(x,y) so với trục OX.
ceil(x)	Làm tròn lên số x.
cos(x)	Trả về cos của x.
exp(x)	Trả về e^x .
floor(x)	Làm tròn xuống số x.

log(x)	Trả về logarit e của x.
max(x, y,..., n)	Trả về số lớn nhất trong danh sách các số truyền vào.
min(x, y,..., n)	Trả về số nhỏ nhất trong danh sách các số truyền vào.
pow(x, y)	Trả về kết quả của x^y .
random()	Trả về số ngẫu nhiên trong khoảng từ 0 đến 1.
round(x)	Làm tròn một số về số nguyên.
sin(x)	Trả về sin của x.
sqrt(x)	Trả về căn bậc 2 của x.
tan(x)	Trả về tan của x.




```

<style>
    * {
        box-sizing: border-box;
    }
    [class*="col-"] {
        float: left; //chỉ cần 1 hàng này là đủ, 2 cái dưới k cần
        padding: 10px;
        border: 2px solid bisque;
    }
    .col-1 {width: 8.33%;}
    .col-2 {width: 16.66%;}
    .col-3 {width: 25%;}
    .col-4 {width: 33.33%;}
    .col-5 {width: 41.66%;}
    .col-6 {width: 50%;}
    .col-7 {width: 58.33%;}
    .col-8 {width: 66.66%;}
    .col-9 {width: 75%;}
    .col-10 {width: 83.33%;}
    .col-11 {width: 91.66%;}
    .col-12 {width: 100%;}
    div{
        height: 100px;
    }
</style>

```

Còn lại cứ viết như bt4, thiết kế layout