



# JavaScript

Ngôn ngữ lập trình "*kịch bản*" cho web page

Môn học: Nhập môn lập trình [*Buổi 7-8*]

GV: Nguyễn Mai Huy



*Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)*

# Giới thiệu

JavaScript là ngôn ngữ lập trình “*kịch bản*” (*Script*) có cú pháp để biểu diễn mã lệnh “*tựa ngôn ngữ C*”. Trên thực tế, ban đầu JavaScript được phát triển từ ngôn ngữ C (*derived from C*) để thi hành các kịch bản định sẵn cho trang web, trong khi đó ngữ nghĩa và mục tiêu thiết kế của nó lại mang hơi hướng của lập trình hướng đối tượng (*gần giống Java*).

Như vậy, khi lập trình với JavaScript, ngoài việc thao tác với các biến và các cấu trúc lệnh, thì “*đối tượng*” chính là thành phần mà chúng ta thường hay làm việc nhất.

# Giới thiệu

- **“Đối tượng” ở đây là gì ?**. Vâng, rất đơn giản, đối tượng chính là những thành phần thuộc trang web mà bạn đã tạo ra dựa trên mã HTML. Ví dụ như các thành phần: **p, ul, a, li, img, table, div, span, input, h1, h2, ...** những thành phần mà chúng ta vẫn sử dụng để mô tả dữ liệu của trang web.
- **Tại sao phải thao tác trên các đối tượng *HTML* ?**. Điều này rất dễ hiểu, do mã HTML mà bạn đã tạo ra vào thời điểm thiết kế được xem là “*đối tượng tĩnh*” (*static objects*) và những đối tượng này luôn cố định, gần như không thay đổi trong quá trình hoạt động trên trình duyệt. Khi chúng ta muốn trang web “*thông minh hơn*”, có thể “*phản ứng*” theo các tương tác của người dùng thì JavaScript chính là lựa chọn cho phép bạn thực hiện điều đó.

# Làm được gì với JavaScript ?

- Thêm, tạo mới các thành phần **HTML** cho trang web
- Chỉnh sửa, thay đổi *thuộc tính* của HTML (*Đổi màu, kích thước, ...*)
- Xóa bỏ các thành phần HTML đã có
- Kiểm tra tính hợp lệ đối với dữ liệu mà người dùng nhập trước khi gửi về chương trình ở máy chủ web
- ...

# Một số lưu ý

- **Sử dụng trực tiếp mã lệnh trên trang web**: Để viết mã JavaScript trên trang web, bạn chỉ việc giới hạn phạm vi bằng cặp tag <script> theo cú pháp như sau

**<script type= 'text/javascript'>**

...

*Mã lệnh viết bằng Java script;*

...

**</script>**

- **Tham chiếu dưới dạng thư viện**: Bạn cũng có thể khai báo các thư viện chứa mã JavaScript để sử dụng cho trang web của mình bằng cách sử dụng tag link theo cú pháp như sau

**<script type="text/javascript" src="Url\_trở\_đến\_file\_js"></script>**



# Một số khái niệm

- Lệnh & Tập lệnh:
- Hàm (*Function*):

```
function tenHam(thamSo, thamSo, ...)  
{  
    ...  
    // -- Các lệnh thuộc hàm  
    ...  
}
```

- Đối tượng (*Object*):
  - **Object browser**
  - **HTML DOM Object**

# Khai báo biến

- Khai báo biến

```
var ten;  
var x=2;  
var y=15;  
var loaiTien = 'USD', dvt = 'Thùng';
```

hoặc cũng có thể khai báo đồng thời khởi gán giá trị cho biến

```
var tên_biến = Giá_trị;
```



# Phép toán



Số học

Operator	Description	Comparing	Returns
==	equal to	x == 8	false



I

```
Biến = (điều_kiện) ? value1 : value2;
```



I VD



I

```
ketQua = (dtb >= 5) ? 'Đạt yêu cầu' : 'Rớt, học lại';
```



Toán

		x != 5	false
>	greater than	x > 8	false
<	less than	x < 8	true
>=	greater than or equal to	x >= 8	false
<=	less than or equal to	x <= 8	true

ry - nmai huy@bodu a.com

# Cấu trúc điều kiện

```
switch (biến) {  
    case gt1:  
        //--- Các lệnh sẽ thi hành khi biến mang gt1 tương ứng  
        break;  
    case gt2:  
        //--- Các lệnh sẽ thi hành khi biến mang gt2 tương ứng  
        break;  
    . . .  
    default:  
        //--- Các lệnh sẽ thi hành khi biến không phải là một  
        //--- trong các giá trị đã liệt kê ở phía trước  
        break;  
}
```

# Cấu trúc lặp

Hoặc do ... while

```
do
{
    . . .
    //--- Các lệnh sẽ thi hành trong vòng lặp
    . . .
}while (dk_lặp_tiếp)
```

▼ Cấu trúc while

```
while (dk_lặp)
{
    . . .
    //--- Các lệnh sẽ thi hành trong vòng lặp
    . . .
}
```

# Một số phương thức cơ bản

- *element*.**getElementsById()**
- *element*.**getElementsByClassName()**
- *element*.**getElementsByTagName()**
- *element*.**innerHTML()**
- *element*.**createElement()**

# Một số hàm thường dùng

- **confirm**('Chuỗi\_thông\_báo')
- **alert**('Chuỗi\_thông\_báo')
- **prompt**('Chuỗi\_thông\_báo', 'giá\_trị')
- **scrollTo**()

# Sự kiện tiêu biểu

- **onchange()**
- **onfocus()**
- **onblur()**
- **onclick()**
- **onmousedown()**
- **onmouseup()**
- **oncontextmenu ()**

truy xuất sự kiện của đối tượng trong JavaScript như sau

```
object.onchange=function(){//--Mã lệnh JavaScript; };
```

hoặc viết trong HTML trực tiếp trên tag

```
<tag onchange = "Mã lệnh JavaScript;"
```

# Exam #1

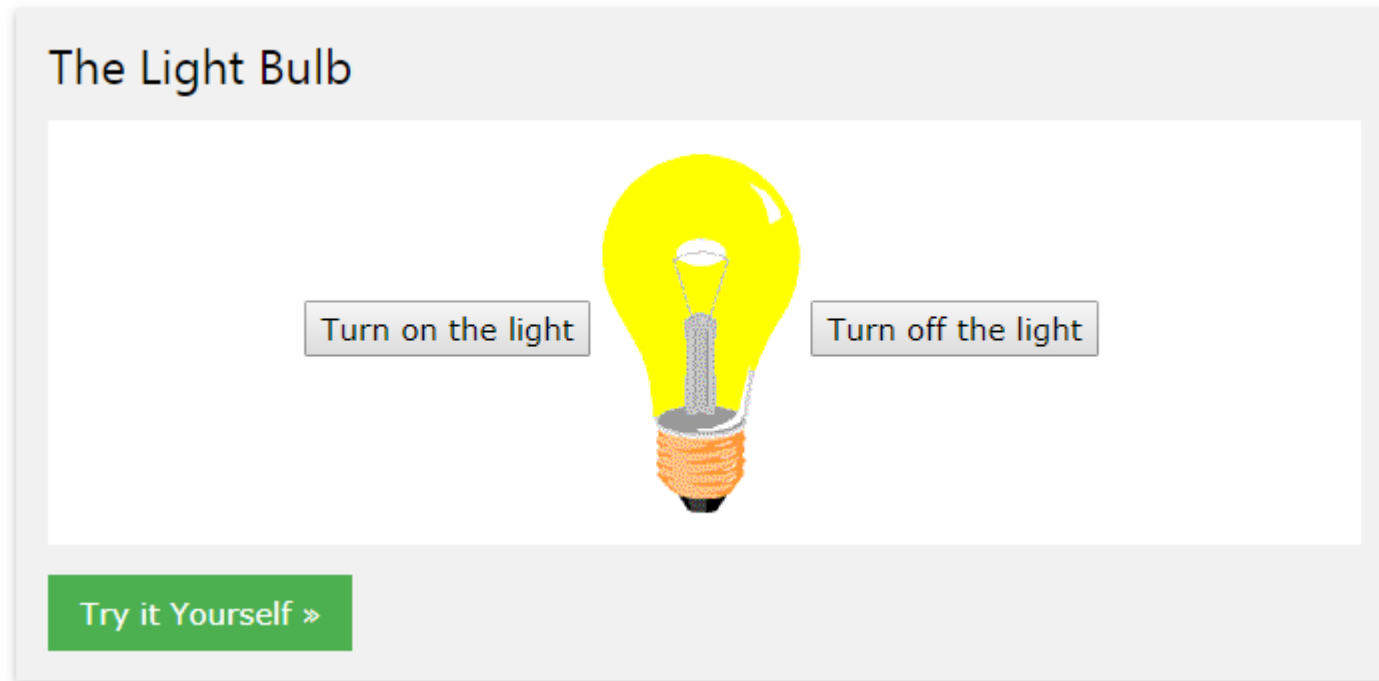
- Trong ví dụ này, chúng ta sẽ viết một kịch bản bằng JavaScript để cấm người dùng đánh dấu, sau chép dữ liệu là nội dung của trang web dựa trên sự kiện onmousedown của thành phần body trong cấu trúc DOM Tree

```
6 <script text='>  
7     function camSaoChep(){  
8         alert('Bạn không được phép sao chép dữ liệu của chúng tôi !.');  
9         return false;  
10    }  
11 </script>  
12 </head>  
13 <body onmousedown='camSaoChep();'>  
14     <table>  
15         <tr>  
16             <td colspan='2'>  
17                 <h1>Phần tiêu đề đầu trang của website</h1>  
18             </td>  
19         </tr>
```

Mã JavaScript được viết trong cặp tag <script> thuộc phần head của trang web



# Exam #2



Để thực hiện ví dụ này, chúng ta sẽ viết một kịch bản bằng **JavaScript** để thay đổi thuộc tính **src** của **img** tag bằng một hình khác, khi người dùng click vào nút lệnh tương ứng (Như vậy, cần có 2 hình tương ứng với trạng thái bật tắt của bóng đèn)

Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)

# Exam #3

## What Can JavaScript Do?

JavaScript can show hidden HTML elements.



Để thực hiện ví dụ này, chúng ta sẽ viết một kịch bản bằng **JavaScript** để thay đổi giá trị cho thành phần **display** của thuộc tính **style** bằng “**block**” hay “**none**”, đồng thời đổi giá trị của thuộc tính **Text** của button thành “**Show**” hoặc “**Hide**” khi người dùng click vào nút lệnh tương ứng





*Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)*

# What is **jQuery** ?

- **jQuery** is a lightweight, "*write less, do more*", JavaScript library.
- The purpose of **jQuery** is to make it much easier to use **JavaScript** on your website.
- **jQuery** takes a lot of common tasks that require many lines of **JavaScript** code to accomplish, and wraps them into methods that you can call with a single line of code.
- **jQuery** also simplifies a lot of the complicated things from **JavaScript**, like **AJAX** calls and **DOM** manipulation.

# What **jQuery** can do ?

The jQuery library contains the following features:

- ✓ HTML/DOM manipulation
- ✓ CSS manipulation
- ✓ HTML event methods
- ✓ Effects and animations
- ✓ AJAX
- ✓ Utilities

# How to use **Bootstrap** on your web

## ▪ Adding jQuery to Your Web Pages

- ✓ Downloading jQuery from [jQuery.com](https://jquery.com)
- ✓ The **jQuery** library is a single JavaScript file, and you reference it with the HTML **<script>** tag (*notice that the **<script>** tag should be inside the **<head>** section*):

```
<head>  
    <script src="yourDirectory/jquery-3.3.1.min.js"></script>  
</head>
```



# How to use **Bootstrap** on your web

## ▪ jQuery CDN

- ✓ If you don't want to download and host **jQuery** yourself, you can include it from a CDN (*Content Delivery Network*).
- ✓ Both Google and Microsoft host jQuery.
- ✓ To use **jQuery** from Google or Microsoft, use one of the following:

Google CDN:

```
<head>  
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>  
</head>
```

Nguyễn Mai Huy - [nmaihuy@boduac.com](mailto:nmaihuy@boduac.com)

# jQuery Syntax

- The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).
- Basic syntax is: ***\$(selector).action()***
  - ✓ A \$ sign to define/access jQuery
  - ✓ A (*selector*) to "query (or find)" HTML elements
  - ✓ A jQuery *action()* to be performed on the element(s)

# jQuery Syntax :: Examples

- `$(this).hide()` - hides the current element.
- `$("p").hide()` - hides all `<p>` elements.
- `$(".test").hide()` - hides all elements with `class="test"`.
- `$("#test").hide()` - hides the element with `id="test"`.

# jQuery **Selectors**

- jQuery selectors allow you to select and manipulate HTML element(s).
- jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing [CSS Selectors](#), and in addition, it has some own custom selectors.
- All selectors in jQuery start with the dollar sign and parentheses: \$().

# The **element** Selector

- The jQuery element selector selects elements based on the element name. You can select all <p> elements on a page like this: **`$("p")`**

## Example

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("p").hide();  
    });  
});
```

# The **#id** Selector

- The jQuery **#id** selector uses the id attribute of an HTML tag to find the specific element. An id should be unique within a page, so you should use the **#id** selector when you want to find a single, unique element.
- To find an element with a specific id, write a hash character, followed by the id of the HTML element:

## Example

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#test").hide();  
    });  
});
```

# The **.class** Selector

- The jQuery class selector finds elements with a specific class.  
To find elements with a specific class, write a period character, followed by the name of the class: **`$(".test")`**

## Example

```
$(document).ready(function(){  
    $("button").click(function(){  
        $(".test").hide();  
    });  
});
```



# More Examples of jQuery Selectors

Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <p> elements with class="intro"
<code>\$("p:first")</code>	Selects the first <p> element
<code>\$("ul li:first")</code>	Selects the first <li> element of the first <ul>
<code>\$("ul li:first-child")</code>	Selects the first <li> element of every <ul>

[https://www.w3schools.com/jquery/jquery\\_selectors.asp](https://www.w3schools.com/jquery/jquery_selectors.asp)

Nguyễn Mai Huy - [nmaihuy@bodu.edu.vn](mailto:nmaihuy@bodu.edu.vn)

## What are Events?

All the different visitors' actions that a web page can respond to are called events. An event represents the precise moment when something happens.

Examples:

- ❖ moving a mouse over an element
- ❖ selecting a radio button
- ❖ clicking on an element

## jQuery Syntax For Event Methods

- In jQuery, most DOM events have an equivalent jQuery method.
- To assign a click event to all paragraphs on a page, you can do this: **`$("p").click();`**

The next step is to define what should happen when the event fires. You must pass a function to the event:

```
$("p").click(function(){  
    // action goes here!!  
});
```

# Common DOM **events**

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

# Example :: The click() method

- The click() method attaches an event handler function to an HTML element. The function is executed when the user clicks on the HTML element.
- The following example says: When a click event fires on a <p> element; hide the current <p> element:

## Example

```
$("#p").click(function(){  
    $(this).hide();  
});
```

# Example :: The `dblclick()` method

- The `dblclick()` method attaches an event handler function to an HTML element. The function is executed when the user double-clicks on the HTML element:

## Example

```
$("#p").dblclick(function(){  
    $(this).hide();  
});
```

# Example :: The **mouseenter()** method

- The `mouseenter()` method attaches an event handler function to an HTML element. The function is executed when the mouse pointer enters the HTML element:

## Example

```
$("#p1").mouseenter(function(){  
    alert("You entered p1!");  
});
```



# Example :: The **mouseleave()** method

- The `mouseleave()` method attaches an event handler function to an HTML element. The function is executed when the mouse pointer leaves the HTML element:

## Example

```
$("#p1").mouseleave(function(){  
    alert("Bye! You now leave p1!");  
});
```

# Example :: The **mousedown()** method

- The mousedown() method attaches an event handler function to an HTML element. The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

## Example

```
$("#p1").mousedown(function(){  
    alert("Mouse down over p1!");  
});
```

# Example :: The **mouseup()** method

- The `mouseup()` method attaches an event handler function to an HTML element. The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

## Example

```
$("#p1").mouseup(function(){  
    alert("Mouse up over p1!");  
});
```

# Example :: The **hover()** method

- The `hover()` method takes two functions and is a combination of the `mouseenter()` and `mouseleave()` methods. The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

## Example

```
$("#p1").hover(function(){  
    alert("You entered p1!");  
},  
function(){  
    alert("Bye! You now leave p1!");  
});
```

Nguyễn Mai Huy - [nmai.huy@bodu.com](mailto:nmai.huy@bodu.com)

# Example :: The **focus()** method

- The `focus()` method attaches an event handler function to an HTML form field. The function is executed when the form field gets focus:

## Example

```
$("#input").focus(function(){  
    $(this).css("background-color", "#cccccc");  
});
```

# Example :: The **blur()** method

- The blur() method attaches an event handler function to an HTML form field. The function is executed when the form field loses focus:

## Example

```
$("#input").blur(function(){  
    $(this).css("background-color", "#ffffff");  
});
```

# Example :: The **on()** method

- The on() method attaches one or more event handlers for the selected elements. Attach a click event to a <p> element:

## Example

```
$("#p").on({  
    mouseenter: function(){  
        $(this).css("background-color", "lightgray");  
    },  
    mouseleave: function(){  
        $(this).css("background-color", "lightblue");  
    },  
    click: function(){  
        $(this).css("background-color", "yellow");  
    }  
});
```

# The Document Ready Event

- You might have noticed that all jQuery methods in our examples, are inside a document ready event. This is to prevent any jQuery code from running before the document is finished loading (is ready).
- It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

```
$(document).ready(function(){  
  
    // jQuery methods go here...  
  
});
```

Nguyễn Mai Huy - [nmaihuy@boduac.com](mailto:nmaihuy@boduac.com)



# jQuery **hide()**, **show()** & **toggle()**

- With jQuery, you can hide and show HTML elements with the **hide()** and **show()** methods

```
$(selector).hide(speed,callback);  
  
$(selector).show(speed,callback);  
  
$(selector).toggle(speed,callback);
```

- And, you can also **toggle** between the **hide()** and **show()** methods with the **toggle()** method.

# jQuery Effects - **Fading**

- With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

❖ **fadeIn()**

❖ **fadeOut()**

❖ **fadeToggle()**

❖ **fadeTo()**

```
$(selector).fadeIn(speed, callback);
```

```
$(selector).fadeOut(speed, callback);
```

```
$(selector).fadeToggle(speed, callback);
```

Nguyễn Mai Huy - [nmaihuy@bodu.com](mailto:nmaihuy@bodu.com)

# jQuery Effects - **Sliding**

You also can create a sliding effect on elements with jQuery.  
some slide methods are:

❖ **slideDown()**

❖ **slideUp()**

❖ **slideToggle()**

```
$(selector).slideDown(speed, callback);
```

```
$(selector).slideUp(speed, callback);
```

```
$(selector).slideToggle(speed, callback);
```

# jQuery Effects - Animation

The jQuery `animate()` method lets you create custom animations.

```
$(selector).animate({params}, speed, callback);
```

- ❖ The required `params` parameter defines the CSS properties to be animated.
- ❖ The optional `speed` parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- ❖ The optional `callback` parameter is a function to be executed after the animation completes.

# jQuery animate() - Manipulate Multiple Properties

## Example

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px',  
        opacity: '0.5',  
        height: '150px',  
        width: '150px'  
    });  
});
```

# jQuery animate() - Using Relative Values

## Example

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px',  
        height: '+=150px',  
        width: '+=150px'  
    });  
});
```

# jQuery animate() - Uses Queue Functionality

## Example 1

```
$("#button").click(function(){  
    var div = $("#div");  
    div.animate({height: '300px', opacity: '0.4'}, "slow");  
    div.animate({width: '300px', opacity: '0.8'}, "slow");  
    div.animate({height: '100px', opacity: '0.4'}, "slow");  
    div.animate({width: '100px', opacity: '0.8'}, "slow");  
});
```



# Nhớ gì ?!!!



*Nguyễn Mai Huy - [nmaihuy@boduca.com](mailto:nmaihuy@boduca.com)*



# Tài liệu tham khảo


- Jose M. Garrido, “**Object-Oriented Programming: From Problem Solving to Java**”
- Paul Deitel, Harvey Deitel, “**Java : How to program**”, 9<sup>th</sup> edition, 2012
- Oracle, “**The Java™ Tutorials**”,  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>,  
6:20PM, 18/01/2018
- Java tutorial, <https://www.javatpoint.com/java-tutorial> , 6:20PM,  
18/01/2018

# Facebook Group


facebook.com/groups/1508092902561925/

Facebook


Quản lý nhóm


 **Web programming [ASP.NET]**   
Nhóm Riêng tư

← Nếu muốn, bạn có thể chuyển sang Facebook.com thông thường trong 48 giờ để quản lý nhóm.  
[Chuyển ngay](#)


 Trang chủ

Công cụ quản trị

-  Yêu cầu tham gia
-  Tự động phê duyệt thành viên



**Web programming [ASP.NET]**  
Nhóm Riêng tư · 378 thành viên


 [+ Mời](#)

[Chỉnh sửa](#)

Nguyễn Mai Huy - [nmai-huy@boduca.com](mailto:nmai-huy@boduca.com)

# Youtube chanel :: Lecture

bodua group


 **BODUA Group**  
197 người đăng ký

TÙY CHÍNH KÊNH QUẢN LÝ VIDEO

TRANG CHỦ VIDEO **DANH SÁCH PHÁT** KÊNH CUỘC THẢO LUẬN GIỚI THIỆU


Danh sách phát đã tạo

SẮP XẾP THEO




**Tổng quan** 3

Web layout  
Cập nhật 5 ngày trước  
XEM TOÀN BỘ DANH SÁCH




**Tổng quan** 10  
Lập trình giao diện

UI Programming in Windows  
XEM TOÀN BỘ DANH SÁCH




**Overview of DBMS** 12

SQL Server  
XEM TOÀN BỘ DANH SÁCH



Mô phỏng: Quản lý đơn hàng  
**Purchase Order** 17

Video đã thích  
Riêng tư  
XEM TOÀN BỘ DANH SÁCH




**Giới thiệu ngôn ngữ Java** 8

Lập trình hướng đối tượng [JAVA]  
XEM TOÀN BỘ DANH SÁCH



# Youtube chanel :: Practice



**Huy Nguyen Mai**  
246 người đăng ký

ĐÃ ĐĂNG KÝ

TRANG CHỦ VIDEO **DANH SÁCH PHÁT** KÊNH CUỘC THẢO LUẬN GIỚI THIỆU

Video tải lên ▶ PHÁT TẤT CẢ

Web layout: HTML Document #2

**Your site & Data**  
Image, Audio, Video file and Folder structure

CÔNG CHIẾU

Web layout :: Practice 2 - Your site & Data

Ngày công chiếu: 08:15, 25/10/2020

ĐẶT LỜI NHẮC

Web layout: HTML Document #1

**Structure & Text**  
Heading, Paragraph and Format tag

CÔNG CHIẾU

Web layout :: Practice 01 Structure Text

Ngày công chiếu: 19:15, 18/10/2020

ĐẶT LỜI NHẮC

Mô phỏng: Quản lý đơn hàng

**Orders tracking**

2:19:00

08 - Quản lý đơn hàng :: Order Tracking

365 lượt xem • 1 tuần trước

Mô phỏng: Quản lý đơn hàng

**Purchase Order**

1:56:42

07 - Quản lý đơn hàng :: Purchase order [Part 03 - ...]

103 lượt xem • 2 tuần trước

Mô phỏng: Quản lý đơn hàng

**Purchase Order**

1:54:16

07 - Quản lý đơn hàng :: Purchase order [Part 02 - UI...]

170 lượt xem • 2 tuần trước



**Working hard,  
Success in the future !**

*Nguyễn Mai Huy - [nmai.huy@bodu.edu.vn](mailto:nmai.huy@bodu.edu.vn)*