

JS

Uploaded using RayThis

```
function fn1(){
    setTimeout(function(){
        console.log(1);
    }, 500)
}
function fn2(){
    console.log(2);
}
fn1();
fn2();

//2
//1
```

```
str = "ABCDEFGHIJKLMNPQRSTUVWXYZ";
console.log("length: " + str.length); // length: 26
console.log(str.indexOf("C")); // 2
console.log("ABACABA".lastIndexOf("A")); // 4
console.log("Hello World".slice(6, 11)); //World
console.log("Hello World".substr(6, 5)); //World
console.log("Apple, Banana, Kiwi".slice(-12, -6)); //Banana
console.log("ABC".toLowerCase()); //abc
console.log(" THUc tron ".trim()); //THUc tron
console.log("ABC".charAt(2)); //C
console.log("ABC".charCodeAt(2)); //67 (C)
console.log("A, B, C".split(",")); // ["A", "B", "C"]
console.log("Hello".replace("Hello", "World")); //World
```

Uploaded using RayThis Extension

```
var fruit = ["apple", "banana", "cherry"];
fruit.forEach(function(e, i){
    console.log(i + ": " + e );
});
//0: apple
//1: banana
//2: cherry
```

Uploaded using RayThis Extension

```
//Khai báo hàm
function sum1(a, b){
    return a + b;
}
var sum2 = (a, b) => {return a + b};

console.log(sum1(5, 7)); // 12
console.log(sum2(10, 20)); // 30
```

```
function multiply(a, b) {b = (typeof b !==
'undefined')? b : 1    return a * b}
```

Uploaded using RayThis Extension

```
function sum(){
    let total = 0;
    for(let i = 0; i < arguments.length; i++){
        total += arguments[i];
    }
    return total;
}
var t = sum(1,2,3);
console.log(t); //6
```

Uploaded using RayThis Extension

```
var num = 0;
console.log(num);
switch(num){
    case 0:
        alert("Less than 0.");
        break;
    case 1:
        alert("Equal to 0.");
        break;
    case 2:
        alert("Greater than 0.");
        break;
    default:
        alert("Unknown.");
}
```

Uploaded using RayThis Extension

```
//Khai báo hàm
function sum1(a, b){
    return a + b;
}
var sum2 = (a, b) => {return a + b};

console.log(sum1(5, 7)); // 12
console.log(sum2(10, 20)); // 30

function logger(msg){
    console.log(msg);
    console.log(arguments[0]);
}
logger("Hello", "World"); // Hello World
```

Uploaded using RayThis Extension

```
function sum(...args) {  
    let total = 0;  
    for (let i in args) total += args[i];  
    return total;  
}  
  
var t = sum(1, 2, 3);  
console.log(t); //6
```

Uploaded using RayThis Extension

```
function fun1(str){  
    console.log("Hello " + str);  
}  
  
function fun2(str, myfunc){  
    console.log("Welcome");  
    myfunc(str);  
}  
  
fun2("World", fun1); // Welcome Hello World
```

Uploaded using RayThis Ext

```
function fun2(str, myfunc){  
    console.log("Welcome");  
    myfunc(str);  
}  
  
fun2("World", function(str){  
    console.log("Hello " + str);  
}); // Welcome Hello World
```

Uploaded using RayThis Extension

```
var myCar = new Object();  
myCar.make = "Toyota";  
myCar.model = "Camry";  
  
var object = {  
    name: "John Doe",  
    age: 30,  
    city: "New York",  
    printDetails: function () {  
        console.log("Name: " + this.name);  
        console.log("Age: " + this.age);  
        console.log("City: " + this.city);  
    },  
};  
  
console.log(object.name); //John Doe  
console.log(object.printDetails()); //Name: John Doe Age: 30 City: New York  
  
var profile2 = Object.create(object);  
console.log(profile2.name); //John Doe  
console.log(profile2.printDetails()); //Name: John Doe Age: 30 City: New York  
  
function Profile(name, age, city) {  
    this.name = name;  
    this.age = age;  
    this.city = city;  
    this.printDetails = function () {  
        console.log("Name: " + this.name);  
        console.log("Age: " + this.age);  
        console.log("City: " + this.city);  
    };  
}  
var profile1 = new Profile("John", 10, "New York");  
profile1.printDetails(); //Name: John Age: 10 City: New York  
profile1.name = "Phu"; //Cach 1  
profile1["name"] = "John"; //Cach 2  
Object.defineProperty(profile1, "name", { value: "John", writable: true }); //Cach 3
```

Uploaded using RayThis Extension

```
console.log(Math.PI);  
// π = 3.14159  
console.log(Math.abs(-10));  
// |-10| = 10  
console.log(Math.acos(1))  
//acos(1) = 0;  
console.log(Math.ceil(1.1))  
//trà về số lớn hơn hoặc bằng x 2  
console.log(Math.floor(1.1))  
//trà về số lớn hơn hoặc bằng x 0  
console.log(Math.log(10))  
//log4 = ?  
console.log(Math.max(1, 2, 3, 4, 5));  
//max(1, 2, 3, 4, 5) = 5  
console.log(Math.min(1, 2, 3, 4, 5));  
//min(1, 2, 3, 4, 5) = 1  
console.log(Math.pow(2, 3));  
//pow(2, 3) = 8  
console.log(Math.random());  
//random number between 0 and 1  
console.log(Math.round(12.5)); //13  
console.log(Math.sqrt(16));  
// sqrt(16) = 4
```

Uploaded using RayThis Extension

```
var obj = new Date();  
  
console.log(obj.getDate()); //Trả về giá trị ngày (1-31)  
console.log(obj.getDay()); //Trả về giá trị ngày trong tuần (0-6) 0 là ngày chủ nhật  
console.log(obj.getMonth()); //Trả về giá trị tháng trong năm (0-11) 0 là tháng 1  
console.log(obj.getFullYear()); //Trả về năm (4 số)  
console.log(obj.getYear()); //Trả về năm (2 số)  
console.log(obj.getMinutes());  
console.log(obj.getHours()); //Trả về giờ của hệ thống (0-23)  
console.log(obj.getSeconds());  
console.log(obj.getHours());
```



Uploaded using RayThis Extension

```
const cars = new Array("Phantom", "John", "BMW");
const cars1 = ["Phantom1", "John1", "BMW1"];
var cars2 = cars.concat(cars1); // ["Phantom", "John", "BMW", "Phantom1", "John1", "BMW1"]
var cars3 = cars.join("+"); // Phantom+John+BMW
var cars4 = cars.slice(1, 3); // index bắt đầu từ 1 // ["John", "BMW"]
var cars5 = cars.reverse(); // ["BMW", "John", "Phantom"]
var cars6 = cars.values();
const cars6Array = [...cars6]; // Store values in an array
console.log(cars6Array); // ["BMW", "John", "Phantom"]
for (let car of cars6Array) {
    console.log(car);
}
// Kết quả:
// Phantom
// John
// BMW
var cars7 = cars.pop(); // ["BMW", "John"]
cars.push("John2"); // ["BMW", "John", "John2"]
var car8 = cars.shift(); // "BMW" cars = ["John", "John2"]
cars.sort();
```

DOM



Uploaded using RayThis Extension

```
console.log(document.baseURI); //địa chỉ trang web
console.log(document.body); //nội dung thẻ body
console.log(document.cookie); //trả về cookie
console.log(document.doctype); //trả về phần tử doctype
console.log(document.documentElement); //trả về phần tử <html> root element
console.log(document.domain); //trả về tên miền của trang web
console.log(document.forms); //trả về tên miền của trang web
console.log(document.head); //trả về phần tử <head>
console.log(document.images); //trả về các ảnh trong trang web
console.log(document.inputEncoding); //tải về dạng mã hóa của trang
console.log(document.location); //trả về địa chỉ hiện tại của trang web
console.log(document.links); //trả về các đường dẫn trong trang web
console.log(document.readyState); //trả về trạng thái của trang web
console.log(document.scripts); //trả về các script trong trang web
console.log(document.title); //trả về tiêu đề của trang web
```

```

Uploaded using RayThis Extension
Uploaded using RayThis Extension

<body>
  <main>
    <div class="e-class"></div>
    <div id="e-id"></div>
    <p>Hello world</p>
    <input type="text" name="hi">
  </main>
  <script>
    var x = document.getElementById("e-id");
    var t = document.getElementsByName("hi")[0];
    var y = document.getElementsByClassName("e-class");
    var z = document.getElementsByTagName("p")[0];
    var u = document.querySelectorAll("div, p");
  </script>
</body>

<h1>Chọn tất cả các checkbox</h1>
<body>
  <main>
    <form action="return.html" method="post" enctype="multipart/form-data">
      Javascript <input type="checkbox" name="course" id="" value="js">
      PHP <input type="checkbox" name="course" id="" value="php">
      <button type="button" onclick="checkall()>Checkall</button>
    </form>
  </main>
  <script>
    function checkall(){
      var x = document.getElementsByName("course");
      for(var i=0; i < x.length; i++) {
        if(x[i].type == "checkbox" ){
          x[i].checked = true;
        }
      }
    }
  </script>
</body>

```

```

Uploaded using RayThis Extension

<body id="body">
  <div class="hi">
    <main>
      <p>This is an element node</p>
      This is an text node
      <h3>This is an heading element</h3>
    </main>
  </div>
  <script>
    var root = document.getElementsByTagName("main")[0];
    var childNodes = root.childNodes; // trả về danh sách các node con
    //tức là bao gồm cả This is an text node
    var element = root.children; //trả về danh sách các element node con
    var firstChild = root.firstChild; //Node con đầu tiên, có thể là cả text
    var firstElementChild = root.firstElementChild; //Element con đầu tiên (p)
    var lastChild = root.lastChild; //Node con cuối cùng, có thể là cả text
    var lastElementChild = root.lastElementChild; //Element con cuối cùng (h3)

    var parentNode = root.parentNode; //Node cha, có thể cả text (div)
    var parentElement = root.parentElement; //element node cha (div)
    var closest = root.closest("#body"); //Tìm node tổ tiên ancestor của node hiện tại
    (body)
    var nextSibling = root.nextSibling; //Node anh em tiếp theo ruột phải, có thể là cả
    text
    var nextElementSibling = root.nextElementSibling; //Element node anh em tiếp theo ruột
    phải
    var previousSibling = root.previousSibling; //Node anh em ruột trái, có thể là cả text
    var previousElementSibling = root.previousElementSibling; //Element node anh em ruột
    trái

    for(let i = 0; i < element.length; i++) {
      console.log(element[i].innerText);
      //Nội dung của node, không bao gồm HTML, khoảng trắng và nội dung có thuộc tính
      css ẩn
      console.log(element[i].textContent);
      //Tương tự như innerText nhưng bao gồm cả khoảng trắng, nội dung có thuộc tính CSS
      ẩn
      console.log(element[i].innerHTML);
      //Trả về toàn bộ text, mã html và khoảng trắng
    }
  </script>
</body>

```

```

Upgraded using RayThis Extension

<body>
    <h1 id="myTitle" class="title">Hello, world!</h1>
    <button onclick="changeProperties()">Thay đổi thuộc tính</button>
    <script>
        function changeProperties() {
            let node = document.getElementById("myTitle");

            // 1. document.write() - Ghi giá trị trực tiếp vào trang HTML
            // (lưu ý: sẽ thay thế toàn bộ nội dung trang nếu dùng sau khi DOM đã tải)
            document.write("<p>Đây là văn bản mới!</p>");

            // 2. node.setAttribute - Truy xuất hoặc gán giá trị trực tiếp cho thuộc tính
            console.log(node.className); // Lấy giá trị của class (title)
            node.className = "new-class"; // Gán giá trị mới cho class

            // 3. node.getAttribute(attrib) - Lấy giá trị của một thuộc tính cụ thể
            console.log(node.getAttribute("id")); // Kết quả: "myTitle"

            // 4. node.setAttribute(attrib, val) - Gán giá trị cho một thuộc tính cụ thể
            node.setAttribute("data-info", "Thông tin mới"); // Thêm thuộc tính `data-info`

            // 5. node.style.property - Thay đổi giá trị định dạng của node
            node.style.color = "red"; // Thay đổi màu chữ thành đỏ
            node.style.fontSize = "24px"; // Tăng kích thước chữ
        }
    </script>
</body>

```

```

Upgraded using RayThis Extension

<body>
    <ul id="list"></ul>
    <script>
        var element = document.getElementById("list");
        var browsers = ["chrome", "edge", "firefox"];

        var textul = document.createTextNode("xin chao the gioi"); // tao doan text
        element.appendChild(textul); // = element.innerText = "Xin chao the gioi";
        var attribute = document.createAttribute("name");
        attribute.value = "Xin chao the gioi";
        element.setAttributeNode(attribute); // = element.name = "newName";

        let clone = element.cloneNode(true); // Sao chép toàn bộ nội dung
        let clone1 = element.cloneNode(false);
        // Chỉ sao chép phần tử, không sao chép nội dung con
        var frag = document.createDocumentFragment(); // vung nhor tam
        for (var i = 0; i < browsers.length; i++) {
            var li = document.createElement("li");
            li.textContent = browsers[i];
            frag.appendChild(li);
        }
        element.appendChild(frag);
        // output: chrome
        //          edge
        //          firefox
    </script>
</body>

```

```

Upgraded using RayThis Extension

<body>
    <label for="txt">Enter a skill: </label><input type="text" id="txt">
    <button onclick = "additem()">Add</button>
    <p>Your technical skill: </p> <ul id="list"></ul>
    <script>
        function additem(){
            var txt = document.getElementById("txt");
            var list = document.getElementById("list");
            var item = document.createElement("li");
            item.textContent = txt.value;
            txt.value = ""; // = null;
            list.appendChild(item);
            txt.focus(); // lấy focus
        }
    </script>
</body>

```

```

Upgraded using RayThis Extension

<body>
    <label for="username">Tên đăng nhập:</label>
    <input type="text" id="username" placeholder="Nhập tên..." />
    <p id="message" style="color: red;"></p>

    <script>
        document.getElementById("username").addEventListener("blur",
            function() {
                let input = this.value.trim();
                // Xóa những dấu cách thừa ở đầu và ở cuối
                let message = document.getElementById("message");

                if (input === "") {
                    message.textContent = "Tên đăng nhập không được để trống!";
                } else {
                    message.textContent = "";
                }
            });

        document.getElementById("username").addEventListener("click",
            function(event){
                });
    </script>
</body>

```

```

Upgraded using RayThis Extension

<body>
    <button onclick="sayHi()">Click me 1</button>
    <button id="test1">Click me 2</button>
    <button id="test2">Click me 3</button>

    <script>
        var btn1 = document.getElementById('test1');
        var btn2 = document.getElementById('test2');

        function sayHi() { alert('Hi'); }
        // Cách 1: Gán sự kiện trực tiếp trong HTML (inline event handler)
        // Đã thực hiện trong thẻ button đầu tiên: onclick="sayHi()"
        // Cách 2: Sử dụng addEventListener (recommended)
        btn1.addEventListener('click', sayHi);
        // Cách 3: Gán sự kiện qua thuộc tính DOM (onclick)
        btn2.onclick = sayHi;
    </script>
</body>

```

Uploaded using RayThis Extension

```

<body>
    <!-- Cấu trúc DOM với phần tử cha và con -->
    <div id="parent">
        Parent Div (Click vào vùng xanh)
        <div id="child">Child Div (Click vào vùng vàng)</div>
    </div>

    <script>
        const parent = document.getElementById('parent');
        const child = document.getElementById('child');

        // Xử lý sự kiện cho phần tử cha (sử dụng bubbling phase)
        parent.addEventListener('click', function(event) {
            console.log('--- PARENT EVENT HANDLER ---');
            console.log('event.target : ', event.target.id); // Phần tử gốc gây ra sự kiện
            console.log('event.currentTarget : ', event.currentTarget.id); // Phần tử đang xử lý
            // sự kiện (luôn là parent)
        });

        </script>
    </body>

```

Uploaded using RayThis Extension

```

<body>
    <form name="myForm" onsubmit="return frmValiddate5()" method="POST">
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br><br>
        <label for="hoten">Full Name:</label>
        <input type="text" id="hoten" name="hoten" required><br><br>
        <label for="noidung">Content:</label>
        <textarea id="noidung" name="noidung" required></textarea><br><br>
        <button type="submit">Submit</button>
    </form>
    <script>
        function frmValiddate5() {
            var emailReg = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
            var email = document.getElementById("email").value;
            if (emailReg.test(email) == false) {
                alert("Please enter a valid email address");
                return false;
            }
            var password = document.getElementById("password").value;
            if (password.length < 8) {
                alert("Password should be at least 8 characters long");
                return false;
            }

            var hoten = document.getElementById("hoten").value;
            if (hoten.length < 4) {
                alert("Name should be at least 4 characters long");
                return false;
            }
            var noidung = document.getElementById("noidung").value;
            if (noidung.length < 10) {
                alert("Content should be at least 10 characters long");
                return false;
            }
        }
    </script>
</body>

```

Uploaded using RayThis Extension

```

<body>
    <h1>Ví dụ sử dụng localStorage</h1>
    <div>
        <input type="text" id="keyInput" placeholder="Nhập key">
        <input type="text" id="valueInput" placeholder="Nhập value">
    </div>
    <div>
        <button onclick="setItem()">Lưu vào localStorage (setItem)</button>
        <button onclick="getItem()">Lấy giá trị (getItem)</button>
        <button onclick="removeItem()">Xóa item (removeItem)</button>
        <button onclick="clearStorage()">Xóa tất cả (clear)</button>
    </div>
    <div class="output" id="output"></div>

    <script>
        const output = document.getElementById('output');
        const keyInput = document.getElementById('keyInput');
        const valueInput = document.getElementById('valueInput');
        function setItem() {
            const key = keyInput.value.trim();
            const value = valueInput.value.trim();
            localStorage.setItem(key, value);
            output.innerHTML = `Đã lưu: ${key} = ${value}`;
        }
        function getItem() {
            const key = keyInput.value.trim();
            const value = localStorage.getItem(key);
            output.innerHTML = value
                ? `Giá trị của ${key}: ${value}`
                : 'Không tìm thấy key ${key}';
        }
        function removeItem() {
            const key = keyInput.value.trim();
            localStorage.removeItem(key);
            output.innerHTML = `Đã xóa key ${key}`;
        }
        function clearStorage() {
            localStorage.clear();
            output.innerHTML = 'Đã xóa toàn bộ dữ liệu trong localStorage';
        }
    </script>
</body>

```

Uploaded using RayThis Extension

```

<body>
    <form id="regform" action="#" method="post" enctype="multipart/form-data">
        <label for="username">Username:</label> <input type="text" id="username" name="username">
        <label for="password">Password:</label> <input type="password" id="password" name="password">
        <button type="submit">Submit</button>
    </form>

    <script>
        var form = document.getElementById("regform");
        form.addEventListener("focusin", function(event) {
            if(event.target.id == "username" || event.target.id == "password") {
                event.target.style.backgroundColor = 'yellow';
            }
        });
        form.addEventListener("focusout", function(event) {
            if(event.target.id == "username" || event.target.id == "password") {
                event.target.style.backgroundColor = 'initial';
            }
        });
    </script>
</body>

```

Uploaded using RayThis Extension

```

<body>
    <form name="myForm" id="form" method="POST">
        ...
    </form>
    <button type="button" onclick="frmValiddate5()">Submit</button>
    <script>
        function frmValiddate5() {
            var form = document.forms['myForm'];
            form.submit();
        }
    </script>
</body>

```

Uploaded using RayThis Extension

```

<body>
    <main>
        <h1>Danh sách sản phẩm</h1>
        <div id="product-list"></div>
    </main>
    <script>
        const productJSON = `{
            "products": [
                { "id": 1,      "name": "John" },
                { "id": 2,      "name": "Doe" }
            ]
        }`;
        const productList = document.getElementById('product-list');
        const productData = JSON.parse(productJSON);
        productData.products.forEach(function(product){
            const productItem = document.createElement('div');
            productItem.textContent = `ID: ${product.id}, Name: ${product.name}`;
            productList.appendChild(productItem);
        });
    </script>
</body>

```

Uploaded using RayThis Extension

```

<body>
    <div id="demo">Nội dung tải về sẽ hiển thị ở đây...</div>
    <button onclick="loadDoc()">Tải Dữ Liệu</button>

    <script>
        function loadDoc() {
            // Tạo đối tượng XMLHttpRequest
            const xhttp = new XMLHttpRequest();

            // Xử lý khi dữ liệu được tải về
            xhttp.onload = function() {
                if (this.status === 200) {
                    document.getElementById("demo").innerHTML = this.responseText;
                } else {
                    document.getElementById("demo").innerHTML = "Lỗi khi tải dữ liệu: " + this.status;
                }
            };

            // Xử lý khi có lỗi
            xhttp.onerror = function() {
                document.getElementById("demo").innerHTML = "Yêu cầu thất bại";
            };
            // Mở kết nối và gửi yêu cầu
            xhttp.open("GET", "dulieu.txt", true);
            xhttp.send();
        }
    </script>
</body>

```

```

<script>
$(document).ready(function() {
    // 1. Selector và DOM Traversal
    $("#btnTraversal").click(function() {
        let output = "";
        output += "ID của div đầu tiên: " + $("#div1").attr("id") + "<br>";
        output += "Số lượng div: " + $(".box").length + "<br>";
        output += "Nội dung của div2: " + $("#div2").text() + "<br>";
        output += "HTML của div3: " + $("#div3").html() + "<br>";
        output += "Parent của list: " + $("#list").parent().attr("class") + "<br>";
        output += "Children của list: " + $("#list").children().length + "<br>";
        output += "First li: " + $("#list li:first").text() + "<br>";
        output += "Last li: " + $("#list li:last").text() + "<br>";

        $("#result").html(output);
    });

    // 2. Thao tác DOM
    $("#btnHtml").click(function() {
        $("#content").html("<strong>Nội dung mới với HTML</strong>"); });
    $("#btnText").click(function() {
        $("#content").text("<strong>Nội dung mới chỉ text</strong>"); });
    $("#btnAppend").click(function() {
        $("#content").append("<span> - appended text</span>"); });
    $("#btnPrepend").click(function() {
        $("#content").prepend("<span>prepended text - </span>"); });
    $("#btnAfter").click(function() {
        $("#content").after("<div>Element added after</div>"); });
    $("#btnBefore").click(function() {
        $("#content").before("<div>Element added before</div>"); });
    $("#btnRemove").click(function() { $(".container:last").remove(); });
    $("#btnEmpty").click(function() { $("#content").empty(); });

    // 3. CSS và Attributes
    $("#btnAddClass").click(function() {
        $("#styleBox").addClass("highlight"); });
    $("#btnRemoveClass").click(function() { $("#styleBox").removeClass("highlight"); });
    $("#btnToggleClass").click(function() { $("#styleBox").toggleClass("highlight"); });
    $("#btnCss").click(function() {
        $("#styleBox").css({"border": "5px solid black", "border-radius": "10px"}); });
    $("#btnAttr").click(function() {
        $("#styleBox").attr("title", "This is a box");
        alert("Title attribute set to: " + $("#styleBox").attr("title")); });
    $("#btnData").click(function() {
        $("#styleBox").data("info", {id: 123, name: "Demo Box"});
        alert("Data stored: " + JSON.stringify($("#styleBox").data("info"))); });

    // 4. Sự kiện
    $("#btnClick").click( function() { $("#eventLog").append("Button clicked<br>");});
    $("#btnHover").hover( function() { $("#eventLog").append("Mouse entered<br>"); },
                        function() { $("#eventLog").append("Mouse left<br>"); });
    $("#btnOn").click(function() {
        $("body").on("click", "button", function() {
            $("#eventLog").append("Button clicked via on(): " + $(this).text() + "<br>"); });
    });
    $("#btnOff").click(function() { $("body").off("click", "button"); });

    // 5. Hiệu ứng
    $("#btnShow").click(function() { $("#effectBox").show(); });
    $("#btnHide").click(function() { $("#effectBox").hide(); });
    $("#btnToggle").click(function() { $("#effectBox").toggle(); });
    $("#btnFadeIn").click(function() { $("#effectBox").fadeIn(); });
    $("#btnFadeOut").click(function() { $("#effectBox").fadeOut(); });
    $("#btnSlideDown").click(function() { $("#effectBox").slideDown(); });
    $("#btnSlideUp").click(function() { $("#effectBox").slideUp(); });
    $("#btnAnimate").click(function() {
        $("#effectBox").animate({
            width: "200px",
            height: "50px",
            opacity: 0.5
        }, 1000); });

```

```
function checkDate() {
    var dateInput = document.getElementById("date").value;
    var date = new Date(dateInput);
    var today = new Date();
    if (date > today) {
        alert("Ngày sinh không hợp lệ!");
        document.getElementById("date").value = ""; // Reset the input field
    }
}
```

```
var gender = document.querySelector('input[name="gender"]:checked');
if (!gender) {
    alert("Vui lòng chọn giới tính!");
    return false;
}
```

```
var email = document.getElementById("email").value;
var emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
if (!emailPattern.test(email)) {
    alert("Email không hợp lệ!");
    return false;
}

var password = document.getElementById("password").value;
var passReg = /^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[#!@#$%^&*-]).{8,}$/g;
if(passReg.test(password) == false){
    alert("Mật khẩu không hợp lệ!");
    return false;
}

var url = document.getElementById("url").value;
var urlReg = /^((http(s?)):\/\/)?([wW]{3}\.)([a-zA-Z0-9\-\.\.]+\.[a-zA-Z]{2,})(\.[a-zA-Z]{2,})?$/g;
if(urlReg.test(url) == false){
    alert("URL không hợp lệ!");
    return false;
}
```