

BKFA CN-Web Sheet

XML

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE dssk [
<!ELEMENT dssk (sk)+ >
<!ATTLIST dssk ten CDATA #IMPLIED>
<!ELEMENT sk (dsvdv, kq) >
<!ATTLIST sk ten CDATA #REQUIRED>
<!ELEMENT dsvdv (vdv)+ >
<!ELEMENT vdv (vitri)? >
<!ATTLIST vdv ten CDATA #REQUIRED>
<!ATTLIST vdv quocgia CDATA #REQUIRED>
<!ATTLIST vdv ten CDATA #REQUIRED>
<!ELEMENT kq (nhat, nhi, ba) >
<!ELEMENT nhat (vdv)* >
<!ELEMENT nhi (vdv)* >
<!ELEMENT ba (vdv)* >
]>
<dssk>
  <sk ten="dienkinh">
    <dsvdv>
      <vdv ten="ulsanbolt" quocgia="VN">
        </vdv>
      </dsvdv>
    <kq>
      <nhat>
        </nhat>
      <nhi>
        </nhi>
      <ba>
        </ba>
      </kq>
    </sk>
  </dssk>
```

1. Giới thiệu XML

```
<?xml version="1.0" encoding="utf-8"?>
<student>
    <name>Đàm Minh Tiến</name>
    <age>21</age>
    <class>CNTT</class>
    <description>dz</description>
</student>
```

Văn bản có cấu trúc XML cho phép biểu diễn thông tin về các đối tượng trong thực tế. XML dùng để phục vụ cho việc mô tả dữ liệu (thông tin lưu trữ bao gồm những gì, lưu trữ ra sao) để các hệ thống khác nhau có thể đọc và sử dụng những thông tin này một cách thuận tiện. Các thẻ tag ủa XML thường không được định nghĩa trước mà chúng được tạo ra theo quy ước của người theo những quy ước riêng. XML sử dụng các khai báo kiểu dữ liệu DTD (Document Type Definition) hay lược đồ Schema để mô tả dữ liệu.

Định chuẩn XML

Có ba kiểu tài liệu XML là:

- Well-formed Document: tài liệu XML chuẩn cú pháp
- DTD - Constrained Document: Tạo XML có khai bá DTD (Document type definition) để mô tả cấu trúc dữ liệu trong XML
- XML-Schema - Constrained Document: Tạo XML có sử dụng lược đồ Schema để kiểm tra tính hợp lệ của XML.

2. Well-formed XML Document

Các quy ước:

- Có duy nhất một phần tử thuộc cấp cao nhất trong tài liệu, còn gọi là nút gốc
- Mỗi một thẻ mở đều phải có thẻ đóng và tên thẻ là phân biệt hoa thường
- Các thẻ khi đóng phải theo đúng trình tự (mở sau đóng trước)
- Tên thẻ không nên có khoảng trống, không nên bắt đầu bằng “xml”
- Các thuộc tính (attributes) của một thẻ luôn luôn tồn tại theo cặp quy ước `<tên> = “giá trị”`; không nên đặt tên thuộc tính trùng nhau và giá trị của thuộc tính phải đặt trong cặp dấu nháy kép hoặc nháy đơn. Tên của thuộc tính sẽ theo quy luật như tên thẻ.
- Các thẻ (tag) trong XML có thể lồng nhau.

3. DTD (Document Type Definition)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE BIEU_THUC [
  <!ELEMENT BIEU_THUC (PHAN_SO|TICH_SO)+ >
    <!ATTLIST BIEU_THUC Ten CDATA #IMPLIED
      <!-- Ten : A_String -->
    >
  <!ELEMENT PHAN_SO EMPTY >
    <!ATTLIST PHAN_SO
      Tu_so CDATA #REQUIRED
        <!-- Tu_so : A_Int -->
      Mau_so CDATA #REQUIRED
        <!-- Mau_so : A_Int // > 0 -->
    >
  <!ELEMENT TICH_SO(PHAN_SO)+ >
]>
```

Ví dụ đặc tả một biểu thức phân số

DTD định nghĩa các thành phần - thuộc tính và cấu trúc của tài liệu XML.

Các phần trong DTD:

- Elements: các thẻ chính của tài liệu XML
- Attributes: thuộc tính của elements
- Entities: các ký tự có ý nghĩa đặc biệt trong tài liệu XML, ví dụ như “<“ đổi thành <
- PCDATA: các đoạn text sẽ được parser chuyển đổi thành dữ liệu
- CDATA: các đoạn text sẽ được parser bỏ qua

Cú pháp đặc tả chung cấu trúc của 1 thẻ

<!ELEMENT Ten_the Bieu_thuc_dac_ta_cau_truc>

Bieu_thuc sẽ chứa các thẻ con trong Ten_the. Với A, B là 2 thẻ con của thẻ X:

A, B A, B sắp xếp tuần tự A đến B A* A có thể lặp lại ít nhất 0 lần ($>=0$) B+ B có thể lặp lại ít nhất 1 lần ($>=1$) A? A có thể xuất hiện 0 hoặc 1 lần (0 hoặc 1) A|B Có thể sử dụng A hoặc B

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

Ví dụ trên W3S

4. XML Schema (XSD)

XML Schema hay XML Schema Definition miêu tả cấu trúc tài liệu XML 1 cách chi tiết hơn. Ví dụ 1 tài liệu XML

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

XML well-formed

DTD File:

```
<!ELEMENT note (to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

DTD

XSD File:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="https://www.w3schools.com"
xmlns="https://www.w3schools.com"
elementFormDefault="qualified">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

XML Schema File

XML Schema có 3 loại kiểu chính:

- Kiểu định nghĩa sẵn (Builtin Type)
- Kiểu đơn giản (simpleType)
- Kiểu phức hợp (complexType)

Kiểu định nghĩa sẵn:

Được sử dụng khi:

- Thẻ không có thuộc tính
- Thẻ không chứa thẻ khác và có miền giá trị thích hợp với kiểu

Các loại kiểu: string, int, integer, float, double, boolean, date, month, ID, binary
Cú pháp

```
<xs:element name="Ten_the" type="Ten_kieu_co_so" .../><xs:attribute name="Ten_thuoc_tinh" type="Ten_kieu_co_so" .../>
```

Kiểu đơn giản:

Được sử dụng khi:

- Thẻ không có thuộc tính
- Thẻ không chứa thẻ khác và có miền giá trị là tập con của miền giá trị của 1 kiểu cơ sở nào đó

Cú pháp:

```
<xs:simpleType name="Ten_kieu"><xs:restriction base="Ten_kieu_co_so"> Giới hạn trên miền giá trị</xs:restriction></xs:simpleType>
```

Khai báo cận dưới: minInclusive (cho phép sử dụng biên), minExclusive (không sử dụng biên)
Khai báo cận trên: maxInclusive (cho phép sử dụng biên), maxExclusive (không sử dụng biên)

Ngoài ra còn có kiểu liệt kê:

```
<xs:simpleType name="HOC_LUC"><xs:restriction base="string"> <xs:enumeration value="Xuất xác"/> <xs:enumeration value="Giỏi"/></xs:restriction></xs:simpleType>
```

Kiểu phức hợp:

Được sử dụng khi:

- Thẻ có thuộc tính
- Thẻ chứa thẻ khác

Các dạng

- Tuần tự
- Tùy chọn
- Lặp

Tuần tự:

```

<xs:complexType name="Ten_kieu"> <xs:sequence>
    Thanh_phan_1
    Thanh_phan_2
    ...
</xs:sequence>
...
</xs:complexType>

```

Tùy chọn:

```

<xs:complexType name="Ten_kieu"> <xs:choice>
    Thanh_phan_1
    Thanh_phan_2
    ...
</xs:choice>
...
</xs:complexType>

```

Lặp:

```

<xs:complexType name="Ten_kieu"> <xs:sequence> ...
    <xs:element name="Ten_the_con"
        type="Kieu_the_con" minOccurs="So_lan_lap_min"
        maxOccurs="So_lan_lap_max"
    />
    ...
</xs:sequence>
...
</xs:complexType>

```

Thuộc tính của thẻ:

```

<xs:attribute name="Ten_thuoc_tinh" type="Kieu_thuoc_tinh"
    Tu_khoa_1="Gia_tri_1" Tu_khoa_2="Gia_tri_2"/>

```

Một số tính chất thông dụng:

- default: giá trị mặc định
- fixed: giá trị cố định
- use: tùy chọn sử dụng (có hay không)

MVC

[StudentExampleController.php](#)

<?php	include_once("../model/StudentExampleModel.php");
	class StudentController {
	public \$student;
	public function __construct(){
	\$this->student = new Student();
	}
	public function invoke(){
	if (!isset(\$_GET["student"])){
	// if not get specify student -> get all
	\$students = \$this->student->getAll();

```

        include "../view/student/studentlist.php";
    }
else{
    // if get specify student
    $student = $this->student->find($_GET["student"]);
    include "../view/student/viewstudent.php";
}
}
}

?>

```

Model Database

```

<?php
// lớp kết nối và truy xuất cơ sở dữ liệu
class Database {
    public static $hostname = "localhost";
    public static $username = "root";
    public static $password = "";
    public static $dbname = null;
    protected $conn = null;
    public function __construct(){
        $this->connect();
        echo "Set default database!";
    }
    public function connect(){
        $this->conn = mysqli_connect(static::$hostname, static::$username,
static::$password, static::$dbname) or die ('Connect Error!');
        mysqli_query($this->conn, "SET NAMES 'utf8'");
    }
    public function disconnect(){
        if($this->conn)
            mysqli_close($this->conn);
    }
    public static function query($sql){
        try{
            $result = mysqli_query($this->conn, $sql) or die ('Query Error!');
            return $result;
        catch(mysqli_sql_exception $e){
            return false;
        }
    }
?>

```

Model Model

```
<?php
    // lớp chung cho từng bảng dữ liệu
    class Model extends Database{
        public $tableName;
        public $properties = array();
        public $primaryKey;
        public function __construct($tableName){
            parent::__construct();
            $this->tableName = $tableName;
        }
        public getAll(){
            $sql = "SELECT * FROM {$tableName}";
            $result = self::query($sql);
            return fetch($result);
        }
        public function find($pKeyValue){
            $pKey = isset($primaryKey) ? $primaryKey : "id";
            $sql = "SELECT * FROM {$tableName} WHERE {$pKey} = {$pKeyValue}";
            $result = self::query($sql);
            return fetch($result)[0];
        }
        public function findBy($key, $value){
            $sql = "SELECT * FROM {$tableName} WHERE {$key} = {$value}";
            $result = self::query($sql);
            return fetch($result);
        }
        public function findLike($key, $value){
            $sql = "SELECT * FROM {$tableName} WHERE {$key} = %{$value}%";
            $result = self::query($sql);
            return fetch($result);
        }
        public function delete($key, $value){
            $sql = "DELETE FROM {$tableName} WHERE {$key} = {$value}";
            $result = self::query($sql);
            if($result)
                echo "Delete successful!";
            else
                echo "Delete failed!";
        }
        public function fetch($data){
            $result = array();
            if ($data->num_rows > 0) {
                // output data of each row
                while($row = $data->fetch_assoc()) {
                    $result[count($result)] = $row;
                }
            } else {
                echo "0 results";
            }
            return $result;
        }
    }
?>
```

StudentExampleModel.php

```
<?php
//examples class
class Student extends Model{
    public $id;
    public $name;
    public $class;
    public function __construct(){
        parent::__construct();
        echo "Create object without define properties!";
    }
    public function __construct($id, $name, $class){
        $this->id = $id;
        $this->name = $name;
    }
}
```

	<pre> \$this->class = \$class; \$this->primaryKey = isset(\$primaryKey) ? \$primaryKey : "id"; \$this->tableName = \$tableName get_class(\$this); } public function save(){ \$this->primaryKey = isset(\$primaryKey) ? \$primaryKey : "id"; \$this->tableName = isset(\$tableName) ? \$tableName : get_class(\$this); \$sql = "SELECT * FROM {\$tableName} WHERE {\$primaryKey} = {\$this->\$primaryKey}"; \$result = self::query(\$sql); if (\$result->num_rows > 0) { \$sql = "UPDATE {\$tableName} SET id = '{\$id}', name = '{\$name}', class = '{\$class}' WHERE {\$primaryKey} = {\$this->\$primaryKey}"; } else { \$sql = "INSERT INTO {\$tableName} VALUES ('{\$id}', '{\$name}', '{\$class}')"; } \$result = self::query(\$sql); if(\$result) echo "Save successful!"; else echo "Save failed"; } } ?></pre>
--	--

[studentlist.php](#)

<html>	<pre> <head></head> <body> <table> <tbody><tr><td>Title</td><td>Author</td><td>Description</td></tr></tbody> <?php foreach (\$students as \$id => \$student) { echo '<tr><td>id . '">' . \$student->id . '</td><td>' . \$student->name . '</td><td>' . \$student->class . '</td></tr>'; } ?> </table> </body> </html></pre>
--------	---

[viewstudent.php](#)

<html>	<pre> <head></head> <body> <?php echo "Id: {\$student->id}
"; echo "Name: {\$student->name}
"; echo "Class: {\$student->class}
"; ?> </body> </html></pre>
--------	--

config.php

```
<?php
    // config Database
    const $APP_NAME = "PHP-MVC";
    const $AUTHOR = "DMT";
    const $HOST_NAME = "localhost";
    const $USER_NAME = "root";
    const $PASSWORD = "";
    const $DBNAME = null;
?>
```

index.php

```
<?php
    include_once("config.php");
    include_once("model/Database.php");
    include_once("controller/StudentExampleController.php");
    Database::$hostname = $HOST_NAME;
    Database::$username = $USER_NAME;
    Database::$password = $PASSWORD;
    Database::$dbname = $DBNAME;
    $studentController = new StudentController();
    $studentController->invoke();
?>
```

 controller	complete version1	15 hours ago
 model	complete version1	15 hours ago
 views/student	complete version1	15 hours ago
 README.md	complete version1	15 hours ago
 config.php	Update config.php	15 hours ago
 index.php	complete version1	15 hours ago

HTML

1xx: Information

Message:	Description:
100 Continue	The server has received the request headers, and the client should proceed to send the request body
101 Switching Protocols	The requester has asked the server to switch protocols
103 Checkpoint	Used in the resumable requests proposal to resume aborted PUT or POST requests

2xx: Successful

Message:	Description:
200 OK	The request is OK (this is the standard response for successful HTTP requests)
201 Created	The request has been fulfilled, and a new resource is created
202 Accepted	The request has been accepted for processing, but the processing has not been completed
203 Non-Authoritative Information	The request has been successfully processed, but is returning information that may be from another source
204 No Content	The request has been successfully processed, but is not returning any content
205 Reset Content	The request has been successfully processed, but is not returning any content, and requires that the requester reset the document view
206 Partial Content	The server is delivering only part of the resource due to a range header sent by the client

3xx: Redirection

Message:	Description:
300 Multiple Choices	A link list. The user can select a link and go to that location. Maximum five addresses
301 Moved Permanently	The requested page has moved to a new URL
302 Found	The requested page has moved temporarily to a new URL
303 See Other	The requested page can be found under a different URL
304 Not Modified	Indicates the requested page has not been modified since last requested
305 Switch Proxy	<i>No longer used</i>
307 Temporary Redirect	The requested page has moved temporarily to a new URL
308 Resume Incomplete	Used in the resumable requests proposal to resume aborted PUT or POST requests

4xx: Client Error

Message:	Description:
400 Bad Request	The request cannot be fulfilled due to bad syntax
401 Unauthorized	The request was a legal request, but the server is refusing to respond to it. For use when authentication is possible but has failed or not yet been provided
402 Payment Required	<i>Reserved for future use</i>
403 Forbidden	The request was a legal request, but the server is refusing to respond to it
404 Not Found	The requested page could not be found but may be available again in the future
405 Method Not Allowed	A request was made of a page using a request method not supported by that page
406 Not Acceptable	The server can only generate a response that is not accepted by the client
407 Proxy Authentication Required	The client must first authenticate itself with the proxy
408 Request Timeout	The server timed out waiting for the request
409 Conflict	The request could not be completed because of a conflict in the request
410 Gone	The requested page is no longer available
411 Length Required	The "Content-Length" is not defined. The server will not accept the request without it
412 Precondition Failed	The precondition given in the request evaluated to false by the server
413 Request Entity Too Large	The server will not accept the request, because the request entity is too large
414 Request-URI Too Long	The server will not accept the request, because the URL is too long. Occurs when you convert a POST request to a GET request with a long query information
415 Unsupported Media Type	The server will not accept the request, because the media type is not supported
416 Requested Range Not Satisfiable	The client has asked for a portion of the file, but the server cannot supply that portion
417 Expectation Failed	The server cannot meet the requirements of the Expect request-header field

5xx: Server Error

Message:	Description:
500 Internal Server Error	A generic error message, given when no more specific message is suitable
501 Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfill the request
502 Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server
503 Service Unavailable	The server is currently unavailable (overloaded or down)
504 Gateway Timeout	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server
505 HTTP Version Not Supported	The server does not support the HTTP protocol version used in the request
511 Network Authentication Required	The client needs to authenticate to gain network access

CSS

Background Image

```
body {  
    background-image: url("paper.gif");  
}
```

Hình ảnh được lặp lại chỉ theo chiều ngang

```
body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x; //đọc-> background-repeat: repeat-y;  
}
```

Không lặp

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
}
```

Chỉ định vị trí với background-position

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

Xác định hình ảnh cố định, sử dụng thuộc tính background-attachment

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed;  
}
```

Viết gọn trong một thuộc tính background

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

Border

```
p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.groove {border-style: groove;}  
p.ridge {border-style: ridge;}  
p.inset {border-style: inset;}  
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}
```

Kết quả:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

Các kiểu viền khác nhau

Thí dụ

```
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

Khung viền tròn

border-radius: 5px;

Xóa gạch chân thẻ a

```
a {  
    text-decoration: none;  
}
```

This is heading 1

This is heading 2

This is heading 3

```
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}
```

THIS IS SOME TEXT.

this is some text.

This Is Some Text.

```
p.uppercase {  
    text-transform: uppercase;  
}  
  
p.lowercase {  
    text-transform: lowercase;  
}  
  
p.capitalize {  
    text-transform: capitalize;  
}
```

Xác định khoảng cách

This is heading 1

This is heading 2

```
h1 {  
    letter-spacing: 3px;  
}  
  
h2 {  
    letter-spacing: -3px;  
}
```

```
p {  
    direction: rtl;  
}  
  
Đổ bóng text
```

```
h1 {  
    text-shadow: 3px 2px red; //ngang dọc màu  
}
```

Định dạng thẻ a

```
a:link {  
    text-decoration: none;  
}  
  
a:visited {  
    text-decoration: none;  
}  
  
a:hover {  
    text-decoration: underline;  
}  
  
a:active {  
    text-decoration: underline;  
}
```

Đánh mục danh sách

Example of unordered lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola

- a. Coffee
- b. Tea
- c. Coca Cola

```
ul.a {  
    list-style-type: circle;  
}
```

```
ul.b {  
    list-style-type: square;  
}  
  
ol.c {  
    list-style-type: upper-roman;  
}  
  
ol.d {  
    list-style-type: lower-alpha;  
}
```

Dùng ảnh

```
ul {  
    list-style-image: url('sqpurple.gif');  
}
```

Định dạng lại viền bảng

```
table {  
    border-collapse: collapse; // bỏ viền  
}  
  
table, th, td {  
    border: 1px solid black; // viền mới  
}
```

Bảng sọc vằn

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

Vị trí tĩnh

```
div.static {  
    position: static;  
    border: 3px solid #73AD21;  
}
```

Vị trí tương đối

```
div.relative {  
    position: relative;  
    left: 30px;  
    border: 3px solid #73AD21;  
}
```

Vị trí cố định

```
div.fixed {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```

Vị trí tuyệt đối so với thẻ cha, nếu ko có thẻ cha thì nó sẽ lấy phần thân html và di chuyển cùng với việc cuộn trang

```
div.absolute {  
    position: absolute;  
    top: 80px;  
    right: 0;  
    width: 200px;  
    height: 100px;  
    border: 3px solid #73AD21;  
}
```

Vị trí dính (Dính vào top hoặc bên phải, ... của trang web. Vd như fixed menu trên top)

```
div.sticky {  
    position: -webkit-sticky; /* Safari */  
    position: sticky;  
    top: 0;  
    background-color: green;  
    border: 2px solid #4CAF50;  
}
```

So với display: block, sự khác biệt chính là display: inline-block không thêm một ngắt dòng sau phần tử, do đó phần tử có thể ngồi bên cạnh các phần tử khác.

Căn chỉnh ảnh vào giữa

```
img {  
    display: block;  
    margin-left: auto;  
    margin-right: auto;  
    width: 40%;  
}
```

Lấy phần tử liền kề bên dưới thì sử dụng +

```
div + p {  
    background-color: yellow;  
}
```

Lấy phần tử đầu:

```
p:first-child {  
    color: blue;  
}
```

Lấy chữ cái đầu:

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}
```

Trước phần tử

```
h1::before { // sau ::after  
    content: url(smiley.gif);  
}
```

Hiệu ứng ảnh mờ

```
img {  
    opacity: 0.5;  
    filter: alpha(opacity=50); /* For IE8 and earlier */  
}
```

Tạo menu dọc 1 cấp

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    width: 200px;  
    background-color: #f1f1f1;  
}
```

```
li a {  
    display: block;  
    color: #000;  
    padding: 8px 16px;  
    text-decoration: none;  
}
```

```
/* Change the link color on hover */  
li a:hover {  
    background-color: #555;  
    color: white;  
}
```

Tạo menu ngang một cấp

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #333;  
}
```

```
li {  
    float: left;  
}
```

```
li a {  
    display: block;  
    color: white;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
}
```

```
/* Change the link color to #111 (black) on hover */  
li a:hover {  
    background-color: #111;  
}
```

Chuyển dọc thành ngang

```
@media screen and (max-width: 600px){
```

```
    li {
      float: none;
    }

  }
```

Tạo menu ngang 2 cấp

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
  ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
  }
```

```
  li {
```

```
    float: left;
  }
```

```
  li a, .dropbtn {
```

```
    display: inline-block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
  }
```

```
  li a:hover, .dropdown:hover .dropbtn {
```

```
    background-color: red;
  }
```

```
  li.dropdown {
```

```
    display: inline-block;
  }
```

```
}

.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
}

.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
    text-align: left;
}

.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content {
    display: block;
}

</style>
</head>
<body>

<ul>
<li><a href="#home">Home</a></li>
<li><a href="#news">News</a></li>
<li class="dropdown">
    <a href="javascript:void(0)" class="dropbtn">Dropdown</a>
    <div class="dropdown-content">
```

```

<a href="#">Link 1</a>
<a href="#">Link 2</a>
<a href="#">Link 3</a>
</div>
</li>
</ul>

<h3>Dropdown Menu inside a Navigation Bar</h3>
<p>Hover over the "Dropdown" link to see the dropdown menu.</p>

</body>
</html>

```

Using CSS to style an HTML Form

The form consists of three input fields and one button:

- First Name:** An input field with placeholder text "Your name.." and a light gray border.
- Last Name:** An input field with placeholder text "Your last name.." and a light gray border.
- Country:** A select dropdown menu showing "Australia" with a small downward arrow icon.
- Submit:** A large, solid green rectangular button with the word "Submit" centered in white text.

```

<!DOCTYPE html>
<html>
<style>
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
}

```

```
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
}

input[type=submit] {
    width: 100%;
    background-color: #4CAF50;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type=submit]:hover {
    background-color: #45a049;
}

div {
    border-radius: 5px;
    background-color: #f2f2f2;
    padding: 20px;
}

```

</style>

<body>

<h3>Using CSS to style an HTML Form</h3>

<div>

<form action="/action_page.php">

<label for="fname">First Name</label>

<input type="text" id="fname" name="firstname" placeholder="Your name..">

```

<label for="lname">Last Name</label>
<input type="text" id="lname" name="lastname" placeholder="Your last name..">

<label for="country">Country</label>
<select id="country" name="country">
  <option value="australia">Australia</option>
  <option value="canada">Canada</option>
  <option value="usa">USA</option>
</select>

<input type="submit" value="Submit">
</form>
</div>

</body>
</html>

```

Tạo bóng cho thẻ

```

div {
  box-shadow: 10px 10px;
}

```

chuyển tiếp cho cả thuộc tính chiều rộng và chiều cao, với thời lượng là 2 giây cho chiều rộng và 4 giây cho chiều cao:

```

div {
  -webkit-transition: width 2s, height 4s; /* Safari */
  transition: width 2s, height 4s;
}

```

Tạo Tooltip

```

<style>
/* Tooltip container */
.tooltip {
  position: relative;
  display: inline-block;
  border-bottom: 1px dotted black; /* If you want dots under the hoverable text */
}

/* Tooltip text */
.tooltip .tooltiptext {
  visibility: hidden;
  width: 120px;
  background-color: black;
  color: #fff;
  text-align: center;
}

```

```
padding: 5px 0;
border-radius: 6px;

/* Position the tooltip text - see examples below! */
position: absolute;
z-index: 1;
}

/* Show the tooltip text when you mouse over the tooltip container */
.tooltip:hover .tooltiptext {
  visibility: visible;
}
</style>

<div class="tooltip">Hover over me
  <span class="tooltiptext">Tooltip text</span>
</div>

Phân trang

<!DOCTYPE html>

<html>
  <head>
    <style>
      .pagination {
        display: inline-block;
      }

      .pagination a {
        color: black;
        float: left;
        padding: 8px 16px;
        text-decoration: none;
      }

      .pagination a.active {
        background-color: #4CAF50;
        color: white;
        border-radius: 5px;
      }

      .pagination a:hover:not(.active) {
        background-color: #ddd;
      }
    </style>
  </head>
  <body>
    <div class="pagination">
      <a href="#">1</a>
      <a href="#">2</a>
      <a href="#">3</a>
      <a href="#">4</a>
      <a href="#">5</a>
    </div>
  </body>
</html>
```

```
border-radius: 5px;  
}  
</style>  
</head>  
<body>  
  
<h2>Rounded Active and Hover Buttons</h2>  
  
<div class="pagination">  
  <a href="#">&laquo;</a>  
  <a href="#">1</a>  
  <a href="#" class="active">2</a>  
  <a href="#">3</a>  
  <a href="#">4</a>  
  <a href="#">5</a>  
  <a href="#">6</a>  
  <a href="#">&raquo;</a>  
</div>  
  
</body>  
</html>
```

[Back To Top](#)

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
<style>  
  body {  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 20px;  
  }
```

```
#myBtn {  
    display: none;  
    position: fixed;  
    bottom: 20px;  
    right: 30px;  
    z-index: 99;  
    font-size: 18px;  
    border: none;  
    outline: none;  
    background-color: red;  
    color: white;  
    cursor: pointer;  
    padding: 15px;  
    border-radius: 4px;  
}  
  
#myBtn:hover {  
    background-color: #555;  
}  
</style>  
</head>  
<body>  
  
<button onclick="topFunction()" id="myBtn" title="Go to top">Top</button>  
  
<div style="background-color:black;color:white;padding:30px">Scroll Down</div>  
<div style="background-color:lightgrey;padding:30px 30px 2500px">This example demonstrates how to create a "scroll to top" button that becomes visible when the user starts to scroll the page.</div>  
  
<script>  
// When the user scrolls down 20px from the top of the document, show the button  
window.onscroll = function() {scrollFunction()};
```

```
function scrollFunction() {
    if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
        document.getElementById("myBtn").style.display = "block";
    } else {
        document.getElementById("myBtn").style.display = "none";
    }
}

// When the user clicks on the button, scroll to the top of the document
function topFunction() {
    document.body.scrollTop = 0;
    document.documentElement.scrollTop = 0;
}
</script>
</body>
</html>
```

JavaScript

1. Object trong JS

```
var person = {
    firstName: "John",
    lastName : "Doe",
    id       : 5566,
    fullName : function() {
        return this.firstName + " " + this.lastName;
    }
};
```

1.1 Tùy khóa this:

Default Binding:

In a browser the Global object is [**object Window**]: When used in a function, **this** refers to the **Global object**.

Example

```
var x = this;
```

[Try it Yourself »](#)

Example

```
function myFunction() {
    return this;
}
```

In strict mode, **this** will be **undefined**, because strict mode does not allow default binding:

Example

```
"use strict";
function myFunction() {
    return this;
}
```

In these examples, **this** is the **person** object (The person object is the "owner" of the function):

Example

```
var person = {
    firstName : "John",
    lastName  : "Doe",
    id        : 5566,
    myFunction: function() {
        return this;
    }
};
```

Hàm **call()** và **apply()** để gọi hàm cùng tên của đối tượng khác.

```

var person1 = {
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}
var person2 = {
  firstName:"John",
  lastName: "Doe",
}
person1.fullName.call(person2); // Will return "John Doe"

```

2. Phạm vi biến

- Biến toàn cục tự định nghĩa:

```

myFunction();

// code here can use carName

function myFunction() {
  carName = "Volvo";
}

```

3. Các phương thức string

Hàm indexOf(str): trả về vị trí đầu tiên của str trong chuỗi.

Hàm lastIndexOf(str): trả về vị trí cuối cùng của str trong chuỗi.

Hàm search(str): trả về vị trí của str trong chuỗi.

The two methods, indexOf() and search(), are equal?

They accept the same arguments (parameters), and return the same value?

The two methods are NOT equal. These are the differences:

- The search() method cannot take a second start position argument.
- The indexOf() method cannot take powerful search values (regular expressions).

Các hàm cut string:

There are 3 methods for extracting a part of a string:

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

end: vị trí của phần tử cuối + 1

Hàm replace(str1, str2): thay chuỗi str1 = chuỗi str2 trong str ban đầu.

Hàm str.toUpperCase(), str.toLowerCase()

Hàm str.concat(str1, str2)

Hàm charAt(i), charCodeAt()

Hàm split(char) cắt chuỗi theo ký tự char vào mảng

```
var str = "Hello";  
var arr = str.split("");  
⇒ ['H', 'e', 'l', 'l', 'o']
```

4. NaN

```
var x = 100 / "Apple"; // x will be NaN (Not a Number)  
var x = 100 / "10"; // x will be 10  
var x = 100 / "Apple";  
isNaN(x); // true  
  
var x = NaN;  
var y = 5;  
var z = x + y; // z will be NaN  
  
var x = NaN;  
var y = "5";  
var z = x + y; // z will be NaN5  
  
typeof NaN; // returns "number"
```

5. Infinity

```
var myNumber = 2;  
while (myNumber != Infinity) { // Execute until Infinity  
    myNumber = myNumber * myNumber;  
}  
  
var x = 2 / 0; // x will be Infinity  
var y = -2 / 0; // y will be -Infinity  
  
typeof Infinity; // returns "number"
```

6. Number class

```
var x = 500; var x = 500;  
var y = new Number(500); var y = new Number(500);  
  
// (x == y) is true because x and y have equal values // (x === y) is false because x and y have different types  
  
var x = new Number(500);  
var y = new Number(500);  
  
// (x == y) is false because objects cannot be compared
```

7. Array Method

Hàm toString đưa 1 arr thành chuỗi, các phần tử phân cách nhau bởi dấu ,

Hàm join(char) nối arr thành chuỗi, phân cách nhau bởi char

pop() bỏ phần tử cuối cùng, push() thêm vào cuối.

shift() thêm phần tử vào đầu, unshift() xóa phần tử ở đầu.

splice(x,y,str1,str2) thêm str1, str2 vào vị trí x trong mảng và bỏ y phần tử ở vị trí x.

arr1.concat(arr2) nối arr2 vào arr1.

arr1.slice(x,y) cắt arr1 từ x đến y vào arr nguồn.

sort() sắp xếp arr asc, reverser() sắp xếp arr desc.

The Compare Function

The purpose of the compare function is to define an alternative sort order.

The compare function should return a negative, zero, or positive value, depending on the arguments:

```
function(a, b){return a-b}
```

When the sort() function compares two values, it sends the values to the compare function, and sorts the values according to the returned (negative, zero, positive) value.

8. Date

var d = new Date(); lấy time hiện tại

Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)

9. Math

Math.random() trả về 1 số từ 0->1

10. Regexp

Regular Expression Modifiers

Modifiers can be used to perform case-insensitive more global searches:

Modifier	Description
i	Perform case-insensitive matching
g	Perform a global match (find all matches rather than stopping after the first match)
m	Perform multiline matching

Metacharacters are characters with a special meaning:

Metacharacter	Description
\d	Find a digit
\s	Find a whitespace character
\b	Find a match at the beginning or at the end of a word
\uxxxx	Find the Unicode character specified by the hexadecimal number xxxx

Regular Expression Patterns

Brackets are used to find a range of characters:

Expression	Description
[abc]	Find any of the characters between the brackets
[0-9]	Find any of the digits between the brackets
(x y)	Find any of the alternatives separated with

Quantifiers define quantities:

Quantifier	Description
n+	Matches any string that contains at least one n
n*	Matches any string that contains zero or more occurrences of n
n?	Matches any string that contains zero or one occurrences of n

Bootstrap

More Typography Classes

The Bootstrap 4 classes below can be added to style HTML elements further:

Class	Description	Example
<code>.font-weight-bold</code>	Bold text	Try it
<code>.font-italic</code>	Italic text	Try it
<code>.font-weight-light</code>	Light weight text	Try it
<code>.font-weight-normal</code>	Normal text	Try it
<code>.lead</code>	Makes a paragraph stand out	Try it
<code>.small</code>	Indicates smaller text (set to 85% of the size of the parent)	Try it
<code>.text-left</code>	Indicates left-aligned text	Try it
<code>.text-*-right</code>	Indicates left-aligned text on small, medium, large or xl screens	Try it
<code>.text-center</code>	Indicates center-aligned text	Try it
<code>.text-*-center</code>	Indicates center-aligned text on small, medium, large or xl screens	Try it
<code>.text-right</code>	Indicates right-aligned text	Try it
<code>.text-*right</code>	Indicates right-aligned text on small, medium, large or xl screens	Try it
<code>.text-justify</code>	Indicates justified text	Try it
<code>.text-monospace</code>	Monospaced text	Try it
<code>.text-monospace</code>	Monospaced text	Try it
<code>.text nowrap</code>	Indicates no wrap text	Try it
<code>.text-lowercase</code>	Indicates lowercased text	Try it
<code>.text-uppercase</code>	Indicates uppercased text	Try it
<code>.text-capitalize</code>	Indicates capitalized text	Try it
<code>.initialism</code>	Displays the text inside an <code><abbr></code> element in a slightly smaller font size	Try it
<code>.list-unstyled</code>	Removes the default list-style and left margin on list items (works on both <code></code> and <code></code>). This class only applies to immediate children list items (to remove the default list-style from any nested lists, apply this class to any nested lists as well)	Try it
<code>.list-inline</code>	Places all list items on a single line (used together with <code>.list-inline-item</code> on each <code></code> elements)	Try it
<code>.pre-scrollable</code>	Makes a <code><pre></code> element scrollable	Try it

Background Colors

```
<p class="bg-primary text-white">This text is important.</p>
<p class="bg-success text-white">This text indicates success.</p>
<p class="bg-info text-white">This text represents some information.</p>
<p class="bg-warning text-white">This text represents a warning.</p>
<p class="bg-danger text-white">This text represents danger.</p>
<p class="bg-secondary text-white">Secondary background color.</p>
<p class="bg-dark text-white">Dark grey background color.</p>
<p class="bg-light text-dark">Light grey background color.</p>
..
```

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark grey background color.

Table

```
<table class="table table-hover">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>Doe</td>
      <td>john@example.com</td>
    </tr>
    <tr>
      <td>Mary</td>
      <td>Moe</td>
      <td>mary@example.com</td>
    </tr>
    <tr>
      <td>July</td>
      <td>Dooley</td>
      <td>july@example.com</td>
    </tr>
  </tbody>
</table>
```

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Bootstrap 4 Images

```


```

Aligning images

Use the float classes to float the image to the left or to the right:



```

```



 :: căn giữ

Bootstrap 4 Jumbotron

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1>Bootstrap Tutorial</h1>
    <p>Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile-first projects on the web.</p>
  </div>
</div>

<div class="container">
  <p>This is some text.</p>
  <p>This is another text.</p>
</div>
```

Bootstrap Tutorial

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile-first projects on the web.

This is some text.

This is another text.

Bootstrap 4 Alerts

```

<div class="alert alert-success">
  <strong>Success!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
<div class="alert alert-info">
  <strong>Info!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
<div class="alert alert-warning">
  <strong>Warning!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
<div class="alert alert-danger">
  <strong>Danger!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
<div class="alert alert-primary">
  <strong>Primary!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
<div class="alert alert-secondary">
  <strong>Secondary!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
<div class="alert alert-dark">
  <strong>Dark!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
<div class="alert alert-light">

```

Alert Colors

Add the alert-link class to any links inside the alert box to create "matching colored links".

Success! You should [read this message](#).

Info! You should [read this message](#).

Warning! You should [read this message](#).

Danger! You should [read this message](#).

Primary! You should [read this message](#).

Secondary! You should [read this message](#).

Dark! You should [read this message](#).

Bootstrap 4 Buttons

```

<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
<button type="button" class="btn btn-outline-light text-dark">Light</button>

```

Button Outline

[Primary](#) [Secondary](#) [Success](#) [Info](#) [Warning](#) [Danger](#)

[Dark](#) [Light](#)

```

<button type="button" class="btn btn-primary btn-lg">Large</button>
<button type="button" class="btn btn-primary btn-md">Default</button>
<button type="button" class="btn btn-primary btn-sm">Small</button>

```

Button Sizes

[Large](#) [Default](#) [Small](#)

```

<button type="button" class="btn btn-primary">Primary Button</button>
<button type="button" class="btn btn-primary active">Active Primary</button>
<button type="button" class="btn btn-primary" disabled>Disabled Primary</button>
<a href="#" class="btn btn-primary disabled">Disabled Link</a>

```

Button States

[Primary Button](#) [Active Primary](#) [Disabled Primary](#)

[Disabled Link](#)

```

<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>

```

Button Group

The .btn-group class creates a button group:

[Apple](#) [Samsung](#) [Sony](#)

```

<div class="btn-group btn-group-lg">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>
<h3>Default Buttons:</h3>
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>
<h3>Small Buttons:</h3>
<div class="btn-group btn-group-sm">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>

```

```

<div class="btn-group-vertical">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>

```

```

<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <div class="btn-group">
    <button type="button" class="btn btn-primary dropdown-toggle" data-
    toggle="dropdown">
      Sony
    </button>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Tablet</a>
      <a class="dropdown-item" href="#">Smartphone</a>
    </div>
  </div>
</div>

```

Button Groups Sizes

Add class .btn-group-* to size all buttons in a button group.

Large Buttons:

Apple Samsung Sony

Default Buttons:

Apple Samsung Sony

Small Buttons:

Apple Samsung Sony

Vertical Button Group

Use the .btn-group-vertical class to create a vertical button group:

Apple
Samsung
Sony

Nesting Button Groups

Nest button groups to create dropdown menus:

Apple Samsung Sony ▾
Tablet
Smartphone

Bootstrap 4 Badges

```

<span class="badge badge-primary">Primary</span>
<span class="badge badge-secondary">Secondary</span>
<span class="badge badge-success">Success</span>
<span class="badge badge-danger">Danger</span>
<span class="badge badge-warning">Warning</span>
<span class="badge badge-info">Info</span>
<span class="badge badge-light">Light</span>
<span class="badge badge-dark">Dark</span>

```

```

<span class="badge badge-pill badge-primary">Primary</span>
<span class="badge badge-pill badge-secondary">Secondary</span>
<span class="badge badge-pill badge-success">Success</span>
<span class="badge badge-pill badge-danger">Danger</span>
<span class="badge badge-pill badge-warning">Warning</span>
<span class="badge badge-pill badge-info">Info</span>
<span class="badge badge-pill badge-light">Light</span>
<span class="badge badge-pill badge-dark">Dark</span>

```

Contextual Badges

Primary Secondary Success Danger Warning Info Light Dark

Pill Badges

Primary Secondary Success Danger Warning Info Light Dark

```
<button type="button" class="btn btn-primary">
  Messages <span class="badge badge-light">4</span>
</button>
<button type="button" class="btn btn-danger">
  Notifications <span class="badge badge-light">7</span>
</button>
```

Badge inside a Button

Messages 4 Notifications 7

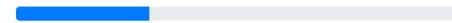
Basic Progress Bar

```
<div class="progress">
  <div class="progress-bar" style="width:30%"></div>
</div>
```

```
<div class="progress">
  <div class="progress-bar" style="width:70%">70%</div>
</div>
```

Basic Progress Bar

To create a default progress bar, add a .progress class to a container element and add the progress-bar class to its child element. Use the CSS width property to set the width of the progress bar:



Progress Bar With Label

70%

```
<div class="progress">
  <div class="progress-bar bg-success" style="width:40%">
    Free Space
  </div>
  <div class="progress-bar bg-warning" style="width:10%">
    Warning
  </div>
  <div class="progress-bar bg-danger" style="width:20%">
    Danger
  </div>
```

Multiple Progress Bars

Create a stacked progress bar by placing multiple bars into the same div with class="progress":



Bootstrap 4 Pagination

```
<ul class="pagination">
  <li class="page-item"><a class="page-link" href="#">Previous</a></li>
  <li class="page-item"><a class="page-link" href="#">1</a></li>
  <li class="page-item"><a class="page-link" href="#">2</a></li>
  <li class="page-item"><a class="page-link" href="#">3</a></li>
  <li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
```

Pagination

To create a basic pagination, add the .pagination class to an ul element. Then add the .page-item to each li element and a .page-link class to each link inside li:

Previous 1 2 3 Next

```
<ul class="breadcrumb">
  <li class="breadcrumb-item"><a href="#">Photos</a></li>
  <li class="breadcrumb-item"><a href="#">Summer 2017</a></li>
  <li class="breadcrumb-item"><a href="#">Italy</a></li>
  <li class="breadcrumb-item active">Rome</li>
</ul>
```

Breadcrumbs

The .breadcrumb class indicates the current page's location within a navigational hierarchy:

Photos / Summer 2017 / Italy / Rome

```

<ul class="pagination pagination-lg">
  <li class="page-item"><a class="page-link" href="#">Previous</a></li>
  <li class="page-item"><a class="page-link" href="#">1</a></li>
  <li class="page-item"><a class="page-link" href="#">2</a></li>
  <li class="page-item"><a class="page-link" href="#">3</a></li>
  <li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>

</>Default:</p>
<ul class="pagination">
  <li class="page-item"><a class="page-link" href="#">Previous</a></li>
  <li class="page-item"><a class="page-link" href="#">1</a></li>
  <li class="page-item"><a class="page-link" href="#">2</a></li>
  <li class="page-item"><a class="page-link" href="#">3</a></li>
  <li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>

<p>Small:</p>
<ul class="pagination pagination-sm">
  <li class="page-item"><a class="page-link" href="#">Previous</a></li>
  <li class="page-item"><a class="page-link" href="#">1</a></li>
  <li class="page-item"><a class="page-link" href="#">2</a></li>
  <li class="page-item"><a class="page-link" href="#">3</a></li>
  <li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>

```

Pagination - Sizing

Add class .pagination-lg for larger blocks or .pagination-sm for smaller blocks.

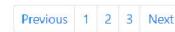
Large:



Default:



Small:



Bootstrap 4 List Groups

```

<h2>Basic List Group</h2>
<ul class="list-group">
  <li class="list-group-item">First item</li>
  <li class="list-group-item">Second item</li>
  <li class="list-group-item">Third item</li>
</ul>

```

Basic List Group

First item
Second item
Third item

```

<ul class="list-group">
  <li class="list-group-item list-group-item-success">Success item</li>
  <li class="list-group-item list-group-item-secondary">Secondary item</li>
  <li class="list-group-item list-group-item-info">Info item</li>
  <li class="list-group-item list-group-item-warning">Warning item</li>
  <li class="list-group-item list-group-item-danger">Danger item</li>
  <li class="list-group-item list-group-item-primary">Primary item</li>
  <li class="list-group-item list-group-item-dark">Dark item</li>
  <li class="list-group-item list-group-item-light">Light item</li>
</ul>

```

List Group With Contextual Classes

Success item
Secondary item
Info item
Warning item
Danger item
Primary item
Dark item
Light item

Bootstrap 4 Dropdowns

```

<div class="dropdown">
  <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">
    Dropdown button
  </button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Link 1</a>
    <a class="dropdown-item" href="#">Link 2</a>
    <a class="dropdown-item" href="#">Link 3</a>
  </div>
</div>

```

Dropdowns

The .dropdown class is used to indicate a dropdown menu.

Use the .dropdown-menu class to actually build the dropdown menu.

To open the dropdown menu, use a button or a link with a class of .dropdown-toggle and data-toggle="dropdown".



```

<div class="dropdown">
  <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">
    Dropdown button
  </button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Link 1</a>
    <a class="dropdown-item" href="#">Link 2</a>
    <a class="dropdown-item" href="#">Link 3</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="#">Another link</a>
  </div>
</div>

```

Dropdowns

The `.dropdown-divider` class is used to separate links inside the dropdown menu with a thin horizontal line:



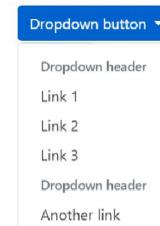
```

<div class="dropdown">
  <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">
    Dropdown button
  </button>
  <div class="dropdown-menu">
    <h5 class="dropdown-header">Dropdown header</h5>
    <a class="dropdown-item" href="#">Link 1</a>
    <a class="dropdown-item" href="#">Link 2</a>
    <a class="dropdown-item" href="#">Link 3</a>
    <h5 class="dropdown-header">Dropdown header</h5>
    <a class="dropdown-item" href="#">Another link</a>
  </div>
</div>

```

Dropdowns

The `.dropdown-header` class is used to add headers inside the dropdown menu:



Dropdown Position

Dropright ▶

◀ **Dropleft**

You can also create a "dropright" or "dropleft" menu, by adding the `.dropright` or `.dropleft` class to the dropdown element. Note that the caret/arrow is added automatically:

Dropright

```
<div class="dropdown dropright">
```

[Try it Yourself »](#)

Dropleft

```
<div class="dropdown dropleft">
```

[Try it Yourself »](#)

Dropdown Menu Right

Wide dropdown button to demonstrate this example ▾

To right-align the dropdown menu, ad-

Link 1
Link 2
Link 3

menu-right class to the element with `.dropdown-menu`:

Example

```
<div class="dropdown-menu dropdown-menu-right">
```

Dropup

Dropdown button ▾

If you want the dropdown menu to expand upwards instead of downwards, change the `<div>` element with `class="dropdown"` to `"dropup"`:

Example

```
<div class="dropup">
```

Dropdown Text

Dropdown button ▾

Link 1 em-text class is used to add plain text to a dropdown item, or used on links for default link styling.
Link 2
Text Link
Just Text

```
<div class="dropdown-menu">  
  <a class="dropdown-item" href="#">Link 1</a>  
  <a class="dropdown-item" href="#">Link 2</a>  
  <a class="dropdown-item-text" href="#">Text Link</a>  
  <span class="dropdown-item-text" href="#">Just Text</span>  
</div>
```

Grouped Buttons with a Dropdown

Apple Samsung Sony ▾

Example

Tablet

Smartphone

```
<div class="btn-group">  
  <button type="button" class="btn btn-primary">Apple</button>  
  <button type="button" class="btn btn-primary">Samsung</button>  
  <div class="btn-group">  
    <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">  
      Sony  
    </button>  
    <div class="dropdown-menu">  
      <a class="dropdown-item" href="#">Tablet</a>  
      <a class="dropdown-item" href="#">Smartphone</a>  
    </div>  
  </div>  
</div>
```

Bootstrap 4 Navs

```

<li class="nav-item">
  <a class="nav-link active" href="#">Active
</li>
<li class="nav-item">
  <a class="nav-link" href="#">Link
</li>
<li class="nav-item">
  <a class="nav-link" href="#">Link
</li>
<li class="nav-item">
  <a class="nav-link disabled" href="#">Disabled
</li>
</ul><br>

<p>Justified tabs:</p>
<ul class="nav nav-tabs nav-justified">
<li class="nav-item">
  <a class="nav-link active" href="#">Active
</li>
<li class="nav-item">
  <a class="nav-link" href="#">Link
</li>
<li class="nav-item">
  <a class="nav-link" href="#">Link
</li>
<li class="nav-item">
  <a class="nav-link disabled" href="#">Disabled
</li>
</ul>

```

Justified Tabs/Pills

To justify tabs/pills (equal width), use the .nav-justified class:

Active	Link	Link	Disabled
--------	------	------	----------

Justified tabs:

Active	Link	Link	Disabled
--------	------	------	----------

Bootstrap 4 Navigation Bar

```

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <!-- Brand -->
  <a class="navbar-brand" href="#">Logo</a>

  <!-- Links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">Link 1
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 2
    </li>
  </ul>

  <!-- Dropdown -->
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-
    toggle="dropdown">
      Dropdown link
    </a>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Link 1</a>
      <a class="dropdown-item" href="#">Link 2</a>
      <a class="dropdown-item" href="#">Link 3</a>
    </div>
  </li>
</ul>
</nav>

```



Bootstrap 4 Forms

```

<div class="form-group">
  <label for="email">Email:</label>
  <input type="email" class="form-control" id="email" placeholder="Enter email"
  name="email">
</div>
<div class="form-group">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd" placeholder="Enter password"
  name="pwd">
</div>
<div class="form-group form-check">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" name="remember"> Remember me
  </label>
</div>
<button type="submit" class="btn btn-primary">Submit</button>

```

Stacked form

Email:

Password:

Remember me

Submit

Bootstrap 4 Form Inputs

```
<div class="form-group">
  <label for="usr">Name:</label>
  <input type="text" class="form-control" id="usr" name="username">
</div>
<div class="form-group">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd" name="password">
</div>
<button type="submit" class="btn btn-primary">Submit</button>
```

Form control: input

The form below contains two input elements; one of type text and one of type password:

Name:

Password:

Submit

Bootstrap 4 Carousel

```
<div id="demo" class="carousel slide" data-ride="carousel">
  <ul class="carousel-indicators">
    <li data-target="#demo" data-slide-to="0" class="active"></li>
    <li data-target="#demo" data-slide-to="1"></li>
    <li data-target="#demo" data-slide-to="2"></li>
  </ul>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption">
        <h3>Los Angeles</h3>
        <p>We had such a great time in LA!</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption">
        <h3>Chicago</h3>
        <p>Thank you, Chicago!</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption">
        <h3>New York</h3>
        <p>We love the Big Apple!</p>
      </div>
    </div>
  </div>
  <a class="carousel-control-prev" href="#demo" data-slide="prev">
    <span class="carousel-control-prev-icon"></span>
  </a>
  <a class="carousel-control-next" href="#demo" data-slide="next">
    <span class="carousel-control-next-icon"></span>
  </a>
</div>
```



Class	Description
.carousel	Creates a carousel
.carousel-indicators	Adds indicators for the carousel. These are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user are currently viewing)
.carousel-inner	Adds slides to the carousel
.carousel-item	Specifies the content of each slide
.carousel-control-prev	Adds a left (previous) button to the carousel, which allows the user to go back between the slides
.carousel-control-next	Adds a right (next) button to the carousel, which allows the user to go forward between the slides
.carousel-control-prev-icon	Used together with .carousel-control-prev to create a "previous" button
.carousel-control-next-icon	Used together with .carousel-control-next to create a "next" button
.slide	Adds a CSS transition and animation effect when sliding from one item to the next. Remove this class if you do not want this effect

Bootstrap 4 Modal

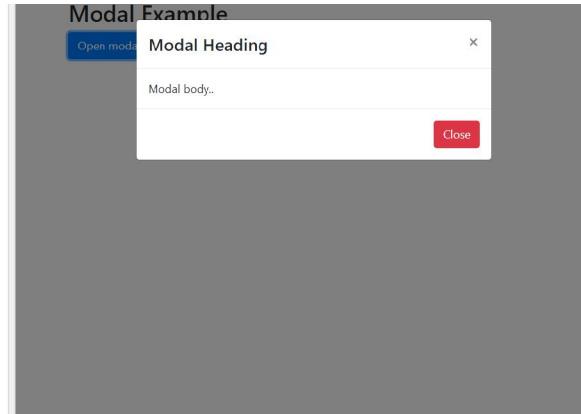
```
<!-- Button to Open the Modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal">
  Open modal
</button>

<!-- The Modal -->
<div class="modal" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">

      <!-- Modal Header -->
      <div class="modal-header">
        <h4 class="modal-title">Modal Heading</h4>
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>

      <!-- Modal body -->
      <div class="modal-body">
        Modal body...
      </div>

      <!-- Modal footer -->
      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```



Bootstrap 4 Tooltip

The screenshot shows a browser window with the title "Tooltip Example". The page content includes a container with several links. One link has a tooltip with the text "Hover me!". The browser's status bar shows the URL "https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_tooltip_pos&stacked=1".

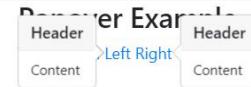
Bootstrap 4 Popover

```

<a href="#" title="Header" data-toggle="popover" data-placement="top" data-
contents="Content">Top</a>
<a href="#" title="Header" data-toggle="popover" data-placement="bottom" data-
contents="Content">Bottom</a>
<a href="#" title="Header" data-toggle="popover" data-placement="left" data-
contents="Content">Left</a>
<a href="#" title="Header" data-toggle="popover" data-placement="right" data-
contents="Content">Right</a>
</div>

<script>
$(document).ready(function(){
  $('[data-toggle="popover"]').popover();
});
</script>

```



Bootstrap 4 Utilities

```

<span class="rounded"></span>
<span class="rounded-top"></span>
<span class="rounded-right"></span>
<span class="rounded-bottom"></span>
<span class="rounded-left"></span>
<span class="rounded-circle"></span>
<span class="rounded-0"></span>

```

Borders

Round the corner of an element with the rounded classes:



Bootstrap 4 Flex

```

<div class="d-flex flex-column mb-3">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>


Use .flex-column-reverse to reverse the vertical direction:


<div class="d-flex flex-column-reverse">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>

```

Vertical Direction

Use .flex-column to display the flex items vertically (on top of each other):



Use .flex-column-reverse to reverse the vertical direction:



Bootstrap 4 Grid System

Bootstrap's grid system allows up to 12 columns across the page.

If you do not want to use all 12 column individually, you can group the columns together to create wider columns:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1				
span 4				span 4				span 4							
span 4				span 8											
span 6						span 6									
span 12															

Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.

Grid Classes

The Bootstrap 4 grid system has five classes:

- `.col-` (extra small devices - screen width less than 576px)
- `.col-sm-` (small devices - screen width equal to or greater than 576px)
- `.col-md-` (medium devices - screen width equal to or greater than 768px)
- `.col-lg-` (large devices - screen width equal to or greater than 992px)
- `.col-xl-` (xlarge devices - screen width equal to or greater than 1200px)

	Extra small (<576px)	Small (>=576px)	Medium (>=768px)	Large (>=992px)	Extra Large (>=1200px)
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Grid behaviour	Horizontal at all times	Collapsed to start, horizontal above breakpoints			
Container width	None (auto)	540px	720px	960px	1140px
Suitable for	Portrait phones	Landscape phones	Tablets	Laptops	Laptops and Desktops
# of columns	12	12	12	12	12
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes	Yes

Bootstrap JS Tab

```

<!-- Nav tabs -->


- Home
- Menu 1
- Menu 2



<!-- Tab panes -->


<br>
<h3>HOME</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>



<br>
<h3>Menu 1</h3>
<p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>



<br>
<h3>Menu 2</h3>
<p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam.</p>


```

Toggleable Tabs

Home Menu 1 Menu 2

HOME

lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

AJAX

AJAX - Asynchronous JavaScript And XML.

Ajax PHP

Tạo file index.php để xử lý ajax

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script language="javascript">
      function load_ajax(){
        alert('Clicked');
      }
    </script>
  </head>
  <body>
    <div id="result">
      Nội dung ajax sẽ được load ở đây
    </div>
    <input type="button" name="clickme" id="clickme" onclick="load_ajax()" value="Click Me"/>
  </body>
</html>
```

Trong đó tôi đã tạo một thẻ `div` dùng để chứa nội dung **load ajax**, một button trong đó có sự kiện **onclick**, sự kiện này sẽ gọi đến hàm **load_ajax**.

Tạo file trả về nội dung ajax

```
<?php
echo 'Đây là nội dung trả về';
?>
```

Viết code gọi ajax lấy nội dung từ file result

Viết hàm xử lý **gọi ajax** để lấy nội dung từ file `result.php` và gán nội dung đó vào thẻ `div` như khai báo ở file `index.php` trên

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script language="javascript">
      function load_ajax()
      {
        // Tạo một biến lưu trữ đối tượng XML HTTP. Đối tượng này
        // tùy thuộc vào trình duyệt browser ta sử dụng nên phải kiểm
        // tra như bước bên dưới
        var xmlhttp;

        // Nếu trình duyệt là IE7+, Firefox, Chrome, Opera, Safari
        if (window.XMLHttpRequest)
```

```

{
    xmlhttp = new XMLHttpRequest();
}
// Nếu trình duyệt là IE6, IE5
else
{
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}

// Khởi tạo một hàm gửi ajax
xmlhttp.onreadystatechange = function()
{
    // Nếu đối tượng XML HTTP trả về với hai thông số bên dưới thì mọi chuyện
    // coi như thành công
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
    {
        // Sau khi thành công tiến hành thay đổi nội dung của thẻ div, nội dung
        // ở đây chính là
        document.getElementById("result").innerHTML = xmlhttp.responseText;
    }
};

// Khai báo với phương thức GET, và url chính là file result.php
xmlhttp.open("GET", "result.php", true);

// Cuối cùng là Gửi ajax, sau khi gọi hàm send thì function vừa tạo ở
// trên (onreadystatechange) sẽ được chạy
xmlhttp.send();
}
</script>
</head>
<body>
<div id="result">
    Nội dung ajax sẽ được load ở đây
</div>
<input type="button" name="clickme" id="clickme" onclick="load_ajax()" value="Click Me"/>
</body>
</html>

```

Ajax JS

[XMLHttpRequest](#)

Để tạo một đối tượng XMLHttpRequest:

```
var xhttp = new XMLHttpRequest();
```

Các phương thức phổ biến của lớp XMLHttpRequest:

- **new XMLHttpRequest()**: khởi tạo một đối tượng XMLHttpRequest
- **abort()**: hủy http request hiện tại
- **getAllResponseHeaders()**: trả về thông tin header http request
- **open(*method, url, async, user, psw*)**: tạo 1 http request, với các tham số: method - GET hoặc POST, url - vị trí lưu file, async - true không đồng bộ false đồng bộ, user - tên người dùng, psw - mật khẩu.
- **send()**: gửi request lên server (thường dùng với phương thức GET)
- **send(*string*)**: gửi request lên server (thường dùng với phương thức POST)

Các thuộc tính của lớp XMLHttpRequest:

- onreadystatechange: định nghĩa một hàm sẽ được gọi khi thuộc tính readyState thay đổi
- **readyState** : trạng thái của HTTP request. 0: request chưa được khởi tạo, 1: request đã khởi tạo, 2: request đã nhận được bởi server, 3: request đang trong thời gian xử lý, 4: request đã hoàn thành và dữ liệu cần đã được trả về.
- responseText: lưu dữ liệu trả về dưới dạng chuỗi
- responseXML: lưu dữ liệu trả về dưới dạng XML
- **status** : trạng thái của HTTP request - dạng số (ví dụ 200 , 403, 404)
- statusText: trạng thái của HTTP request - dạng text (ví dụ "OK", "Forbidden", "Not Found")

Ví dụ:

Get XML file:

```
function loadDoc() {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            myFunction(this);  
        }  
    };  
    xhttp.open("GET", "cd_catalog.xml", true);  
    xhttp.send();  
}  
  
function myFunction(xml) {  
    var i;  
    var xmlDoc = xml.responseXML;  
    var table = "<tr><th>Artist</th><th>Title</th></tr>";  
    var x = xmlDoc.getElementsByTagName("CD");  
    for (i = 0; i < x.length; i++) {  
        table += "<tr><td>" +  
    }
```

```
x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +  
"</td><td>" +  
x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +  
"</td></tr>";  
}  
document.getElementById("demo").innerHTML = table;  
}
```

Post dữ liệu:

```
xhttp.open("POST", "ajax_test.asp", true);  
 xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
 xhttp.send("fname=Henry&lname=Ford");
```