

Bài giảng

PHÁT TRIỂN ỨNG DỤNG WEB

Lê Đình Thanh

Khoa Công nghệ Thông tin

Trường Đại học Công nghệ, ĐHQGHN

E-mail: thanhld@vnu.edu.vn Mobile: 0987.257.504

Chương 5

Thư viện phát triển mặt trước

Nội dung

- **jQuery**
- Bootstrap
- React

jQuery

- jQuery là thư viện javascript giúp cho việc lập trình javascript trở nên đơn giản, hiệu quả hơn
- Các tính năng
 - Thao tác HTML/DOM
 - Xử lý sự kiện HTML
 - Thao tác CSS
 - Xử lý AJAX
 - Hiệu ứng và hoạt cảnh
 - Các tiện ích

Technologies

[Content Management](#)
[Server-side Languages](#)
[Client-side Languages](#)
[JavaScript Libraries](#)
[Markup Languages](#)
[Character Encodings](#)
[Image File Formats](#)
[Site Elements](#)
[SSL Certificate Authorities](#)
[Social Widgets](#)
[Web Servers](#)
[Operating Systems](#)
[Content Delivery](#)
[Traffic Analysis Tools](#)
[Advertising Networks](#)
[Tag Managers](#)
[Top Level Domains](#)
[Content Languages](#)

Trends

[Usage Trend](#)
[Market Share Trend](#)

Market

[Top Site Usage](#)
[Market Position](#)

Breakdown

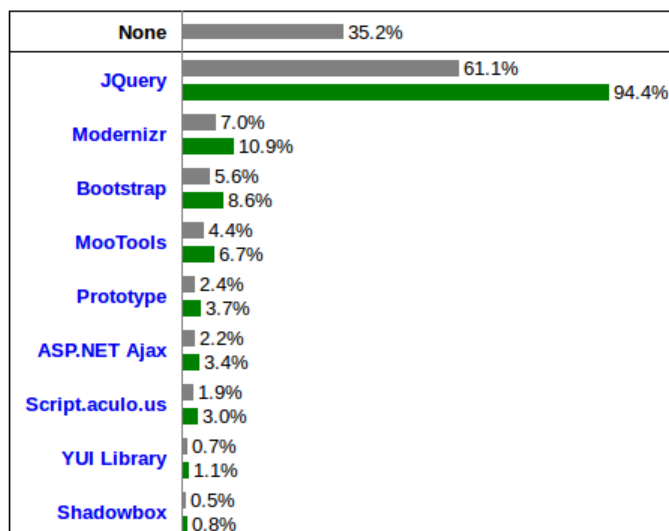
[Technologies](#) > [JavaScript Libraries](#)

Usage of JavaScript libraries for websites

This diagram shows the percentages of websites using various JavaScript libraries. See [technologies overview](#) for explanations on the methodologies used in the surveys. Our reports are updated daily.

How to read the diagram:

35.2% of the websites use none of the JavaScript libraries that we monitor. JQuery is used by 61.1% of all the websites, that is a JavaScript library market share of 94.4%.



Request an extensive
market report of specific
JavaScript libraries.

[Learn more](#)

Technology Brief

JavaScript Libraries

JavaScript libraries provide a basis for building cross-browser web applications based on JavaScript.

Featured Products and Services



Premium Managed
WordPress Hosting

Try it for 60 Days

Secure Premium Wordpress Hosting
[WP Engine](#)

[Present your product or service here](#)

Cài đặt jQuery

- Download thư viện jQuery (tệp .js) tại jquery.com
- Bao hàm tệp thư viện jQuery trong trang HTML

```
<script type="text/javascript" src="jquery.js"></script>
```

Cú pháp jQuery

- jQuery cho phép **chọn** các đối tượng tài liệu rồi thực hiện **hành động** trên đối tượng được chọn
- Cú pháp: **`$(selector).action();`**
- jQuery sử dụng bộ chọn như CSS

Ví dụ

- `$("#test").hide()`



```
document.getElementById("test").hide();
```

Ví dụ

- `$("p").hide()`



```
arr = document.getElementsByTagName("p");  
for (i = 0; i < arr.length; i++)  
    arr[i].hide();
```

Ví dụ

- `$(".test").hide()`



```
arr = document.getElementsByTagName("*");  
for (i = 0; i < arr.length; i++)  
    if (arr[i].className == "test")  
        arr[i].hide();
```

Lưu ý

- Cần sử dụng jQuery khi toàn bộ mã trang đã được tải

```
$(document).ready(function(){
```

```
// All jQuery methods go here...
```

```
});
```

Xử lý sự kiện với jQuery

Kích hoạt sự kiện
Bắt và xử lý sự kiện

Xử lý sự kiện

- Kích hoạt sự kiện trên đối tượng tài liệu
`$(selector).eventName();`
- Bắt và xử lý sự kiện trên các đối tượng tài liệu
`$(selector).eventName(function(args){
 // action goes here!!
});`

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

Ví dụ

- Xử lý sự kiện kích chuột trên nút bấm

```
$("#test").click(function () {  
    alert("hello!");  
});
```



```
<script type="text/javascript">  
    function dotest() { alert("hello!");}  
</script>  
<input type = "button" id = "test"  
onclick = "javascript:dotest();" />
```

Ví dụ

- Khi người dùng gõ enter trên một ô nhập thì chuyển tâm điểm sang ô nhập tiếp theo

```
$("#name").keyup( function(e) {  
    if (e.keyCode ==13) $("#email").focus();  
});
```



```
<script type="text/javascript">  
    function processKeyup(e) {  
        if (window.event) e= window.event;  
        if (e.keyCode == 13)  
            document.getElementById("email").focus();  
    }  
</script>  
<input type="text" id = "name"  
        onKeyup = "javascript:processKeyup(event);" />
```

Đọc và thay đổi thuộc tính các đối tượng tài liệu với jQuery

Đọc thuộc tính đối tượng tài liệu

- Đọc nội dung văn bản của đối tượng tài liệu
`$(selector).text();`
- Đọc nội dung HTML của đối tượng tài liệu
`$(selector).html();`
- Đọc giá trị của đối tượng tài liệu
`$(selector).val()`
- Đọc giá trị thuộc tính của đối tượng tài liệu
`$selector().attr(att);`

Ví dụ đọc text và html

```
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
        alert("HTML: " + $("#test").html());
    });
});
</script>
```

Ví dụ đọc giá trị

```
<script>
```

```
$(document).ready(function(){  
    $("button").click(function(){  
        alert("Value: " + $("#test").val());  
    });  
});
```

```
</script>
```

```
<p>Name: <input type="text" id="test"  
    value="Mickey Mouse"></p>
```

```
<button>Show Value</button>
```

Ví dụ đọc thuộc tính

```
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert($("#w3s").attr("href"));
  });
});
</script>
<p><a href="http://www.w3schools.com"
  id="w3s">W3Schools.com</a></p>
<button>Show href Value</button>
```

Đặt thuộc tính đối tượng tài liệu

- Đặt nội dung văn bản của đối tượng tài liệu
`$(selector).text("new text");`
`$(selector).text(function(i, oldText) {return newText});`
- Đặt nội dung HTML của đối tượng tài liệu
`$(selector).html("new html");`
`$(selector).html(function(i, oldHtml) {return newHtml});`
- Đặt giá trị của đối tượng tài liệu
`$(selector).val("new value");`
`$(selector).val(function(i, oldValue) {return newValue});`
- Đặt giá trị thuộc tính của đối tượng tài liệu
`$selector().attr("attr1", "new value");`
`$selector().attr({"attr1":"new value1", "attr2":"new value2", ...});`
`$(selector).attr("attribute", function(i, oldValue) {return newValue});`

Ví dụ đặt thuộc tính

```
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello world!");  });
  $("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");  });
  $("#btn3").click(function(){
    $("#test3").val("Dolly Duck");  });
});
</script>
```

Ví dụ đặt thuộc tính

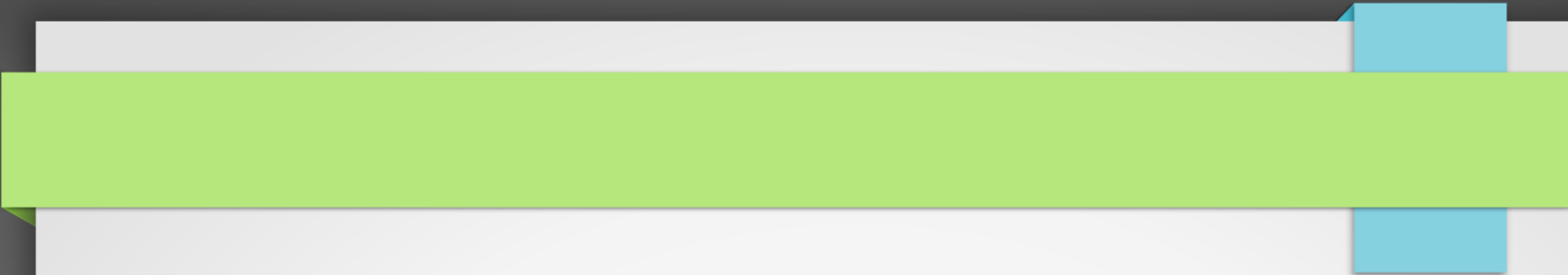
```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#w3s").attr({
            "href" :
            "http://www.w3schools.com/jquery",
            "title" : "W3Schools jQuery Tutorial"
        });
    });
});
```

Ví dụ đặt thuộc tính với hàm gọi lại

```
<script>
```

```
$(document).ready(function(){  
    $("#btn1").click(function(){  
        $("#test1").text(function(i,origText){  
            return "Old text: " + origText + " New text:  
Hello world! (index: " + i + ")";  
        });  
    });  
});
```

```
$("#btn2").click(function(){  
    $("#test2").text(function(i,origText){
```



Xử lý CSS với jQuery

Đọc và thay đổi từng thuộc tính CSS

- Đọc thuộc tính CSS

```
$(selector).css("propertyname");
```

- Đặt thuộc tính CSS

```
$(selector).css({"propertyname":"value","propertyname":"value",...});
```

Thay đổi lớp CSS

- Thêm lớp CSS được áp dụng
`$(selector).addClass("cssclass");`
- Bỏ lớp CSS được sử dụng
`$(selector).removeClass("cssclass");`
- Bật/tắt lớp CSS được sử dụng
`$(selector).toggleClass("cssclass");`

Ví dụ thay đổi lớp CSS

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").addClass("important blue");
    });
});
</script>
<style type="text/css">
.important { font-weight:bold; font-size:xx-
large; }
```

Thêm mới, loại bỏ đối tượng tài liệu với jQuery

Thêm đối tượng con

- Thêm đối tượng con vào đầu (thành con cả)
`$(selector).prepend(child1 [, child2, child3, ...])`
- Thêm đối tượng con vào cuối (thành con út)
`$(selector).append(child1 [, child2, child3, ...])`

- Ví dụ 1

```
$("p").prepend("<b>Appended text</b>.", "hello");
```

- Ví dụ 2

```
var txt1="<p>Text.</p>";           // Create element  
with HTML
```

```
var txt2=$("<p></p>").text("Text."); // Create with  
jQuery
```

Thêm đối tượng anh/em

- Thêm nút anh liền trước
`$(selector).before(presibling1 [, presibling2, presibling3, ...])`
- Thêm đối tượng con vào cuối (thành con út)
`$(selector).after(nextsibling1 [, nextsibling2, nextsibling3, ...])`
- Ví dụ 1
`$("p").before" <b>Appended text</b>.", "hello");`
- Ví dụ 2
`var txt1="<p>Text.</p>";` // Create element with HTML

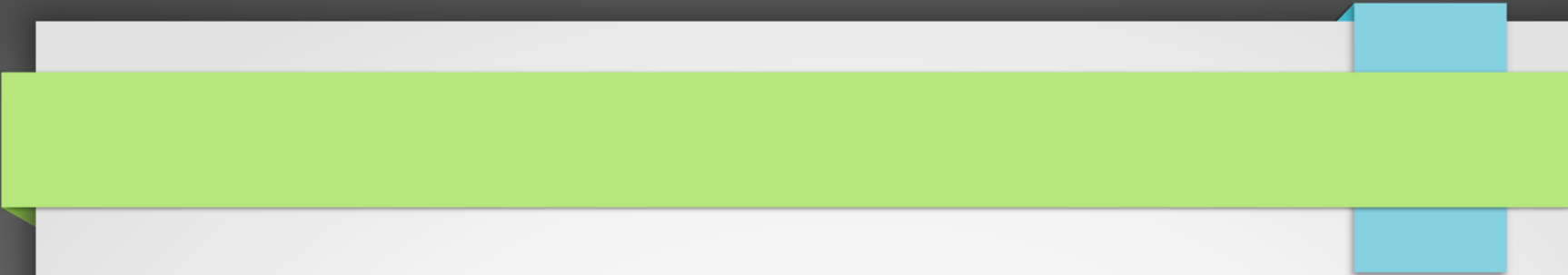
Xóa đối tượng

- Xóa tất cả các đối tượng con của đối tượng được chọn

`$(selector).empty();`

- Xóa đối tượng được chọn và tất cả các đối tượng con của đối tượng được chọn

`$(selector).remove();`



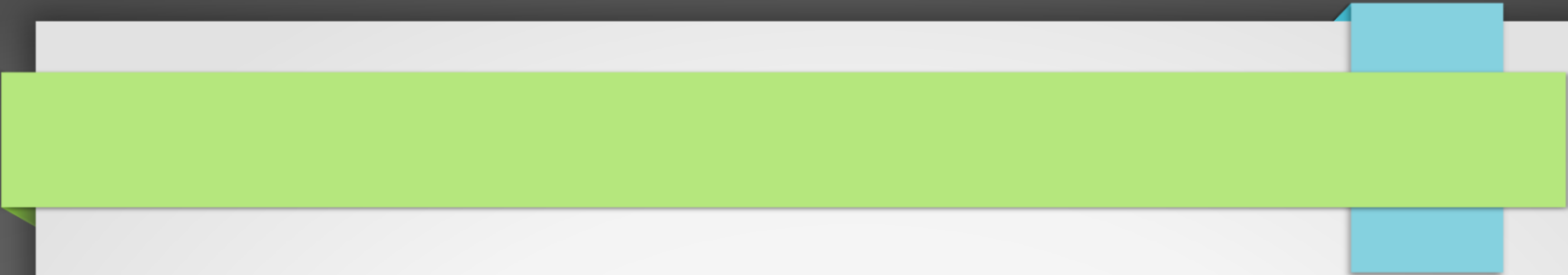
Duyệt DOM với jQuery

Các phương thức duyệt DOM

- `$(selector).parent()`
- `$(selector).parents()`
- `$(selector).parents(selector_filter)`
- `$(selector).parentsUntil(selector_stop)`
- `$(selector).parentsUntil(selector_stop, selector_filter)`
- `$(selector).children()`
- `$(selector).children(selector_filter)`
- `$(selector).find()`
- `$(selector).find(selector_filter)`
- `$(selector).siblings()`
- `$(selector).siblings(selector_filter)`
- `$(selector).next()`
- `$(selector).next(selector_filter)`
- `$(selector).nextAll()`
- `$(selector).nextAll(selector_filter)`
- `$(selector).nextUntil(selector_stop)`
- `$(selector).nextUntil(selector_stop, selector_filter)`
- `$(selector).prev()`
- `$(selector).not(selector_filter)`
- ...

Ví dụ duyệt DOM

```
<!DOCTYPE html><html><head>
  <title>jQuery</title>
  <meta charset="utf-8">
</head><body>
  <div>div not selected</div>
  <span>span not selected</span>
  <div>div (current element)
    <p>p (child)
      <span>span (grandchild)</span>
    </p>
    <p>p (child)
      <span>span (grandchild)</span>
    </p>
  </div>
  <script type="text/javascript" src="jquery-3.2.1.min.js"></script>
  <script type="text/javascript" >
    $(document).ready(function(){
      $("div").find("span").css({"color": "red"});
    });
  </script>
</body></html>
```



Tạo hiệu ứng với jQuery

Các phương thức tạo hiệu ứng

`$(selector).hide([duration] [, callback])`
`$(selector).show([duration] [, callback])`
`$(selector).toggle([duration] [, callback])`
`$(selector).fadeIn([duration] [, callback])`
`$(selector).fadeOut([duration] [, callback])`
`$(selector).fadeToggle([duration] [, callback])`
`$(selector).fadeTo(duration, opacity [, callback])`
`$(selector).slideDown([duration] [, callback])`
`$(selector).slideUp([duration] [, callback])`
`$(selector).slideToggle([duration][, callback])`
`$(selector).animate({cssProperties} [, duration] [, callback])`
`$(selector).stop([clearQueue] [, jumpToEnd])`

Ví dụ tạo hiệu ứng

```
<!DOCTYPE html><html><head>
  <title>jQuery</title><meta charset="utf-8">
</head><body>
  <button id="button1">Start Animations</button>
  <button id="button2">Stop Current Animation</button>
  <button id="button3">Stop All Animations</button>
  <div style="background:#98bf21; height:100px;
    width:100px; position:relative;"></div>
  <script type="text/javascript" src="jquery-3.2.1.min.js"></script>
  <script type="text/javascript" >
    $(document).ready(function() {
      $("#button1").click(function() {
        $("#div").animate({
          left: '250px',
          height: '+=150px',
          width: '350px'
        })
        .animate({left: '0px'}, "slow")
        .animate({height:'-=150px'}, "slow")
        .fadeOut(500);
      });
      $("#button2").click(function() { $("#div").stop(); });
      $("#button3").click(function() {
        $("#div").stop(true, true);
      });
    });
  </script>
</body></html>
```

Sử dụng jQuery để xử lý ajax

load(), \$.get(), \$.post(), \$.ajax()

jQuery AJAX Get

`$.get(path?querystring, [callback]);`

- Gửi yêu cầu lên server theo phương thức GET và nhận kết quả về theo AJAX, sau đó thực hiện hàm callback
 - path: Địa chỉ tệp được yêu cầu
 - callback: Hàm được gọi sau khi tải xong. Hàm có hai tham số là *nội dung*, *trạng thái* trả về

Ví dụ jQuery AJAX Get

- Yêu cầu nội dung trang "clone.php?v1=100&v2=101" với các tham số theo phương thức GET, đặt nội dung nhận được cho đối tượng tài liệu có định danh test1.

```
$.get("clone.php?v1=100&v2=101",  
    function (data, status) {  
        $("#test1").html(data);  
    });
```

jQuery AJAX Post

`$.post(URL, [data], [callback]);`

- Gửi yêu cầu lên server theo phương thức POST và nhận kết quả về theo AJAX, sau đó thực hiện hàm callback
 - url: Địa chỉ tệp được yêu cầu
 - data: Các cặp tham số/giá trị được gửi
 - callback: Hàm được gọi sau khi tải xong. Hàm có hai tham số là *nội dung*, *trạng thái* trả về

Ví dụ jQuery AJAX Post

- Yêu cầu nội dung trang "clone.php" với các tham số v1=10, v2=12 theo phương thức POST, đặt nội dung nhận được cho đối tượng tài liệu có định danh test2.

```
$.post("clone.php", {v1: "10", v2: "12"},  
      function (data, status) {  
          $("#test2").html(data);  
      });
```

jQuery AJAX Load

`$(selector).load(URL, [data], [callback]);`

- Tải nội dung từ URL và đặt vào đối tượng được chọn (Tương đương gửi yêu cầu lên server theo phương thức POST và nhận kết quả về theo AJAX, sau đó đặt kết quả trả về vào innerHTML của đối tượng được chọn), sau đó thực hiện hàm callback
 - url: Địa chỉ tệp được yêu cầu
 - data: Các cặp tham số/giá trị được gửi cùng url
 - callback: Hàm được gọi sau khi tải xong. Hàm có ba tham số là *nội dung*, *trạng thái* trả về và đối tượng *XMLHttpRequest* đã thực hiện các công việc của hàm load

Ví dụ jQuery AJAX Load

- Tải nội dung tệp văn bản “text.txt” và đặt vào innerHTML của đối tượng có định danh test3
`$("#test3").load("text.txt");`
- Tải nội dung tại “clone.php?v1=8&v2=9” và đặt vào innerHTML của đối tượng có định danh test4
`$("#test4").load("clone.php", {v1: "8", v2: "9" });`

Ví dụ jQuery AJAX Load

- Tải nội dung tệp văn bản “text.txt” và đặt vào innerHTML của đối tượng có định danh test3
`$("#test3").load("text.txt");`
- Tải nội dung tại “clone.php?v1=8&v2=9” và đặt vào innerHTML của đối tượng có định danh test4
`$("#test4").load("clone.php", {v1: "8", v2: "9" });`

jQuery ajax

- `$.ajax(url [, settings ]);`
- Settings:
 - type: POST/GET/HEAD, ...
 - data: {Các cặp tham số/giá trị}
 - contentType: Kiểu nội dung và mã hóa được sử dụng
 - dataType: Kiểu dữ liệu muốn nhận về từ server (xml, json, script, html)
 - success: function (data, status, jqXHR) { },
 - error: function (jqXHR, status, errorThrown) { }
 - ...

jQueryUI

- jQueryUI (<http://jqueryui.com>) là mở rộng của jQuery cho mục đích làm giao diện
 - Tương tác: Kéo thả, thay đổi kích thước, lựa chọn, sắp xếp, ...
 - Widget: autocomplete, menu, spinner, slider, ...
 - Hiệu ứng: hiển thị, hoạt cảnh, tô màu, ...



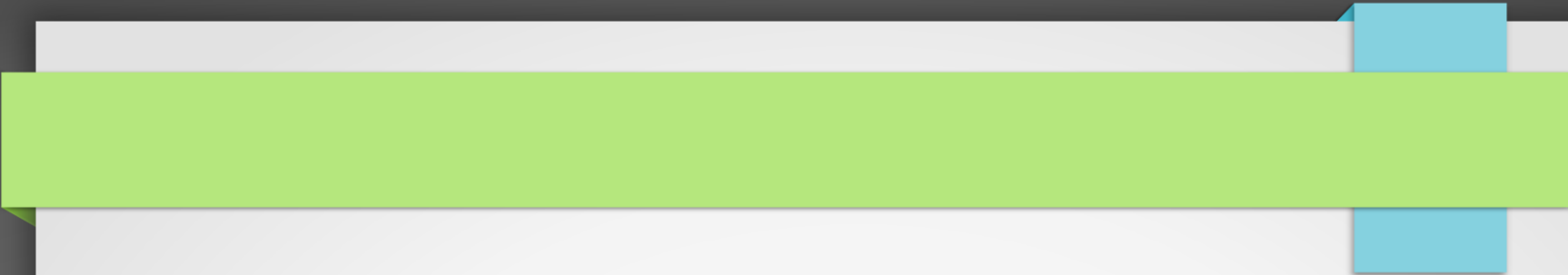
Tự
học

jQuery Mobile

- jQuery Mobile (<http://jquerymobile.com>)
- là mở rộng của jQuery, dùng để phát triển web trên thiết bị di động



Tự
học



Tiếp theo

Bootstrap