**1-1 什么是 jQuery 和 DOM**

[jQuery](https://jquery.com/)是一套最多人使用的 JavaScript 函式库，可以撰写各种网页效果，有丰富的社区和各种 Plugins 外挂。

它的主要作用是操纵网页的 DOM(Document Object Model)：浏览器会解析 HTML 然后显示在画面上，这个解析出来的结构就叫做 DOM，在浏览器内部这是一个树状的资料结构：

html

- head

- title

- a web page

- body class="home"

- h1 id="header"

- a headline

- p

- some

- span

- text

- text

浏览器提供了 DOM API 让开发者可以操纵这些结构，透过 jQuery 我们可以很方便地从 HTML 取得要操纵的 HTML 元素，并在 DOM 里面进行插入、修改、删除元素，这样浏览器的画面就会跟着变化了。

**1-2 新增练习网页**

jQuery 在 Rails 5.0 以前的版本默认就有安装，你可以在 Gemfile 里面看到 gem 'jquery-rails' 以及在app/assets/javascripts/application.js 里面看到 //= require jquery。

在 Gemfile 前几行可以看到你 Rails 的版本。如果你是 Rails 5.1，请在 Gemfile 补上 gem 'jquery-rails'，然后执行 bundle，接着在 app/assets/javascripts/application.js 里面补上 //= require jquery。

让我们新增一个专案，并新增一些页面来进行示范：

rails new ajax-posting-app

cd ajax-posting-app

git init

rails g controller pages

编辑 config/routes.rb

config/routes.rb

get "/jquery-1" => "pages#jquery\_1"

get "/jquery-2" => "pages#jquery\_2"

get "/jquery-3" => "pages#jquery\_3"

get "/jquery-4" => "pages#jquery\_4"

get "/jquery-5" => "pages#jquery\_5"

root "pages#jquery\_1"

编辑 app/views/layouts/application.html.erb

app/views/layouts/application.html.erb

<body>

+ <%= link\_to "jQuery 练习一", jquery\_1\_path %> |

+ <%= link\_to "jQuery 练习二", jquery\_2\_path %> |

+ <%= link\_to "jQuery 练习三", jquery\_3\_path %> |

+ <%= link\_to "jQuery 练习四", jquery\_4\_path %> |

+ <%= link\_to "jQuery 练习五", jquery\_5\_path %>

安装 [bootstrap-sass](https://github.com/twbs/bootstrap-sass) (步骤省略，之后的截图是有安装 Bootstrap，画面比较漂亮，没做不影响功能)

启动服务器 rails server

(请接着做下一节再打开浏览器)

# 1-3 如何动态增加内容到网页上

目标：点击一个 HTML 元素后，动态置换一段内容

新增 app/views/pages/jquery\_1.html.erb

<p>

<a id="my-click">Click Me</a>

</p>

<div id="foo" style="border: 1px solid red;">

<p>bar</p>

</div>

<script>

$("#my-click").click(function(){

$("#foo").html('<h1>zoo</h1>');

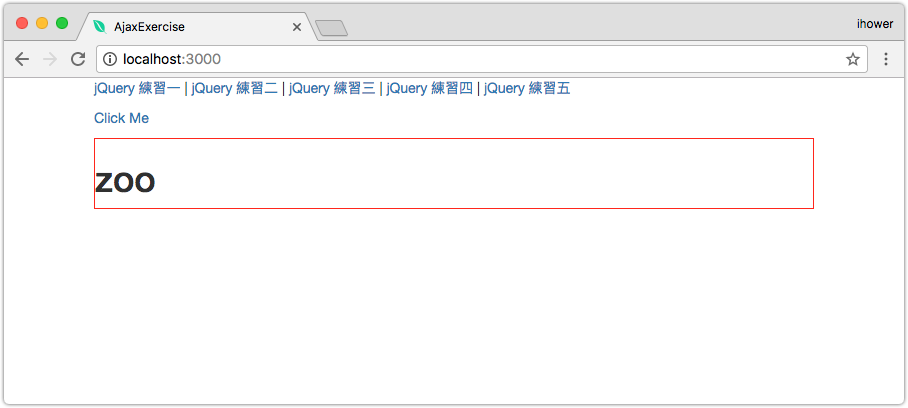
})

</script>

浏览 http://localhost:3000/jquery-1

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/6MKTXNchSfSt6YMinFiK_1.png)

点击 Click Me 连结的话，你会发现 <p>bar</p> 就动态置换成 <h1>zoo</h1> 了。

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/X6DsK83eQqOrdYTjXuH9_2.png)

解说：

* <script>...</script> 包起来的部分就是写 JavaScript 的地方
* 这个钱号 $ 等同于 jQuery
* $("#my-click") 是 jQuery 的选择器用法，会挑出 id 是 my-click 的元素，在这里就是指 Click Me 的超连结。在 HTML 上 id 是唯一的，不能有重复的 id。
* .click( function(){...} ) 会绑订一个点击(click)事件在该元素上面，当用户点击这个元素时，就会执行里面的 function
* $("#foo").html('<h1>zoo</h1>') 会把 #foo 这个元素的内容置换成 <h1>zoo</h1>

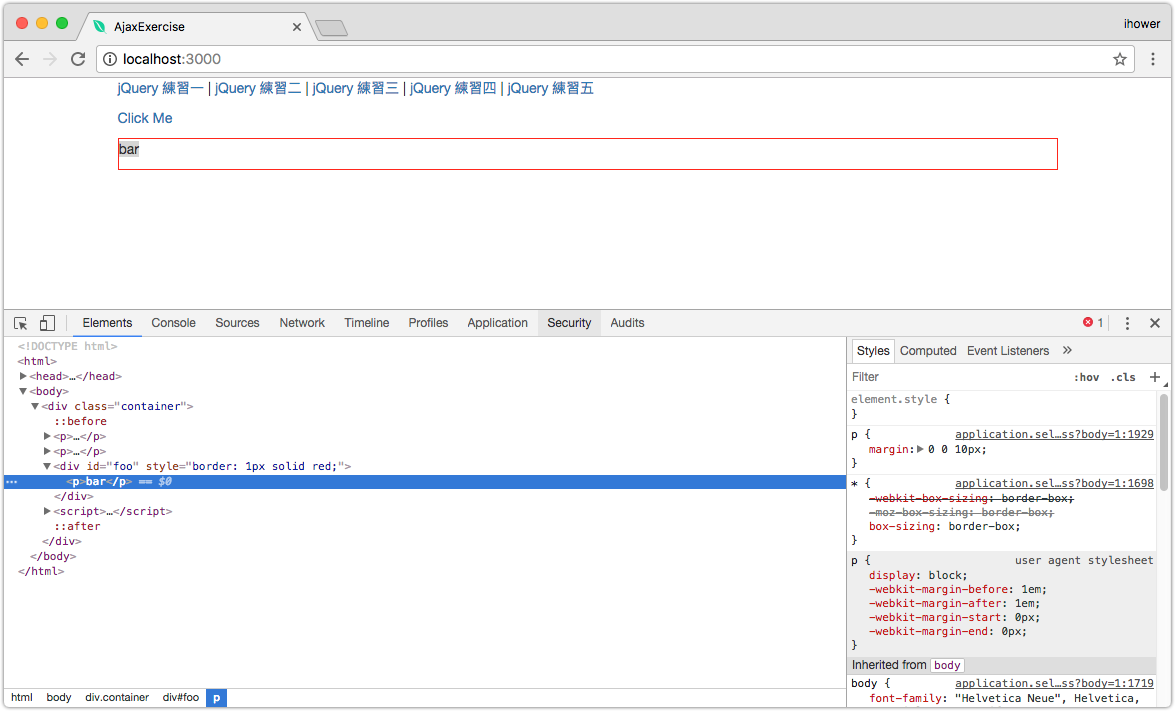
请试试看把 html 换成以下用法，观察看看有什么差别：

* text 文字替换
* prepend 把内容插在指定元素里面的最前面
* append 把内容插在指定元素里面的最后面
* before 把内容插在指定元素的前面
* after 把内容插在指定元素的后面

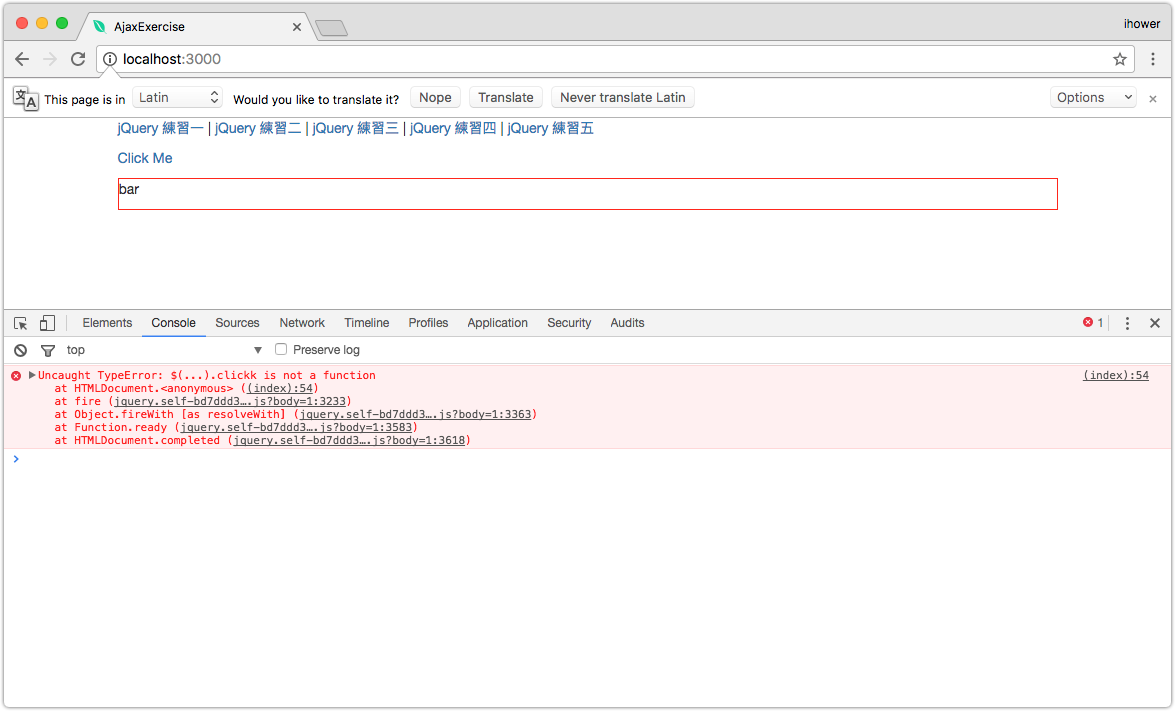
请先搭配看下一节如何除错。

# 1-4 如何除错?

请使用 Chrome 开发者工具：在 HTML 画面上按右键，点 Inspect，就会看到以下画面：

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/9qkFqQi5QMKpl8xwKxpY_3-debug.png)

如果 JavaScript 写错的话，在 Console 可以看到错误，例如我们把 click 拼错成 clickk 的话：

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/qgWeEcpRdK8tHhTxh5KA_4-debug.png)

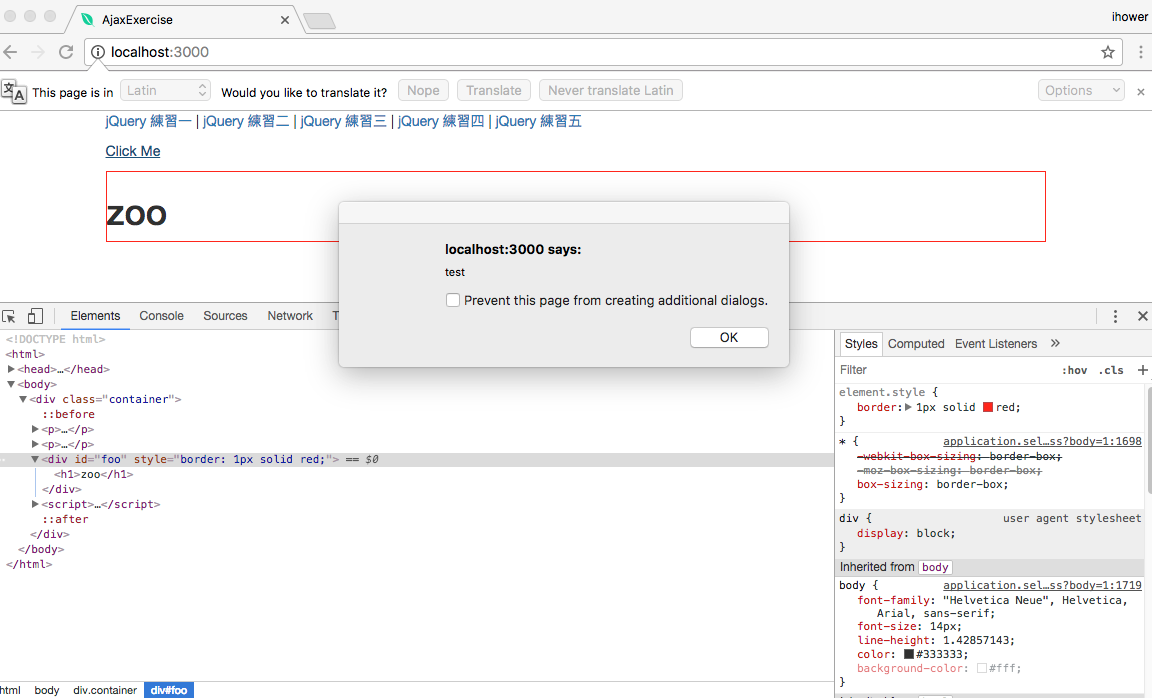
从错误的地方开始，以下的 JavaScript 程序都不会执行喔

想在 JavaScript 程序中观察变量的话，可以用 alert 或 console.log 语法，例如：

$("#my-click").click(function(){

alert("test");

})

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/PNwSIeZqRE2uQYM2EgvY_5-debug.png)

不过跳 alert 有时候太恼人了，可以改用 console.log 就会出现在开发者工具的 Console 里面：

console.log("checkpoint A");

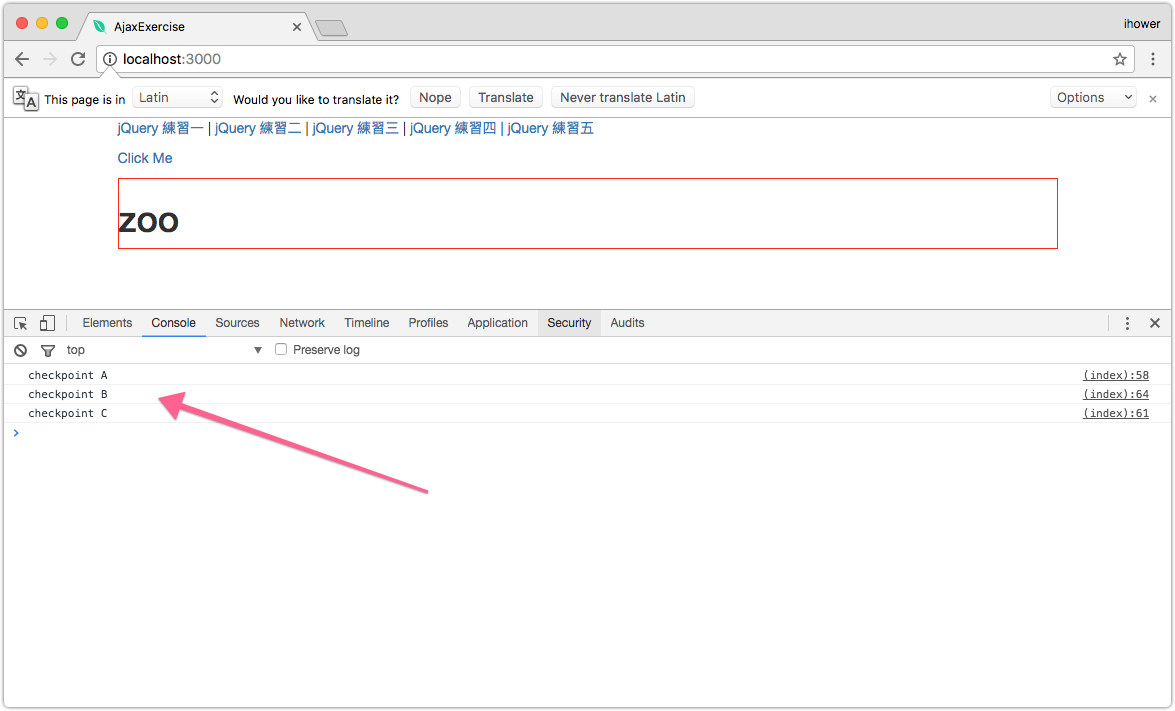
$("#my-click").click(function(){

console.log("checkpoint C");

})

console.log("checkpoint B");

就会出现：

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/J3Eu05hHSgSSTIiZfuR5_6-debug.png)

# 1-5 document.ready 事件解说

在用 $("xxxx") jQuery 选择器的时候，HTML 必须先加载完，jQuery 才能够找得到元素。因此我们刚刚是先写 HTML 在上，把 script 写在下面。如果倒过来的话：

<script>

$("#my-click").click(function(){

$("#foo").html('<h1>zoo</h1>');

})

</script>

<p>

<a id="my-click">Click Me</a>

</p>

<div id="foo" style="border: 1px solid red;">

<p>bar</p>

</div>

你会发现没有效果，点了没反应，因为 click事件并没有顺利绑在超连结上。

但是传统上我们习惯将 JavaScript 放在 HTML 上方加载，例如写在app/assets/javascripts/application.js 里面，因此这时候我们需要用 $(document).ready 把程序包起来，也就是改成：

<script>

+ // 这是 javascript 注解

+ // 也可以缩写成 $(function() {

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

$("#my-click").click(function(){

$("#foo").html('<h1>zoo</h1>');

})

+ })

</script>

<p>

<a id="my-click">Click Me</a>

</p>

<div id="foo" style="border: 1px solid red;">

<p>bar</p>

</div>

$(document).ready 是一个浏览器的事件，会在加载 HTML 完成之后，才执行里面的 function，就可以解决这个问题。

**1-6 如何隐藏和打开内容**

目标：点击一个元素后，动态隐藏和打开一段内容

新增 app/views/pages/jquery\_2.html.erb

<p>

<a id="my-open-click">Open</a>

</p>

<div id="foo" style="border: 1px solid red;">

<a id="my-hide-click">Hide</a>

<p>bar</p>

</div>

<script>

$("#foo").hide();

$("#my-open-click").click(function(){

$("#foo").show();

})

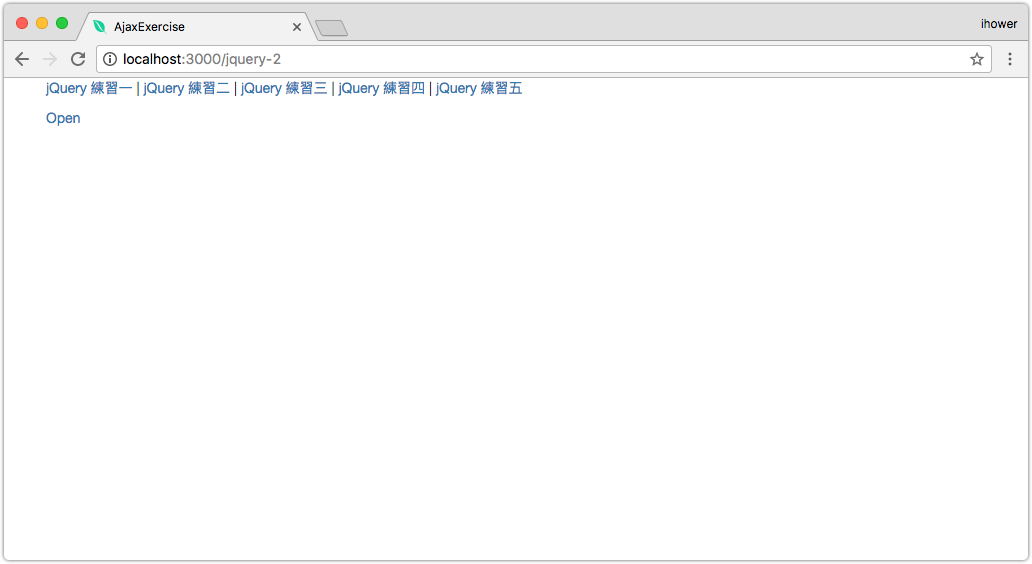
$("#my-hide-click").click(function(){

$("#foo").hide();

})

</script>

浏览 http://localhost:3000/jquery-2

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/A8kZpbE2RhuU8Abav4Ol_7-hide.png)

点击 Open

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/Dj5ejPATbSZslH3EGwnV_8-show.png)

再点击 Hide 就会隐藏起来了。

请试试看把 hide 和 show 分别改成

* fadeOut 和 fadeIn
* slideUp 和 slideDown

会有不同的动画效果。

**1-7 鼠标移过去自动打开和隐藏内容**

目标：鼠标移过去，就会自动打开和隐藏一段内容

新增 app/views/pages/jquery\_3.html.erb

<p>

<a id="my-toggle-click">Toggle</a>

</p>

<div id="foo" style="border: 1px solid red;">

<p>bar</p>

</div>

<script>

$("#foo").hide(); // 先马上藏起来

$("#my-toggle-click").hover(function(){

$("#foo").toggle(); // 移过去会有开关的效果

})

</script>

浏览 http://localhost:3000/jquery-3

鼠标移过去就会打开内容，鼠标移走内容就会隐藏起来。

解说：

这里我们改用 hover 事件，这会在鼠标移过去和移出去的时候，都会触发。然后搭配 toggle 方法就可以一开一关了。

jQuery 支援了非常多的[浏览器事件](https://www.w3schools.com/jquery/jquery_events.asp)，从鼠标单点、双点、移过去、移出去，到键盘输入哪一个按键等等，在浏览器里面发生的事情都可以绑事件上去。

最后，请试试看把 toggle 改成

* slideToggle
* fadeToggle

会有不同的动画效果。

# 1-8 如何替换和移除内容

目标：点击一个元素后，动态替换和删除内容

新增 app/views/pages/jquery\_4.html.erb

<p>

<a id="my-replace-click">Replace</a>

<a id="my-remove-click">Remove</a>

<a id="my-remove-all-click">Remove All</a>

</p>

<div id="foo" style="border: 1px solid red;">

<p>bar</p>

</div>

<div class="zoo"><p>a</p></div>

<div class="zoo"><p>b</p></div>

<div class="zoo"><p>c</p></div>

<script>

$("#my-replace-click").click(function(){

$("#foo").replaceWith('<div id="foo" style="border:1px solid green"><p>foo</p></div>');

})

$("#my-remove-click").click(function(){

$("#foo").remove();

})

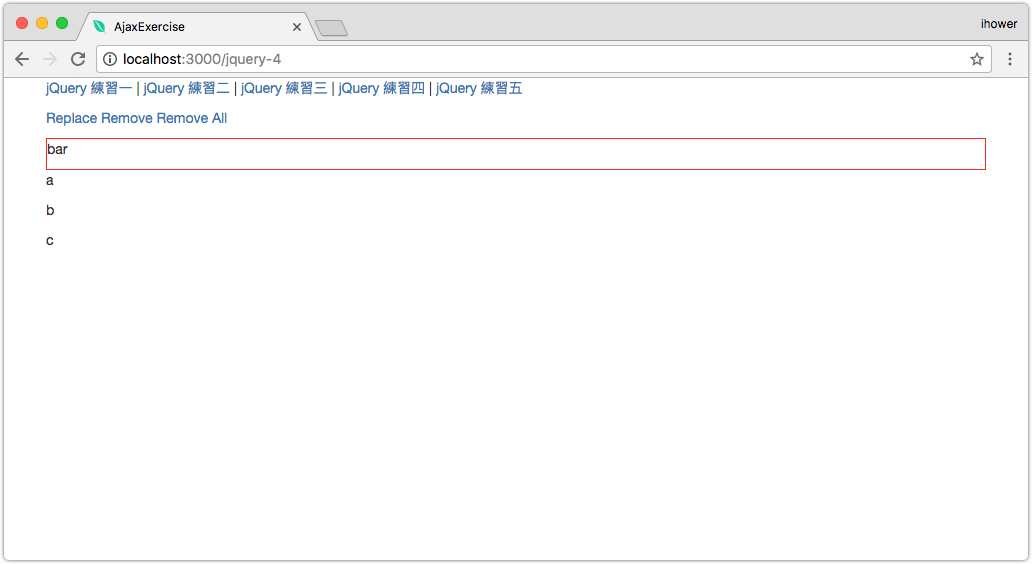
$("#my-remove-all-click").click(function(){

$(".zoo").remove();

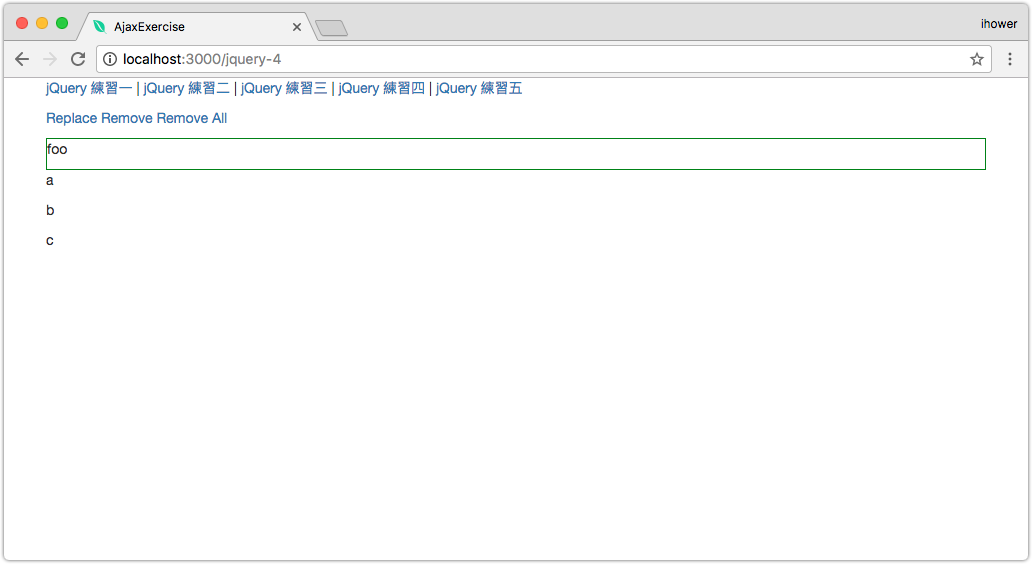
})

</script>

浏览 http://localhost:3000/jquery-4

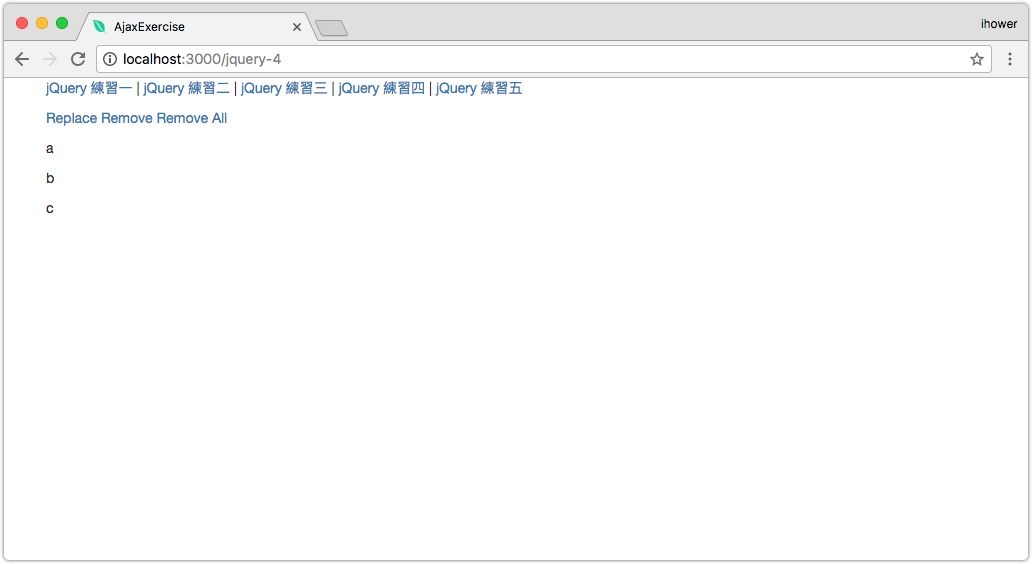
[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/cDDCsqm5TKWHX7uAAvGY_9.png)

点击 Replace

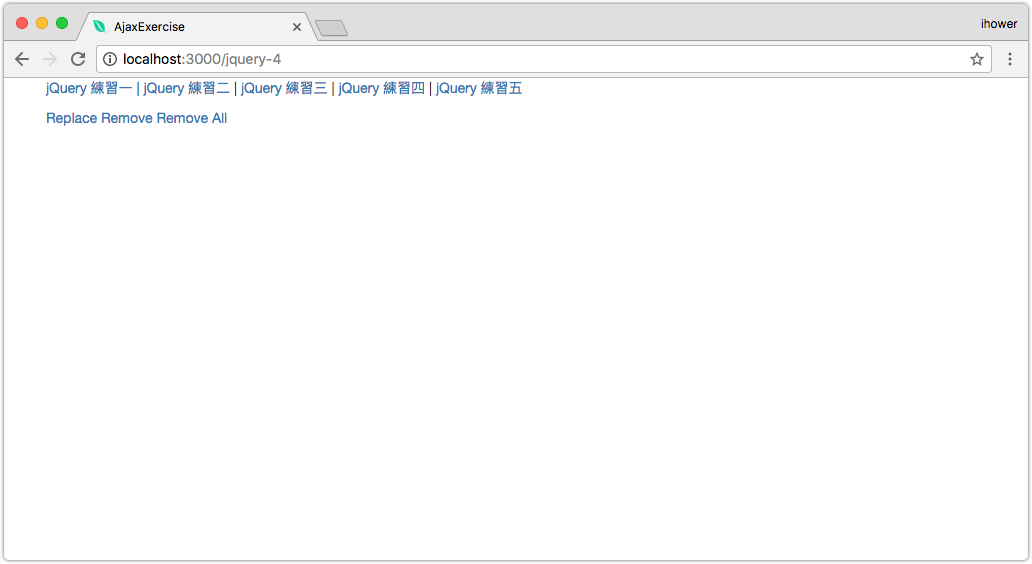
[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/2oVKMzN0R7uPSqLZTXAf_10-replace.png)

解说：replaceWith 会连同指定元素一起置换掉，因此整个 #foo 包含 <div> 都一起被换掉了。

点击 Remove 移除一个元素：

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/zITaXYjRoKS2XHZbVwlr_11-remove.png)

点击 Remove All 移除 a,b,c 三个元素：

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/hFLME8DwR8O04ETxfyJg_12-remove-all.png)

解说：这里我们改用 $(".zoo") 选择器来挑选出所有 class 里面有 zoo 的元素。跟 id 不同，一份 HTML 上可以有多个重复的 class。

# 1-9 如何改变 class 属性

目标：点击一个元素后，动态替换元素的 class，这样就会套用不同的 CSS 来改变外观

新增 app/views/pages/jquery\_5.html.erb

<style>

.error {

color: red;

}

.success {

color: green;

}

</style>

<p>

<a id="my-add-class-click">Add Class</a>

<a id="my-remove-class-click">Remove Class</a>

</p>

<div id="foo">

<h1>bar</h1>

</div>

<script>

$("#my-add-class-click").click(function(){

$("#foo").removeClass('error');

$("#foo").addClass('success');

})

$("#my-remove-class-click").click(function(){

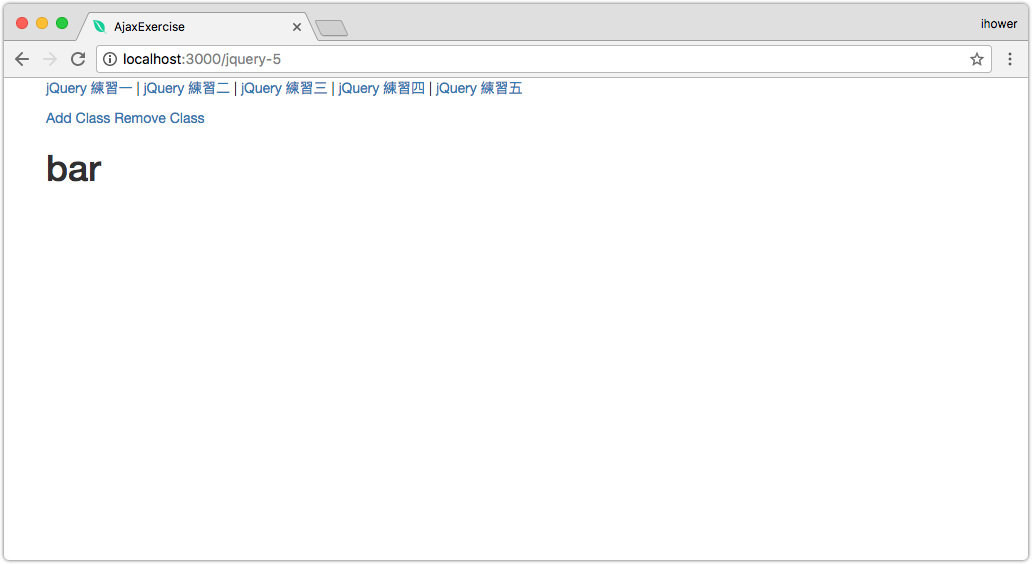
$("#foo").removeClass('success');

$("#foo").addClass('error');

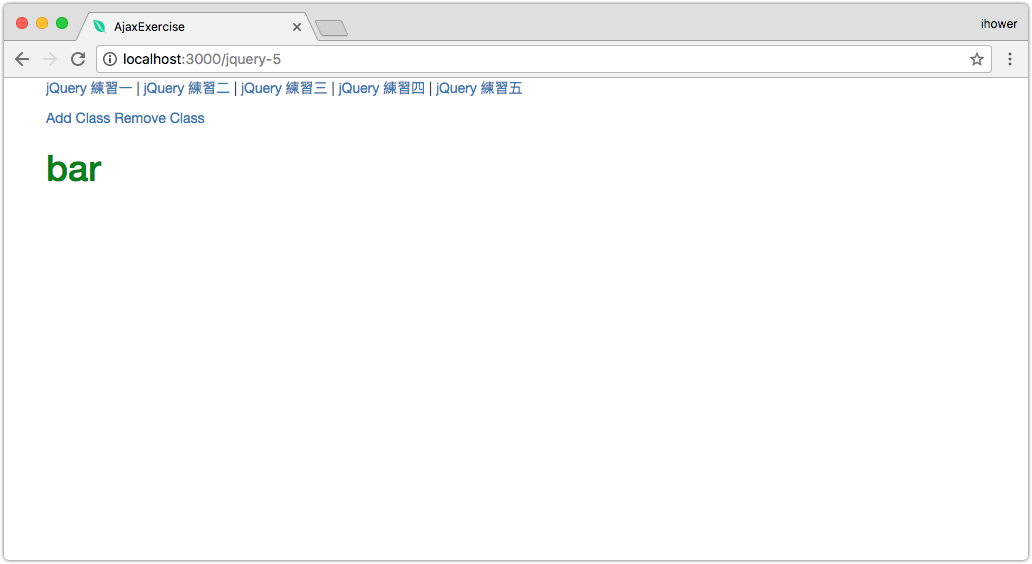
})

</script>

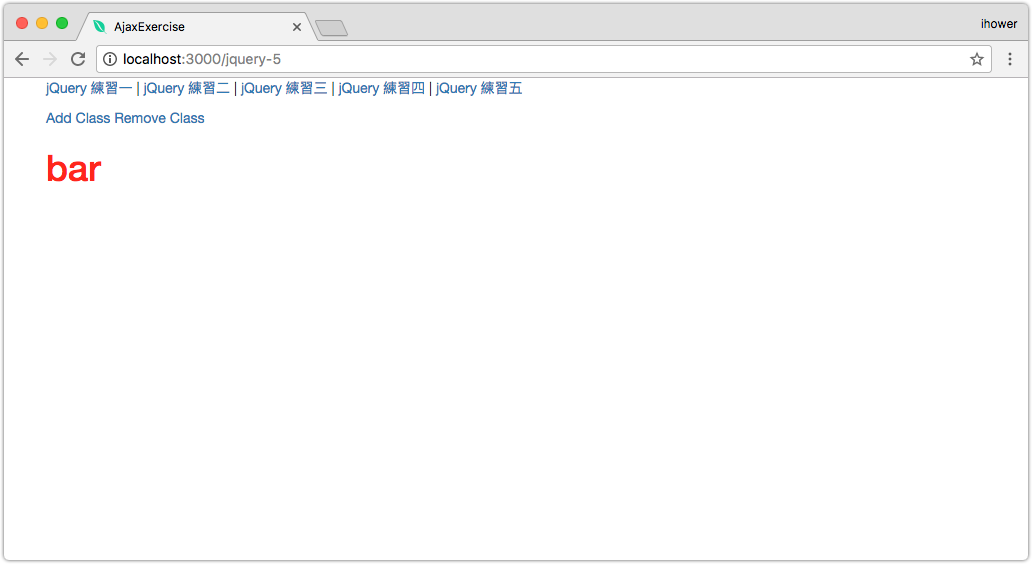
浏览 http://localhost:3000/jquery-5

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/KSyQmOkRmey4nBZmdQSR_13-class.png)

点击 Add Class

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/EiAAw6KTJuJF31ns3B2H_14.png)

点击 Remove Class

[](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/mLwXUn7HQh28w0aHqlLF_15.png)

jQuery 也有 API 可以[直接改 CSS](http://api.jquery.com/css/)。

**1-10 jQuery 小结**

这章示范的就是最常见的 jQuery 使用套路了，包括：

* 用选择器 Selector 选取出目标元素，我们学到最简单的可以用 id 或 class 来选，例如 $("#某个ID")和 $(".某个class")。
* 绑定一个事件(Event)上去，例如 click 和 hover 事件
* 当事件被触发的时候，执行某个 DOM 操作来改变 HTML，可能是改变元素的属性、插入新内容、新增或移除 class 属性等等。目前学到的用法有：
  + html
  + append
  + prepend
  + before
  + after
  + remove
  + replaceWith
  + toggle
  + addClass 和 removeClass

其实大部分的 jQuery 代码的工作就是去新增内容、改变 HTML、调整 CSS 等方式操控网页，例如跳出选单、轮播广告、滑动标题图片等等，这些就是执行上述步骤。