**คูมือสำหรับการสอน jQuery ภาษาไทย**

**jQuery** เป็นการบรรจุเอาคำสั่งต่างๆของ JavaScript Library มาสร้างเป็นฟังก์ชั่นใหม่ในรูปแบบของ โFramework เพื่อให้ง่ายและเพิ่มประสิทธิภาพต่อการใช้งาน และไม่ต้องมาเขียนใหม่เองทั้งหมดตั้งแต่ต้น อย่างเช่นการเขียนโปรแกรมในรูปแบบ Ajax ก็สามารถทำได้แบบง่ายๆเพียง code ไม่กี่บรรทัด หรือจะเขียนโปรแกรมดัก Event (เหตุการณ์) ต่างๆ เช่น การ click, rollover, mouse ก็เขียนได้ง่ายขึ้นและลดปัญหาความไม่เข้ากันของเบราเซอร์ที่ต่างกัน

**jQuery Course Outline**

**หลักสูตร jQuery ระยะเวลาการอบรม 24 ชั่วโมง**

**พื้นฐานผู้เข้าอบรม**

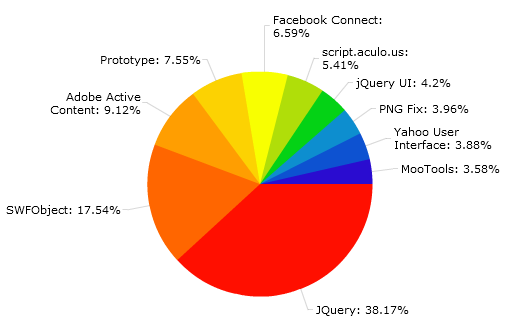
* **มีประสบการณ์ในการพัฒนาเว็บไซต์**
* **เขียนภาษา HTML ได้**
* **มีความเข้าใจ Cascading Style Sheets(css) เบื้องต้น**
* **อาจจะสามารถพัฒนา Web Application ด้วยภาษา PHP**

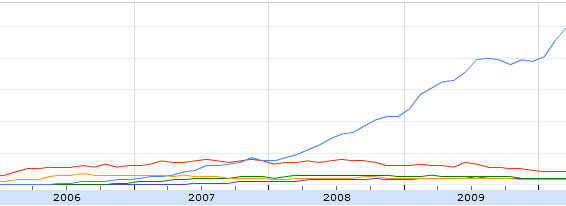
**เนื้อหาการอบรม**

* ทำความเข้าใจเกี่ยวกับ JavaScript Library
* รู้จักกับความสามารถของ jQuery
* รู้จักกับ XHTML และ DOM (Document Object Model)
* รู้จักกับ CSS
* การมองโค้ด XHTML ในรูปแบบของ DOM
* เรียนรู้พื้นฐาน JavaScript
* การใช้งาน jQuery เบื้องต้น
* ติดตั้ง firebug เพื่อเอาไว้ตรวจสอบ Error ของ jQuery
* โครงสร้างของ jQuery
* ความแตกต่างระหว่าง OnLoad Event กับ ready method
* ทำความเข้าใจเกี่ยวกับ Selector ในแบบต่าง ๆ
* ข้อพิจารณาการใช้งาน Selector
* Event ของ jQuery
* การเขียน Function กับ jQuery
* การเข้าถึงและจัดการข้อมูลในรูปแบบของ DOM ด้วย jQuery
* jQuery กับการจัดการ CSS
* เอฟเฟ็กต์(Effects)การสร้างแอนิเมชั่นด้วย jQuery
* Ajax การรับส่งค่าทั้งในแบบ GET and POST
* jQuery with XML
* jQuery with json
* การพัฒนา Web Application ในรูปแบบ AJAX ด้วย jQuery
* jQuery , PHP , Mysql , Ajax (Select, Update, Insert, Delete)
* พัฒนา Plugin ของ jQuery ไว้ใช้งาน
* การใช้งาน Plugin
* การใช้งาน jQuery Ui
* การอบรมเป็นแบบ WorkShop คือ ผู้สอนและผู้เรียนเน้นการปฏิบัติเป็นสำคัญ !

1. การเขียนการตรวจสอบข้อมูลของ Input Form
2. การสร้าง Gallery รูปภาพ
3. การสร้าง Search Auto complete (jquery UI)
4. การสร้าง Slide News (jquery UI)
5. การแบ่งหน้าแสดงผล (jquery UI)
6. การเรียงลำดับข้อมูลในตาราง (jquery UI)
7. การสร้างตะกร้าสินค้า

**ทำความเข้าใจเกี่ยวกับ JavaScript Library**

****

****

Javascript library คือ การนำภาษา javascript มาเขียนเป็นฟังก์ชั่นใหม่ให้ใช้งานง่ายขึ้นมีประสิทธิภาพมากขึ้นเรียกว่า API(Application Programming Interface)เป็นฟังก์ชั่นที่เอาไว้ให้ผู้พัฒนาสามารถเรียกใช้ได้เลยโดยไม่ต้องเขียนเองทั้งหมด็้นgdd JavaScript libraryมีหลากหลายค่ายมากมายยกตัวอย่างเช่น [Prototype](http://en.wikipedia.org/wiki/Prototype_JavaScript_Framework), [MooTools](http://en.wikipedia.org/wiki/MooTools), [jQuery](http://en.wikipedia.org/wiki/JQuery), [Dojo](http://en.wikipedia.org/wiki/Dojo_Toolkit), [YUI](http://en.wikipedia.org/wiki/YUI_Library) โดยแต่ละค่ายก็จะมีข้อดีขอ้เสียต่างกัน javascript library ที่มาแรงและมีคนใช้กันมากทีสุดในขณะนี้คือ javascript library jQuery ซึ่งมีแนวโน้มการใช้งานสูงขึ้นเรื่อยๆ เมื่อเทียบกับ javascript library อื่นๆ

jQuery JavaScript library ก็คือการนำภาษา JavaScript มาสร้างเป็นฟังก์ชั่นใหม่ในรูปแบบของ framework เพื่อให้ง่ายต่อการใช้งาน และไม่ต้องมาเขียนใหม่เองทั้งหมดตั้งแต่ต้น อย่างเช่นการเขียนโปรแกรมในรูปแบบ ajax ก็สามารถทำได้แบบง่ายๆเพียง code ไม่กี่บรรทัด หรือจะเขียนโปรแกรมดัก Event (เหตุการณ์) ต่างๆ เช่น การ click, rollover, mouse ก็เขียนได้ง่ายขึ้นและลดปัญหาความไม่เข้ากันของเบราเซอร์ที่ต่างกัน

**รู้จักกับความสามารถของ jQuery**

การนำ jQuery มาประยุกต์ใช้งานนั้นมีหลากหลายรูปแบบโดยส่วนมากจะเน้นการใช้งาน ในส่วนของ client side User interface ที่ให้ user สามารถใช้งานเว็บไชต์ได้ง่ายยิ่งขึ้นและเร็วขึ้น และยังทำให้ผู้เขียนโปรแกรมก็สามารถเขียนโปรแกรมยากๆ ได้ง่ายขึ้นเพราะ jqueryได้รวบรวม library ที่สำคัญต่างๆที่มีความซับซ้อนรวมเข้าด้วยกันและจัดการผูกฟังก์ชั่นต่างๆ ของ javascript ให้สามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าเดิม ความสามารถที่สำคัญในไลบรารี jQuery มีดังต่อไปนี้

**1 เข้าถึงอิลิเมนต์ในเอกสาร(Selector)**

jQuery มีกลไกลในการทำงานของซีเลกเตอร์ที่มีประสิทธิภาพในการดึงข้อมูลจากเอกสาร โดยดึงเอาความสามารถในการเข้าถึงอิเมนต์ของCSSมาจัดการ จึงทำให้ง่ายในการจัดการกับอิลิเมนต์ต่างๆมากกว่าการใช้ javacript แบบเดิม

**2 การแสดงผลของเว็บ Cross Browser)**

jQuery สามารถแก้ปัญหาของการแสดงผลบนเบราเซอร์ที่ต่างกัน ให้แสดงผลที่ถูกต้อง(ความไม่เข้ากันของบราวเซอร์เป็นปัญหาใหญ่มากสำหรับการนักพัฒนาเว็บไชต์เพราะแต่ละบราวเซอร์ก็ใช้มาตรฐานของตัวเองในการพัฒนา)

**3 แก้ไขคอนเท็นต์ของเอกสาร(DOM)**

jQuery สามารถเปลี่ยนข้อความ,แทรกรูปภาพ ,เรียงลำดับลิสใหม่ หรือสามารถแก้โครงสร้าง htmlใหม่ทั้งหมด หรือต่อเติมจากโครงสร้างเดิมได้อย่างง่ายๆ

**4 โต้ตอบกับการทำงานของผู้ใช้(Event)**

jQuery มีวิธีการดักจับอีเวนต์ที่หลากหลายเช่น อีเวนต์คลิก,อีเวนต์ดับเบิลคลิก สามารถจัดการฟังก์ชั่นดักจับอีเวนต์ต่างๆอย่างเป็นระเบียนมากขึ้น

**5 แสดงภาพเคลื่อนไหวให้เอกสาร(Effect)**

jQuery มีไลบรารีในการจัดการเรื่องเอฟแฟ็กต์ไว้อย่างสมบรูณ์ เช่นการกำหนดอิลีเมนต์ให้ภาพจาง ๆแล้วค่อยๆชัดขึ้นเป็นต้น

**6 ดึงข้อมูลจากเชิร์ฟเวอร์โดยไม่ต้องรีเฟรช(AJax)**

รูป แบบการเขียนโค้ดแบบนี้เป็นที่รู้จักกันดีในชื่อเรียกว่า Asynchronous JavaScript And XML(AJAX) ซึ่งช่วยให้นักพัฒนาเว็บไชต์สร้างเว็บไชต์ที่เต็มไปด้วยคุณลักษณะที่โดดเด่น ในการโต้ตอบกับผู้ใช้ได้อย่างรวดเร็ว มีประสิทธาภาพ เขียนง่าย ไม่มีการ Refresh โดยไลบรารีได้ขจัดความยุ่งยากในกระบวนการทำงานนี้และความแตกต่างของแต่ละเบอร์เซอร์ให้หมดไป

**รู้จักกับ HTML และ DOM (Document Object Model)**

**HTML**

## HTML คือภาษาสำหรับออกแบบหน้าเว็บไซต์

+ HTML ย่อมาจาก Hyper Text Markup Language  
+ HTML ไม่ใช่ภาษาที่ใช้ในการเขียนโปรแกรม แต่เป็นภาษาที่ใช้ตบแต่ง (Markup Language)  
+ Markup Language คือชุดของแท็ก (Tags) เช่น <html> , <p> เป็นต้น

**คำสั่งเริ่มต้น**  
**รูปแบบ** **<HTML>.....</HTML>**  
คำสั่ง <HTML> เป็นคำสั่งเริ่มต้นในการเขียนโปรแกรม และ </HTML>เป็นคำสั่งจุดสิ้นสุดโปรแกรมเหมือนคำสั่ง Beign และ End ใน Pascal   
  
**คำสั่งการทำหมายเหตุ (Comment)**  
**รูปแบบ** **<!-- ..... -->**  
**ตัวอย่าง** **<!-- END WEBSTAT CODE -->**  
ข้อความที่อยู่ในคำสั่งจะปรากฏอยู่ในโปรแกรมแต่ไม่ถูกแสดง บนจอภาพ  
  
**ส่วนหัว**   
**รูปแบบ** **<HEAD>.....</HEAD>**  
ใช้กำหนดข้อความ ในส่วนที่เป็น ชื่อเรื่อง ภายในคำสั่งนี้ จะมีคำสั่งย่อย อีกหนึ่งคำสั่ง คือ <TITLE>   
  
**กำหนดข้อความในไตเติลบาร์   
รูปแบบ <TITLE>.....</TITLE>**  
**ตัวอย่าง <TITLE> บทเรียน HTML </TITLE>**   
เป็นส่วนแสดงชื่อของเอกสาร จะปรากฏ ขณะที่ไฟล์ HTML ทำงานอยู่ ข้อความ ที่กำหนด ในส่วนนี้ จะไม่ถูกนำไปแสดง ผลของ เว็บเบราเซอร์แต่จะปรากฏในส่วนของไตเติลบาร์ (Title bar) ที่เป็นชื่อของวินโดว์ข้างบนไม่ควรให้ยาวเกินไป เพียงให้รู้ว่าเว็บเพจที่กำลังใช้งานอยู่เกี่ยวข้องกับอะไร **ส่วนของเนื้อหา  
รูปแบบ <BODY>.....</BODY>**ส่วนเนื้อหาของโปรแกรมจะเริ่มต้นด้วย คำสั่ง <BODY> และจบลงด้วย </BODY> ภายในคำสั่งนี้ คือ ส่วนที่จะ แสดงทางจอภาพ

**DOM**



DOM ก็คือการประกาศ object เพื่อให้สามารถเข้าถึงได้ (เช่นเราเขียน JavaScript เพื่อเรียก object ต่างๆที่อยู่หน้าเว็บ)จากหน้าเว็บนี้ ถ้าเราเอามาเขียนผังโครงสร้างแบบ tree เพื่อใช้ในการอธิบาย DOM เราก็จะได้ออกมาหน้าตาดังรูปด้านบนจากรูปเราจะเห็นชัดเลย ว่าใครอยู่เหนือใคร  ใครเป็น parent ใครเป็น child และจากพื้นฐานของ DOM ที่ทำให้เราแยกแยะ object ออกมาได้แล้วนั้น จะทำให้เวลาที่เราจะ access เข้าถึงแต่ละ object เราก็จะทำได้อย่างง่ายขึ้นนั่นเอง

เช่น ซ้ายมือจะเห็นว่า title เป็น child ของ head และกลับกัน head เป็น parent ของ childโดย DOM นี้จะถูกประกาศเมื่อ หน้าเว็บโหลดเสร็จแล้วเท่านั้นครับ และ element ใดที่ไม่ปรากฏเมื่อตอนที่หน้าเว็บโหลดเสร็จแล้ว ก็จะไม่มีใน DOM ด้วยเช่นกัน (ส่วนที่สร้างโดยใช้ JavaScript)

**jQuery**มีความสามารถในการเข้าถึง element ใน DOM ซึ่งเป็นความสามารถที่มีประสิทธิภาพอย่างหนึ่งของ jqueryโดยนำความสามารถของซีเลกเตอร์ Cascading Style Sheets(CSS) เพื่อช่วยให้เข้าถึง element ของ DOM

**รู้จักกับ CSS**

****

**CSS (Cascading Style Sheets)** คือ ชุดคำสั่งที่ใช้สำหรับการกำหนดการแสดงผลข้อมูลหน้าเว็บพจ เป็นมาตราฐานหนึ่งของ W3C ที่กำหนดขึ้นมา เพื่อใช้ในการจัดการรูปแบบเอกสารเว็บเพจโดยเฉพาะ การใช้งาน CSS จะเข้ามาช่วยเพิ่มความสามารถให้กับ HTML เดิมที่เราใช้งานกันอยู่ในปัจจุบัน

**ประโยชน์ของ CSS**  
-  ลดขนาดแฟ้มเอกสารเว็บไซต์ ทำให้โหลดเว็บไซต์ได้ไว  
-  ลดเวลาในการออกแบบพัฒนาเว็บ เพราะสามารถนำ CSS ไปใช้กับเอกสารเว็บอื่นๆ ได้ง่าย สะดวก  
-  ลดเวลาในการปรับแก้ไขเอกสารเว็บ  
-  ทำงานได้ดีกับเทคโนโลยีใหม่ๆ เช่น การแสดงผลบนเบราว์เซอร์โมบายต่างๆ

**External CSS**

**เป็นการเรียกใช้ไฟล์ CSS จากภายนอกเข้ามาใช้งานภายในเพจที่ต้องการ**

|  |
| --- |
| **<link href="file\_css.css" rel="stylesheet" type="text/css">** |

โดยระบุไว้ใน head element

**Internal CSS**

การเรียกใช้งานไฟล์ CSS ภายในเพจเดียวกันสามารถเรียกใช้ได้ดังนี้

|  |
| --- |
| **<style type=”text/css”>**  **/\*Code define CSS\*/**  **</style>** |

โดยระบุไว้ใน head element

**รูปแบบคำสั่งของ CSS**

|  |
| --- |
| **Selector { property : value ; property : value ; }** |

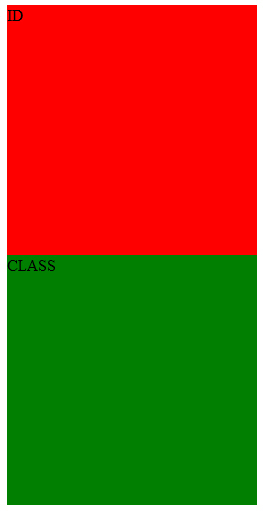
รูปแบบการใส่ Comment

|  |
| --- |
| **/\* Comment \*/** |

**ตัวอย่างการเรียกใช้งาน CSS จากภายใน**

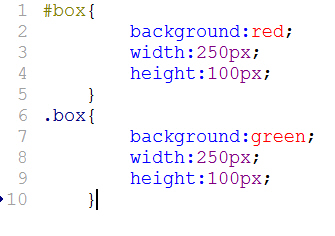


**ผลลัพธ์**

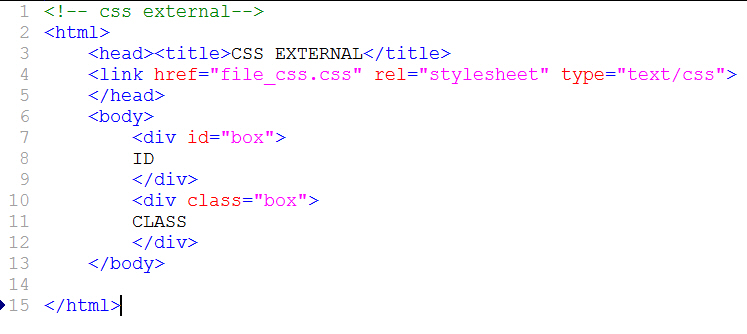


**ตัวอย่างการเรียกใช้งาน CSS จากภายนอก**

ไฟล์ file\_css.css

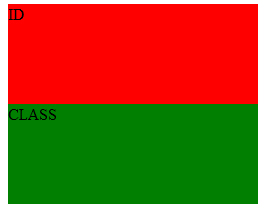


ไฟล์ css\_external.html



rel = เป็นตัวระบุความสำพันธ์ระหว่างหน้าเพจปัจจุบันกับหน้าเพจที่ลิงค์มาจากข้างนอก

**ผลลัพธ์**



**เรียนรู้พื้นฐาน JavaScript**

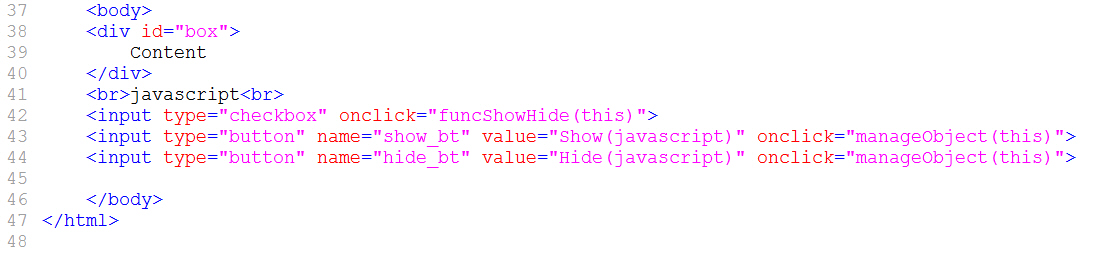
**Java Script คืออะไร?**

JavaScript เป็นภาษาโปรแกรม (programming language) ประเภทหนึ่ง ที่เรียกกันว่า "สคริปต์" (script) ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) ภาษานี้เดิมมีชื่อว่า LiveScript ได้รับการพัฒนาขึ้นโดย Netscape ด้วยวัตถุประสงค์ เพื่อที่จะช่วยให้เว็บเพจสามารถแสดงเนื้อหา ที่มีการเปลี่ยนแปลงไปได้ ตามเงื่อนไขหรือสภาพแวดล้อมต่างๆกัน หรือสามารถโต้ตอบกับผู้ชมได้มากขึ้น ทั้งนี้เพราะภาษา HTML แต่เดิมนั้น เหมาะสำหรับใช้แสดงเอกสาร ที่มีเนื้อหาคงที่แน่นอน และไม่มีลูกเล่นอะไรมากมายนัก  
 เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA ซึ่งเราจะพบว่าปัจจุบัน บราวเซอร์ส่วนใหญ่จะรองรับ JavaScript กันหมดแล้ว  
 การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ดี สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชั่นใหม่ๆออกมาด้วย (ปัจจุบันคือรุ่น 1.5) ดังนั้น ถ้านำโค้ดของเวอร์ชั่นใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด error ได้

การทำงานของ JavaScript เกิดขึ้นบนบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server-side script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ดี จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่ (ความจริง JavaScript ที่ทำงานบนเซิร์ฟเวอร์เวอร์ก็มี ซึ่งต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก)  
  
 การทำงานของ JavaScript จะมีประสิทธิภาพมาก ถ้ามันสามารถดัดแปลงคุณสมบัติ ขององค์ประกอบต่างๆ บนเว็บเพจ (เช่น สี หรือรูปแบบของข้อความ) และสามารถรับรู้เหตุการณ์ ที่ผู้ชมเว็บเพจโต้ตอบกับองค์ประกอบเหล่านั้น (เช่น การคลิก หรือเลื่อนเมาส์ไปวาง) ได้ ดังนั้นจากภาษา HTML เดิม ที่มีลักษณะเนื้อหาแบบคงที่(static) ใน HTML เวอร์ชั่นใหม่ๆ จึงได้มีการพัฒนาให้มีคุณสมบัติบางอย่างเพิ่มขึ้น และมีลักษณะเป็นอ็อบเจ็ค "object" มากขึ้น การทำงานร่วมกันระหว่างคุณสมบัติใหม่ของ HTML ร่วมกับ JavaScript นี้เอง ทำให้เกิดเป็นสิ่งที่เรียกว่า Dynamic HTML คือภาษา HTML ที่สามารถใช้สร้างเว็บเพจที่มีลักษณะพลวัต (dynamic) ได้นั่นเอง  
  
 นอกจากนี้ อีกองค์ประกอบหนึ่งที่เกี่ยวข้อง ก็คือ Cascading Style Sheet (CSS) ซึ่งเป็นภาษาที่ช่วยให้เราควบคุมรูปแบบ ขององค์ประกอบต่างๆ บนเว็บเพจ ได้อย่างมีประสิทธิภาพมากกว่าคำสั่ง หรือแท็ก -(tag) ปกติของ HTML เนื่องจาก JavaScript สามารถดัดแปลงคุณสมบัติของ CSS ได้เช่นกัน ดังนั้นมันจึงช่วยให้เราควบคุมเว็บเพจ ได้อย่างมีประสิทธิภาพมากมากยิ่งขึ้นไปอีก

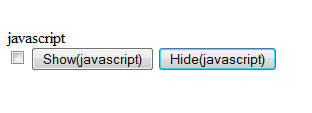
**ตัวอยางการเขียนโปรแกรม Hide Show Object ด้วย javasript**

****

****

**ผลลัพธ์**

เมื่อกดปุ่ม Hide(javascript)ก็จะเป็นการซ่อนกล่องสีแดงที่เราได้เขียนโปรแกรมไว้



เมื่อกดปุ่ม Show(javascript)ก็จะเป็นการแสดงกล่องสีแดง



**การใช้งาน jQuery เบื้องต้น**

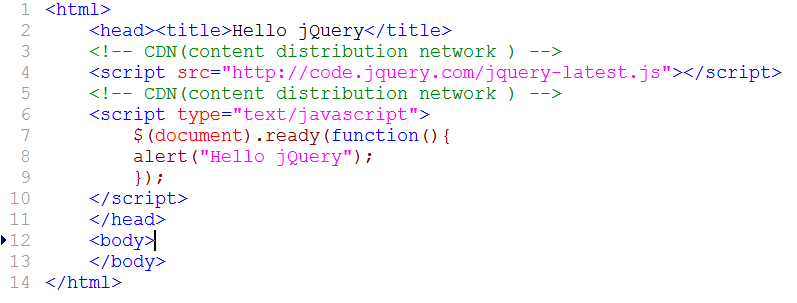
สำหรับเว็บไซต์ในการดาวน์โลด ไฟล์ jquery.js ได้ที่ [http://jquery.com](http://jquery.com/)

การเรียกใช้งาน jquery.js ก็เหมือนเป็นการเรียกใช้งาน ไฟล์ JavaScript ทั่วไปถ้าเครื่องคอมพิวเตอร์ต่ออินเตอร์เน็ตอยู่สามารถเรียกใช้ไฟล์ jquery.js ได้โดยเรียก **Content Delivery Network** or **Content Distribution Network** (**CDN**) ที่ให้บริการ JavaScript library jQuery ต่างๆออนไลน์เช่น เว็บของjQuery(<http://code.jquery.com/jquery-1.7.1.min.js>) หรือgoogle(<http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js>) การเรียกใช้แบบออนไลน์จะมีข้อดีตรงที่ความเร็วในการใช้งานมากขึ้นลดแบนวิดในการดึงข้อมูลพร้อมกัน เพราะserverจะมีหลายเครื่องให้บริการและมีการupdateอยู่ในเวอร์ชั่นที่ใหม่ตลอด หรือจะใช้งานออฟไลน์ก็สามารถโหลดมาเก็บไว้ที่เครื่องแล้วก็เรียกใช้งานตามพาททีเก็บไฟล์ JavaScript library jQuery

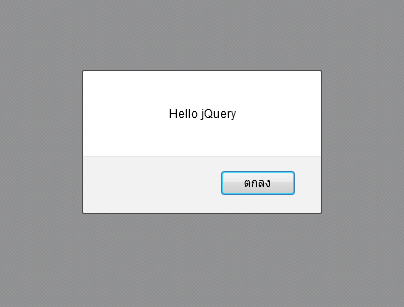
ในการใช้งาน jquery เราไม่จำเป็นต้องติดตั้งโปรแกรมอะไรเลย แค่นำไฟล์ jquery ไปวางไว้ในพื้นที่ ที่สามารถอ้างอิงได้เนื่องจาก JavaScript เป็นภาษาแบบ interpreted(เป็นภาษาที่ใช้ตัวแปรภาษาอ่านคำสั่งทีละคำสั่ง)ไม่มีขั้นตอนการคอมไพล์ หรือขั้นตอนการ build

**ทดสอบเขียนโปรแกรมเพื่อตรวจสอบว่าสามารถใช้งาน jQuery ได้หรือไม่**

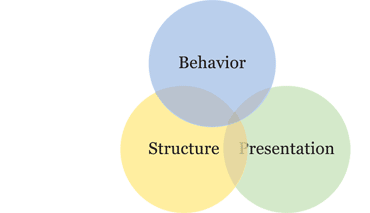
**HELLO JQUERY**

****

**ผลลัพธ์**

****

**โครงสร้างของ jQuery**



**อธิบาย ฟังก์ชั่น $()**

ฟังก์ชั่น $() เป็นฟังก์ชั่นเริ่มต้นของ jquery ต้องประกาศฟังก์ชั่น $() ทุกครั้งที่เรียกใช้งาน jquery  $()เป็นฟังก์ชั่นในการสร้าง object jquery โดยปกติจะเรียกใช้งานฟังก์ชั่นนี้ได้ 2 รูปแบบ คือ $() หรือ $(document) เพราะภายใน object jquery จะห่อหุ้มด้วย อิลิเมนต์ Dom ของเอกสารนั้นๆเพื่อที่จะจัดการกับอิลิเมนต์ได้

เมื่อเราประการฟังก์ชั่น $(document) ในหน้าเอกสารhtml ก็จะทำการห่อหุ้มอิลิเมนต์ Dom ต่างๆทั่งหมดเช่น

ตัวอย่างโค้ด

<div>,<h1>,<body>,<input>,<p>

เพื่อที่จะใช้ในการจัดการกับ element เหล่านี้ได้ง่ายขึ้นเช่นการเพิ่ม css ให้กับ element การใส่ effect การเพิ่ม element การลบ element รวมถึงการอ่านค่าต่างของอิลิเมนต์ได้ง่ายๆ โดยไม่ต้องใช้ ฟังก์ชั่น วนลูปในการเขียนโค้ดยาวๆ เพราะ jqueryมีคุณสมบัติ implicit iteration มีฟังก์การวนลูปในตัว jquery ไม่ต้องเขียนโค้ดเอง

**การประการฟังก์ชั่นแบบไม่ระบุชื่อ** **ปกติจะเรียกฟังก์ดังนี้**

|  |  |
| --- | --- |
|  | function runhighlight(){              $("#content").addClass("highlight");    }    $(document).ready(runhighlight); |

Jquery สามารถเขียนโด้ดให้สั้นลงได้โดยใช้ ฟังก์ชั่น **แบบไม่ระบุชื่อ(anonymous function หรือ lamba function)**

**สามารถเขียนโค้ดให้สั้นลงได้ดังนี้**

|  |  |
| --- | --- |
|  | $(document).ready(function(){    $(‘#content’).addClass(‘highlight’);  }); |
|  |  |

เราจะใช้ฟังก์ชั่นนี้ในกรณีที่ เป็นฟังก์ชั่นที่ไม่ค่อยได้นำกลับมาใช้ใหม่

**ความแตกต่างระหว่าง OnLoad Event กับ ready method**

เมื่อต้องการให้ ฟังก์ชั่น ทำงานทันทีที่โหลดเพจเสร็จสามารถทำได้ 2วิธีคือ

**วิธีที่ 1** รูปแบบเก่าจะใส่ ฟังก์ชั่น onload ใน tag body เช่น

<body onload=” runhighlight();”>

และมีฟังก์ชั่น runhighlight ประกาศฟังก์ชั่น ไว้ที่ tag head เช่น

****

**แต่มีข้อเสียคือ**

เราต้องไปแก้ไขเอกสาร html เพื่อให้ฟังก์ชั่นทำงาน ทำให้โค้ดผูกติดกับเอกสาร html โค้ดไม่เป็นระเบียบ อาจเกิดการทำงานซ้ำๆ ในเพจต่างๆหรือถูกเรียกในอีเวนต์(event handlers)ต่างๆ เช่น การคลิกซึ่งเกิดขึ้นได้ตลอดทุกๆ อิลิเมนต์ของเพจและเพิ่มการทำงานใหม่ๆ เข้าไป จำเป็นต้องแก้โค้ดหลายๆที่ จึงเป็นการเพิ่มโอกาสความผิดผลาดสูงในการทำงาน และ**ไม่สามารถเรียกได้ 2 ฟังก์ชั่นให้ทำงานพร้อมกันได้ในขณะที่โหลดDOM เสร็จ**

**วิธีที่ 2**



วิธีนี้จะไม่ต้องจัดการแก้ไข html โดยตรง แต่จะเป็นการจัดการเขียนโค้ดในส่วนของ JavaScript ทั้งหมด จึงหมดปัญหาจากข้อเสียที่ได้กล่าวไว้ข้างต้น

**ทำความเข้าใจเกี่ยวกับ Selector ในแบบต่าง ๆ**

Selector คือการเลือกอิลิเมนต์ที่ต้องการจาก DOM (Document Object Model)เพื่อนำอิลิเมนต์มากำทำการผูกฟังก์ชั่นต่างๆหรือ กำหนด event ส่วนประกอบพื้นฐานของ selector มี 3 อย่างคือ

|  |  |  |  |
| --- | --- | --- | --- |
| **Selector** | **Css** | **jQuery** | **คำอธิบาย** |
| ชื่อแท็ก | p | $(“p”) | เลือกทุกอิลิเมนต์ที่อยู่ภายใต้แท็ก<p></p> |
| ID | #idName | $(“#idName”) | เลือกหนึ่งอิลิเมนต์ที่มี ID เท่ากับ idName |
| Class | .className | $(“.ClassName”) | เลือกทุกอิลิเมนต์ที่มี Class เท่ากับ ClassName |

**ตัวอย่างการใช้งาน Tag selector**

<h2>Ttitle</h2>

<strong>Ttitle<strong>

<p>Paragraph Content1</p>

<p>Paragraph Content1</p>

<p>Paragraph Content1</p>

$(document).ready(function(){

$(“p”).css(“font-size”,”15px”);

});

กำหนดให้ข้อมูลที่อยู่ภายใต้ Paragraph<p> มีขนาดตัวอักษรเท่ากับ 15px โดยใช้ Method การจัดการ CSS นั้นก็คือ .css()

**ตัวอย่างการใช้งาน ID selector**

<div id=”idName”>

Content…

</div>

$(document).ready(function(){

$(“#idName”).css(“font-size”,”20px”);

});

**ตัวอย่างการใช้งาน Class selector**

<ul>

<li class=”className”>menu1</li>

<li class=”className”>menu2</li>

<li class=”className”>menu3</li>

<li class=”className”>menu4</li>

</ul>

$(document).ready(function(){

$(“. className”).css({“font-size”:”20px”,”float”:”left”});

});

นอกจาก selector พื้นฐานแล้วก็ยังมีselector อื่นๆดังนี้

**Attribute Selector** เป็น selector ย่อยของ CSS ช่วยเลือก element ที่ต้องการโดยใช้ค่า properties ของ HTML เช่น Attribute title(<a href=”link.html” title=”value”></a>)

**ตัวอย่างการใช้งาน Attribute selector**

<style type=”text/css”>

.httpLink{ Background-color:#cccccc;}

.mailLink{Background-color:#ffccffcc;}

.pdfLink{ Background-color:red;}

</style>

<ul class=”menu”>

<li><a href=”http://www.workphp.com”>Link to Workphp</a></li>

<li>><a href=”file.pdf” title=”ducument”>Download PDF </a></li>

<li>><a href=”mailto:nn.it@hotmail.com” >Contact Us</a></li>

<li><a href=”map.html” title=”map”>Map</a></li>

</ul>

$(document).ready(function(){

$(“a[href^=http]”).addClass(“httpLink”);

$(“a[href$=.pdf]”).addClass(“pdfLink”);

$(“a[href^=mailto]”).addClass(“mailLink”);

$(“[title= map]”).css(“background-color”,”bule”);

});

**คัสตอมซีเลกเตอร์**

Custom Selector เป็นselectorที่jQuery สร้างขึ้นมาเพื่อช่วยให้เลือก element ได้แม่นยำขึ้นโดยสามารถระบุตำแหน่ง element ตามต้องการได้เลย เช่น

**ตัวอย่างการใช้งาน Custom selector**

<ul id=”menu”>

<li>HOME</li>

**<li>ABOUT US</li>**

<li>CONTACT US</li>

</ul>

$(ducument).read(function(){

$(“ul#menu li:eq(**1**)”).css(“font-weight”,”bold”);

});

การนับข้อมูลใน array ของ javascript เป็นการนับในรูปแบบ zero-based คือเริ่มต้นที่ 0

**ชีเล็กเตอร์สำหรับฟอร์ม**

Selector สำหรับ Form เป็น selector ในการเลือก element ของ Form โดยเฉพาะ เพื่อความสะดวกในการเข้าถึงelement มากขึ้น โดยมี selector ที่ใช้กับ Formมีดังนี้

|  |  |
| --- | --- |
| Selector | Elementที่ถูกเลือก |
| :text, :checkbox, :radio, :image, :submit, :reset , :password, :file | เลือกอิลิเมนต์ที่มีแอตทริบิวต์ type เท่ากับชื่อซีเลกเตอร์ที่กำหนด เช่น :text เป็นการเลือกอิลิเมนต์<input type=”text”> |
| :input | เลือกอิลิเมนต์ทั้งหมดที่เป็นอิลิเมนต์<input>,<select><textarea><button>  (input form ทั้งหมด) |
| :button | เลือกอิลิเมนต์ <input> ที่มี type=”button” และอิลิเมนต์ <button> |
| :enabled | เลือกอิลิเมนต์ของฟอร์มที่สามารถป้อนข้อมูลได้ |
| :disable | เลือกอิลิเมนต์ของฟอร์มที่ไม่สามารถป้อนข้อมูลได้ |
| :checked | เลือกอิลิเมนต์เช็คบ็อกซ์ที่ถูกเลือกหรือ อิลิเมนต์เรดิโอทีถูกเลือก |
| :selected | เลือกอิลิเมนต์<option> ทั้งหมด |

ตัวอย่างการใช้งาน ชีเล็กเตอร์สำหรับฟอร์ม

<form>

<input type=”text” / >

<input type=”radio” />

<input type=”checkbox” />

<textarea></textarea>

<select >

<option>1</option>

<option>2</option>

</select>

<input type=”submit” value=”submit”>

</form>

$(ducment).ready(function(){

var allValue = $(“:input”). serialize ();

//หากไม่ใช้ซีเลกเตอร์ Form สามารถใช้ซีเลกเตอร์แบบปกติก็คือใช้ attribute selector

var allValue = $(“[type=input]”).serialize();

});

**การทำ Chaining(การเชื่อมเมธอด)**

Chaining คือการเชื่อมเมธอดต่อกันโดยไม่ต้องระบุselectorเพี่อสร้างออบเจ็กต์ใหม่ทุกครั้งที่เรียกใช้เมธอดเป็นการเพิ่มประสิทธิภาพในการทำงานของสคริปต์

**ตัวอย่าง**

<ul>

<li>row1</li>

<li>row2</li>

<li>row3</li>

</ul>

$(ducument).ready(function(){

$(“ul li:eq(1)”).css(“font-weight”,”bold”).end().find(“ul li:eq(0)”).css(“font-size”,”10px”);

});

**อีเวนต์(Events)**

**jQuery Events รูปแบบคำสั่งการใช้ jQuery กับ Events บน jQuery Library Framework**  
**jQuery กับ Events** ในภาษาของ **JavaScript** การใช้งาน **Events** นั้นมีความจำเป็นอย่างยิ่ง แทบจะเรียกได้ว่าเป็นหัวใจของการพัฒนาโปรแกรมของฝั่ง **Client** ได้เลยทีเดียว เพราะเหตุการณ์ต่าง ๆ เหล่านี้ที่เกิดขึ้นจากฝั่งของ **Client** จะเป็นตัวควบคุมและบ่งบอกวัตถุประสงค์ของผู้ใช้ ว่าต้องการทำอะไร ไปที่ไหน และอย่างไร โดย **Events** ต่าง ๆ ที User ได้กระทำนั้น เราสามารถทำการดักตรวจสอบและควบคุม เพื่อให้โปรแกรมนั้นไปในทิศทางที่ได้ออกแบบไว้ และสามารถทำงานได้อย่างถูกต้อง และการนำ jQuery มาใช้เกี่ยวกับการควบคุม **Events** ก็เป็นทางเลือกที่น่าสนใจอย่างยิ่ง เพราะ **jQuery** สามารถดักจับ **Events** ต่าง ๆ ที่เกิดขึ้นในฝั่ง Client ได้อย่างง่ายดายด้วย Code อันสั้น ๆ และสามารถใช้งานได้สะดวกกว่าในการเขียน **JavaScript** ด้วยระบบเดิม ๆ  
  
**รูปแบบคำสั่งของ Events**  
  
**jQuery Events Syntax**

|  |
| --- |
| $("selector").event(); |

**รูปแบบคำสั่งของ Events ของ JavaScript ระดบบเดิมกับ Events ของ jQuery**  
  
**Events ของ JavaScript ระบบเดิม**

|  |
| --- |
| <input type="button" id="btn1" OnClick="alert('Sawatdee');"> |

**Events ของ jQuery**

|  |
| --- |
| $("#btn1").click(alert('Sawatdee')); |

ในการกำหนด **Events ของ jQuery** สามารถเรียกใช้งาน method เพื่อกำหนดเหตุการณ์ให้กับ **Events** นั้น ๆ ได้ในทันที หรือจะเรียก **Callback ของ function** เพื่อให้ทำงานต่าง ๆ ตามที่เราต้องการ โดยสามารถสร้างคำสั่งการทำงานลงใน **Callback ของ function** เพิ่มเติมได้  
  
**Events ที่ไมีมี Callback function**

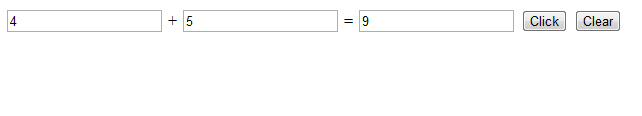
|  |
| --- |
| $("#btn1").click();  $("#div1").hide(); |

**Events ที่มี Callback function**

|  |
| --- |
| $("#btn1").click(function(){  $("#txt3").val(parseFloat($("#txt1").val())+parseFloat($("#txt2").val()));  }); |

**ตัวอย่างการใช้งาน Events ของ jQuery**  
  
**Code (jQuery)**

|  |
| --- |
| <html>  <head>  <title>Event jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  var $value = parseFloat($("#txt1").val())+parseFloat($("#txt2").val());  $("#txt3").val($value).show();  });  $("#btn2").click(function(){  $("#txt1").val("");  $("#txt2").val("");  $("#txt3").val("").hide();  });  });  </script>  <input type="text" id="txt1" />+<input type="text" id="txt2" />=<input type="text" id="txt3" />  <input type="button" id="btn1" value="Submit"/>  <input type="button" id="btn2" value="Clear"/>  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
ตัวอย่างการใช้งาน **Events** ของ **jQuery** แบบไม่มี Callback $("#txt3").hide(); และที่แบบมี Callback $("#btn1").click(function(){ และ $("#btn2").click(function(){  
  
และจากตัวอย่างจะเห็นว่าต่อให้เราเปลี่ยนไปใช้ **jQuery Library Framework** แต่พื้นฐานของคำสั่งนั้นยังเป็นภาษา JavaScript ซึ่งบางคำสั่งของ **JavaScript** สามารถเรียกใช้งานได้ทันที เช่น ฟังก์ชั่นเกี่ยวกับการแปลงค่าหรือตรวจสอบชนิดของ Data Type  
  
**คำสั่ง Event อื่น ๆ ที่น่าสนใจ**  
**$(document).ready(function)** ความหมายคือ มีมีการโหลดหน้าเว็บเพจเสร็จสิ้น ถ้ามีการใส่คำสั่ง JavaScript ที่ไม่ผ่าน Events จะทำงานทันทีหลังจากโหลดหน้าเว็บเพจเรียบร้อย เช่น $("#txt3").hide();  
**$(selector).click(function)** ความหมายคือ เมื่อมีการคลิกยัง Selectors ที่กำหนดขึ้น  
**$(selector).dblclick(function)** ความหมายคือ เมื่อมีการดับเบิ้ลคลิกยัง Selectors ที่กำหนดขึ้น  
**$(selector).focus(function)** ความหมายคือ เมื่อมีการคลิก focus() ยัง Selectors ที่กำหนดขึ้น เช่น textbox  
**$(selector).mouseover(function)** ความหมายคือ เมื่อมีเมาส์เลื่อนผ่าน Selectors ที่กำหนดขึ้น  
**$(selector).error(function)** ความหมายคือ เมื่อมีการ error จาก Selectors ที่กำหนดขึ้น เช่น Tags img โหลดภาพไม่สมบูรณ์  
**$(selector).focusout (function)** ความหมายคือ focusout ความหมายคือ เมื่อมีกาย้ายตำแหน่ง focus() ออกจาก Selectors ที่กำหนดขึ้น เช่น เลื่อน focus() ออกจาก textbox  
**$(selector).keydown(function)** ความหมายคือ เมื่อมีการกดปุ่มลงในทันทีไปยัง Selectors ที่กำหนดขึ้น  
**$(selector).keypress(function)** ความหมายคือ เมื่อมีการกดปุ่มลงจนสุด ไปยัง Selectors ที่กำหนดขึ้น  
**$(selector).keyup(function)** ความหมายคือ เมื่อมีการปล่อยปุ่ม ที่ได้กดลงใน Selectors ที่กำหนดขึ้น

**Effect**

**jQuery Effects รูปแบบคำสั่งการใช้ jQuery กับ Effects บน jQuery Library Framework**  
**jQuery กับ Effects** อีกหนึ่งในความสามารถของ **jQuery** ก็คือการกำหนด **Effects ให้กับ element** ที่สามารถกระทำและกำหนดได้ง่าย ๆ ด้วยคำสั่งอันสั้น ๆ เช่น $("#div1").hide(1000); จะเป็นการ hide ที่ element โดยจะค่อย ๆ เลื่อนจากหายไปใน 1 วินาที หรือ $("div1").toggle(); ใช้ซ่อนและแสดง element ต่าง ๆ สลับไปมา และก็ยงมี **Effect** อื่น ๆ อีกมากมายที่สามารถดัดแปลงการใช้งานจาก **jQuery Library**   
  
**รูปแบบคำสั่งของ Effects**  
  
**jQuery Effects**

|  |
| --- |
| $("selector").effect(); |

**1. ตัวอย่างการใช้งาน Effects ของ .toggle(); ใช้ซ่อนและแสดง element ต่าง ๆ สลับไปมา**  
  
**Code (jQuery)**

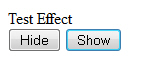
|  |
| --- |
| <html>  <head>  <title>Effect01 jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("#div1").toggle();  });  });  </script>  </head>  <body>  <div id="div1">Test Effect Toggle</div>  <input type="button" id="btn1" value="Click">  </body>  </html> |

**Screenshot**  
  
effect01.html.jpg  
  
**คำอธิบาย**  
เมื่อคลิกที่ปุ่ม ข้อความจะหายไป และ คลิกอีกครั้ง ข้อความจะแสดงกลับมาเหมือนเดิม ซึ่งสามารถทำได้อย่างง่าย ๆ จริง ๆ หรือจะลองกำหนด $("#div1").toggle(1000); หน่วยวินาทีเพื่อหน่วงเวลาก็ได้เช่นเดียวกัน โดย 1000 = 1 วินาที  
  
นอกจาก method ชื่อว่า .toggle() แล้ว method ของ hide() และ show() ก็สามารถสร้าง Effect ได้เช่นเดียวกัน โดยการกำหนดหน่วงเวลาเป็นหน่วยวินาที เช่น

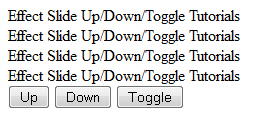
|  |
| --- |
| $("#div1").hide(1000);  $("#div1").show(1000); |

**2. ตัวอย่างการใช้งาน Effects ของ .hide(); และ .show(); ใช้ซ่อนและแสดง element ต่าง ๆ**

|  |
| --- |
| <html>  <head>  <title>jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("#div1").hide(1000);  });  $("#btn2").click(function(){  $("#div1").show(1000);  });  });  </script>  </head>  <body>  <div id="div1">Test Effect</div>  <input type="button" id="btn1" value="Hide">  <input type="button" id="btn2" value="Show">  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
เมื่อคลิกที่ปุ่ม Hide และ Show จะซ่อนและแสดงข้อความ โดยมีการหน่วงเวลาเพื่อสร้าง Effect ให้กับ element  
  
**3. ตัวอย่างการใช้งาน Effects ของ .slideUp(); , .slideDown(); และ .slideToggle(); ใช้ Slide element ต่าง ๆ**

|  |
| --- |
| <html>  <head>  <title>Effect Slide Down Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("#div1").slideUp();  });  $("#btn2").click(function(){  $("#div1").slideDown();  });  $("#btn3").click(function(){  $("#div1").slideToggle();  });  });  </script>  </head>  <body>  <div id="div1">Effect Slide Up/Down/Toggle Tutorials<br />Effect Slide Up/Down/Toggle Tutorials <br />Effect Slide Up/Down/Toggle Tutorials <br />Effect Slide Up/Down/Toggle Tutorials<br /></div>  <input type="button" id="btn1" value="Up">  <input type="button" id="btn2" value="Down">  <input type="button" id="btn3" value="Toggle">  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
เมื่อคลิกที่ปุ่ม Up จะทำการ Slide ข้อความเก็บไว้ด้านบน  
เมื่อคลิกที่ปุ่ม Down จะทำการ Slide ข้อความลงมาด้านล่างเหมือนเดิม  
เมื่อคลิกที่ปุ่ม Toggle จะทำการ Slide ข้อความขึ้นลงสลับไปมา  
  
**สรุป**  
**$("selector").show();** ความหมายคือ แสดง Selectors ของ element ที่กำลังอ้างถึง  
**$("selector").hide();** ความหมายคือ ซ่อน Selectors ของ element ที่กำลังอ้างถึง  
**$("selector").toggle();** ความหมายคือ ซ่อนและแสดง Selectors ของ element ที่กำลังอ้างถึง โดยทำงานสลับกันไปมา  
**$("selector").slideUp();** ความหมายคือ เลือน Selectors ของ element ที่กำลังอ้างถึง โดยเลื่อนขึ้นด้านบน  
**$("selector").slideDown();** ความหมายคือ เลือน Selectors ของ element ที่กำลังอ้างถึง โดยเลื่อนขึ้นลงล่างเหมือนเดิม  
**$("selector").slideToggle();** ความหมายคือ เลือน Selectors ของ element ที่กำลังอ้างถึง โดยทำงานสลับกันไปมา เลือนขึ้น-เลือนลง  
  
และจากตัวอย่างจะเห็นว่าต่อให้เราเปลี่ยนไปใช้ **jQuery Library Framework** แต่พื้นฐานของคำสั่งนั้นยังเป็นภาษา **JavaScript** ซึ่งบางคำสั่งของ **JavaScript** สามารถเรียกใช้งานได้ทันที เช่น ฟังก์ชั่นเกี่ยวกับการแปลงค่าหรือตรวจสอบชนิดของ Data Type  
  
**คำสั่ง Event อื่น ๆ ที่น่าสนใจ**  
**$(selector).fadeIn()** ความหมายคือ fade element ค่อย ๆ แสดงผลจางลง  
**$(selector).fadeOut()** ความหมายคือ fade element ค่อย ๆ แสดงผลชัดขึ้น  
**$(selector).fadeTo()** ความหมายคือ fade element การปรับภาพจากเข้มไปจาง  
**$(selector).animate()** ความหมายคือ การกำหนดเอนิเมชั่นตามความต้องการโดยกำหนด css  
**$(selector).fadeToggle()** ความหมายคือ สลับการะหว่าง faceIn และ fadeOut  
**$(selector).stop()** ความหมายคือ หยุดการทำงานที่กำลังทำงานอยู่ในขณะนั้น

**DOM Document Object Model**

**jQuery HTML รูปแบบคำสั่งการใช้ jQuery กับ HTML บน jQuery Library Framework**  
**jQuery กับ HTML** จากเรื่องบทก่อนหน้านี้ จะเห็นว่า **jQuery** สามารถใช้ **Selector** ในการอ้างถึง **element** ในหน้าเว็บเพจได้อย่างง่ายดาย และหลังจากที่สามารอ้างถึง **element** เหล่านั้น ได้แล้ว สิ่งที่จะทำต่อไปก็คือ การอ่านค่า หรือ กำหนดค่าให้กับ **element** นั้น ๆ ซึ่งในบทนี้ จะเป็นการกำหนดค่าให้กับ **HTML element** ที่อยู่ในรูปแบบ innetText หรือ innetHTML (ในรูปแบบ JavaScript แบบดั่งเดิม) ส่วนการ กำหนด **Events หรือ Attribute** อื่น ๆ จะได้พูดถึงในหัวข้อถัดไป  
  
**1. การใช้งาน method ของ $(selector).text(content) และ $(selector).html(content)**  
  
**innerHTML กับ jQuery**  
  
**Syntax ใน JavaScript ดังเดิม**

|  |
| --- |
| document.getElementById('div1').innerHTML = 'My Text'; |

**Syntax ใน jQuery จะใช้ method สั้น ๆ ว่า html()**

|  |
| --- |
| $("#div1").html("My Text"); |

**innerText กับ jQuery**  
  
**Syntax ใน JavaScript ดังเดิม**

|  |
| --- |
| document.getElementById('div1').innerText = '<b>My Text</b>'; |

**Syntax ใน jQuery จะใช้ method สั้น ๆ ว่า text()**

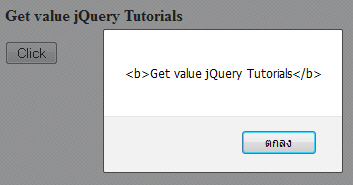
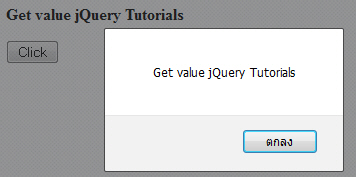
|  |
| --- |
| $("#div1").text("My Text"); |

**ตัวอย่างการใช้ method ของ text() และ html()**   
  
**Code (jQuery)**

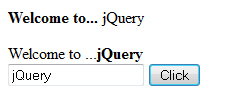
|  |
| --- |
| <html>  <head>  <title>Dom01 jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("p").html($("#txt1").val());  $("#div1").text($("#txt1").val());  });  });  </script>  </head>  <body>  <p></p>  <div id="div1"></div>  <input type="text" id="txt1">  <input type="button" id="btn1" value="Click">  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
เมื่อมีการรับค่า string ที่เป็น HTML Tags และเมื่อใช้ .text และ .html จะได้ผลลัพธ์ต่างกัน เหมือนในรูป  
  
  
**นอกจากนี้ในทางกลับกัน method ก็สามารถทำการอ่านค่าจาก .text() และ .html() ได้เช่นเดียวกัน**  
  
  
**Code (jQuery)**

|  |
| --- |
| <html>  <head>  <title>Get value jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  alert($("p").html());  alert($("p").text());  });  });  </script>  </head>  <body>  <p><b>Get value jQuery Tutorials</b></p>  <input type="button" id="btn1" value="Click">  </body>  </html> |

**Screenshot**  
  
  
  
ใช้ .html() จะได้ค่าเป็น HTML Tags  
  
  
  
ใช้ .text() จะได้ค่าเป็น Text เท่านั้น  
  
**สรุป**  
**$(selector).text(content)** ใช้ในการอ่านค่าและส่งค่าในรูปแบบของ Text  
**$(selector).html(content)** ใช้ในการอ่านค่าและส่งค่าในรูปแบบของ HTML Tags  
  
**คำอธิบาย**  
ใช้ method ของ .html() จะได้ค่าเป็น **HTML Tags** และ ใช้ .text() จะได้ค่าเป็น Text เท่านั้น  
  
  
  
**2. การใช้งาน method ของ $(selector).append(content)**  
สำหรับ **method append()** ใช้สำหรับการเพิ่มเนื้อหาเข้าไปกับ element ที่ได้เลือกไว้ เช่น **<p>Welcome to</p>** เมื่อใช้ **.append("jQuery")** จะได้ string ใหม่มีค่าเป็น **<p>Welcome to jQuery</p>**  
  
**ตัวอย่าง**  
  
**Code (jQuery)**

|  |
| --- |
| <html>  <head>  <title>Dom03 using append jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("p").append($("#txt1").val());  $("#div1").append("<b>"+$("#txt1").val()+"</b>");  });  });  </script>  </head>  <body>  <p><b>Welcome to... </b></p>  <div id="div1">Welcome to ...</div>  <input type="text" id="txt1">  <input type="button" id="btn1" value="Click">  </body>  </html> |

  
  
**คำอธิบาย**  
ทั้ง <p> และ <div id="div1"> ใช้ method append() ซึ่งได้ผลลัพธ์เหมือนกัน โดย append() สามารถรับค่าที่เป็น HTML Tags ได้เช่นเดียวกัน  
  
**ผลลัพธ์ทีได้ในมุมมองของ HTML Tags**

|  |
| --- |
| <p><b>Welcome to... jQuery</b></p> <div id="div1">Welcome to ... jQuery </div> |

**3. การใช้งาน method ของ $(selector).prepend(content)**  
สำหรับ **method prepend()** ใช้สำหรับการเพิ่มเนื้อหาเข้าไปกับ element ที่ได้เลือกไว้ โดยเพิ่มเข้าไปข้างหน้าเนื้อหาเดิมที่มีอยู่ เช่น **<p>Welcome to</p>** เมื่อใช้ **.prepend("**Dom using prepend jQuery Tutorials **")** จะได้ string ใหม่มีค่าเป็น **<p>** Dom using prepend jQuery Tutorials **Welcome to</p>**  
  
**ตัวอย่าง**  
  
**Code (jQuery)**

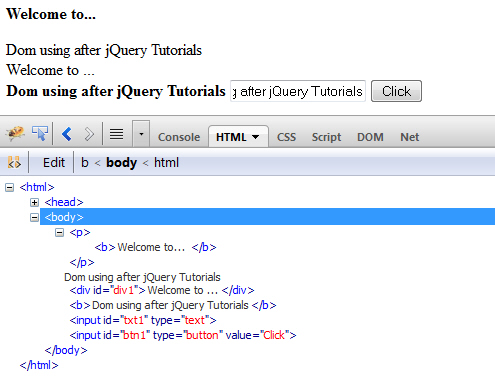
|  |
| --- |
| <html>  <head>  <title>Dom using prepend jQuery Tutorials </title>  <script type="text/javascript" src="jquery-1.6.4.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("p").prepend($("#txt1").val());  $("#div1").prepend("<b>"+$("#txt1").val()+"</b>");  });  });  </script>  </head>  <body>  <p><b>Welcome to... </b></p>  <div id="div1">Welcome to ...</div>  <input type="text" id="txt1">  <input type="button" id="btn1" value="Click">  </body>  </html> |

  
  
**คำอธิบาย**  
ทั้ง <p> และ <div id="div1"> ใช้ method prepend() ซึ่งได้ผลลัพธ์เหมือนกัน โดย prepend() สามารถรับค่าที่เป็น HTML Tags ได้เช่นเดียวกัน  
  
**ผลลัพธ์ทีได้ในมุมมองของ HTML Tags**

|  |
| --- |
| <p><b> Dom using prepend jQuery Tutorials Welcome to... </b></p> <div id="div1"> Dom using prepend jQuery Tutorials Welcome to ...</div> |

**4. การใช้งาน method ของ $(selector).after(content)**  
สำหรับ **method append()** ใช้สำหรับการเพิ่มเนื้อหาหลังจากสิ้นสุด element ที่ได้เลือกไว้ เช่น **<p>Welcome to</p>** เมื่อใช้ **.after("Dom using after jQuery Tutorials")** จะได้ string ใหม่มีค่าเป็น **<p>Welcome to</p> Dom using after jQuery Tutorials**  
  
**ตัวอย่าง**  
  
**Code (jQuery)**

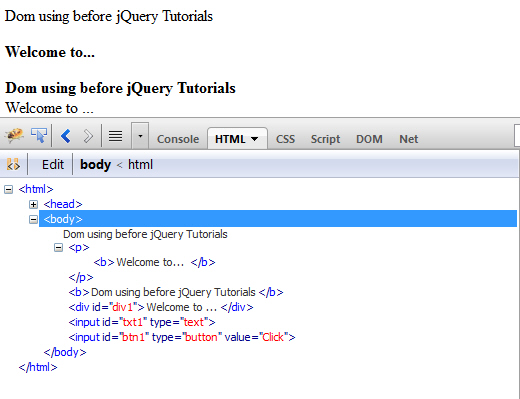
|  |
| --- |
| <html>  <head>  <title>Dom using after jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("p").after($("#txt1").val());  $("#div1").after("<b>"+$("#txt1").val()+"</b>");  });  });  </script>  </head>  <body>  <p><b>Welcome to... </b></p>  <div id="div1">Welcome to ...</div>  <input type="text" id="txt1">  <input type="button" id="btn1" value="Click">  </body>  </html> |

  
  
**คำอธิบาย**  
ทั้ง <p> และ <div id="div1"> เมื่อใช้ method after() ซึ่งได้ผลลัพธ์เหมือนกัน คือมีการแสดงผลขึ้นบรรทัดใหม่ เพราะมีการเพิ่มเนื้อหาหลังจากจบ Tags นั้นแล้ว  
  
**ผลลัพธ์ทีได้ในมุมมองของ HTML Tags**

|  |
| --- |
| <p><b>Welcome to... </b></p> Dom using after jQuery Tutorials  <div id="div1">Welcome to ...</div><b> Dom using after jQuery Tutorials </b> |

**5. การใช้งาน method ของ $(selector).before(content)**  
สำหรับ **method before()** ใช้สำหรับการเพิ่มเนื้อหาก่อนเริ่ม element ที่ได้เลือกไว้ เช่น **<p>Welcome to</p>** เมื่อใช้ **.before("**Dom using before jQuery Tutorials **")** จะได้ string ใหม่มีค่า Dom using before jQuery Tutorials **<p>Welcome to</p>**   
  
**ตัวอย่าง**  
  
**Code (jQuery)**

|  |
| --- |
| <html>  <head>  <title>Dom using before jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("p").before($("#txt1").val());  $("#div1").before("<b>"+$("#txt1").val()+"</b>");  });  });  </script>  </head>  <body>  <p><b>Welcome to... </b></p>  <div id="div1">Welcome to ...</div>  <input type="text" id="txt1">  <input type="button" id="btn1" value="Click">  </body>  </html> |

  
  
**คำอธิบาย**  
ทั้ง <p> และ <div id="div1"> เมื่อใช้ method before() ซึ่งได้ผลลัพธ์เหมือนกัน คือมีการแสดงผลขึ้นบรรทัดใหม่ เพราะมีการเพิ่มเนื้อหาก่อนเริ่ม Tags นั้น  
  
**ผลลัพธ์ทีได้ในมุมมองของ HTML Tags**

Dom using before jQuery Tutorials<p><b>Welcome to... </b></p>  
<b> Dom using before jQuery Tutorials</b> <div id="div1">Welcome to ...</div>

**สรุปจากบทความนี้**  
**$(selector).text(content)** ใช้ในการอ่านค่าและส่งค่าในรูปแบบของ Text  
**$(selector).html(content)** ใช้ในการอ่านค่าและส่งค่าในรูปแบบของ HTML Tags  
**$(selector).append(content)** เพิ่มเนื้อหาเข้าไปใน Tags โดยต่อจากเนื้อหาที่มีอยู่แล้ว  
**$(selector).prepend(content)** เพิ่มเนื้อหาเข้าไปใน Tags โดยเพิ่มแทรกเนื้อหาก่อนเนื้อหาที่มีอยู่แล้ว  
**$(selector).after(content)** เพิ่มเนื้อหาหลังจากจบ Tags  
**$(selector).before(content)** เพิ่มเนื้อหาก่อนเริ่ม Tags  
  
จากบทความนี้จะเห็นว่า jQuery ได้ช่วยให้การเขียน Code สั้นลงและง่ายขึ้นมาก Syntax ที่เขียนก็เข้ากันได้อย่างดี และเข้าใจง่าย ซึ่งถ้าหากอ่านและลองปฏิบัติตามให้เข้าใจทุกส่วน ก็จะพอเข้าใจหลักการทำงานพื้นฐานของ jQuery ได้เป้นอย่างดี

**Ajax**

|  |
| --- |
|  |
|  |

**jQuery Ajax รูปแบบคำสั่งการใช้ jQuery กับ Ajax บน jQuery Library Framework**  
**jQuery กับ Ajax** ใน **jQuery** เองก็ได้ออกแบบ function ที่ทำงานร่วมกับ **Ajax** ได้อย่างมีประสิทธิ์ภาพ ด้วย Code คำสั่งที่สั้นมาก ๆ และยังไม่ต้องกังวลเรื่องปัญหา **Cross ของ Web Browser** อีกด้วย ด้วยประสิทธิ์และการออกแบบที่เป็นมาตรฐานและถูกต้องตามหลักของ **W3C CSS Validation** รองรับการทำงานทุกเว็บบราวเซอร์ และสามารถทำงานได้รวดเร็ว ด้วย Library ไฟล์ขนาดเล็ก องค์กรหลายๆ แห่งได้ประกาศใช้ **jQuery เป็น Core** หลักในการทำงานร่วมกับ **Ajax** เช่น Google ได้ประกาศใช้ Library ของ **jQuery** ใช่ ร่วมกับ Application หลายตัวทีได้พัฒนาขึ้น Open Source ดัง ๆ หลายตัว หรือแม้กระทั้ง Microsoft ก้ได้นำ jQuery มาเป็นส่วนหนึ่งของ Framework ที่ใช้ในการพัฒนา ร่วมกับ ASP.NET (Web Application) ที่ทำงานร่วมกับ **Ajax** ด้วย เหตุผลและความในสามารถของ jQuery นี่เอง คาดว่าในอนาคต jQuery นั้นมีความจำเป็นมาก ๆ ที่จะต้องเรียนรู้ และ ศึกษาการใช้งานอย่างจริงจัง เพื่อสร้างเสริมประสบการณ์และความรู้ใหม่ ๆ ในการพัฒนา **Web Application** ในรูปแบบต่าง ๆ   
  
**พื้นฐาน jQuery Ajax Syntax**  
  
**jQuery Ajax**

|  |
| --- |
| $(selector).load(url,data,callback); |

**1. ตัวอย่างการใช้งาน jQuery แบบง่าย ๆ ผ่าน method load()**  
  
**Code (server\_side.php)**

|  |
| --- |
| <?php  for($i=1;$i<=10;$i++)  {  echo $i."<br>";  }  ?> |

**Code (jQuery)**

|  |
| --- |
| <html>  <head>  <title>Ajax Load jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $("#div1").load(‘server\_side.php');  });  });  </script>  </head>  <body>  <div id="div1"></div>  <input type="button" id="btn1" value="Load">  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
เมื่อคลิกที่ปุ่ม Load จะมีการ load ข้อมูลจากไฟล์ **04\_ajax\_load**.php ซึ่งเป็นการทำงานง่าย ๆ ด้วยคำสั่งสั้น ๆ ของ jQuery   
  
  
**2. ตัวอย่างการใช้งาน jQuery แบบง่าย ๆ ผ่าน method ajax()** โดยวิธีนี้สามารถตรวจสอบสถานะการทำงาน และกำหนดคุณสมบัติต่าง ๆ ได้  
  
**Code (server\_side.php)**

|  |
| --- |
| <?php  for($i=1;$i<=10;$i++)  {  echo $i."<br>";  }  ?> |

**Code (jQuery)**

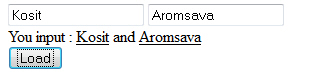
|  |
| --- |
| <html>  <head>  <title>ajax method jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $.ajax({ url: "server\_side.php" })  .success(function(data) { $("#div1").html(data); })  .error(function() { $("#div1").html("error"); })  .complete(function() { $("#div1").after("Ajax load finished"); });  });  });  </script>  </head>  <body>  <div id="div1"></div>  <input type="button" id="btn1" value="Load">  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
เมื่อคลิกที่ปุ่ม Load จะมีการ load ข้อมูลจากไฟล์ server\_side.php โดยมีการตรวจสอบผลการทำงานว่า   
**.success** เมื่อทำงานสมบูรณ์ไม่มีข้อผิดพลาด  
**.error** เมื่อมีการทำงานผิดพลาด เช่น หาไฟล์ไม่เจอ 404   
**.complete** เมื่อ Ajax มีการทำงานเสร็จสิ้น ไม่ว่าจะ suscess หรือ error ก็ตาม  
  
  
**3. ตัวอย่างการใช้งาน jQuery Ajax ส่งค่าแบบ $\_POST**  
  
**Code (**server\_side\_post.PHP**)**

|  |
| --- |
| <?php  echo "You input : <u>".$\_POST["data1"]."</u> and <u>".$\_POST["data2"]."</u>";  ?> |

**Code (jQuery)**

|  |
| --- |
| <html>  <head>  <title>Ajax Post jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $.post("server\_side\_post.php", {  data1: $("#txt1").val(),  data2: $("#txt2").val()},  function(data){  $("#div1").html(data);  }  );  });  });  </script>  </head>  <body>  <input type="text" id="txt1">  <input type="text" id="txt2">  <div id="div1"></div>  <input type="button" id="btn1" value="Load">  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
จะมีการส่งค่าตัวแปรแบบ $\_POST โดยส่งค่า #txt1 และ #txt2 หรือถ้าต้องการส่งค่ามากกว่านี้ก็สามารถเพิ่มตัวแปรได้แบบง่าย ๆ เช่น

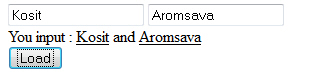
|  |
| --- |
| <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $.post("webpage.php", {  data1: $("#txt1").val(),  data2: $("#txt2").val(),  data3: $("#txt3").val(),  data4: $("#txt4").val(),  data5: $("#txt5").val()},  function(result){  $("#div1").html(result);  }  );  });  });  </script> |

**4. ตัวอย่างการใช้งาน jQuery Ajax ส่งค่าแบบ $\_GET**  
  
**Code (webpage.php)**

|  |
| --- |
| <?php  echo "You input : <u>".$\_GET["data1"]."</u> and <u>".$\_GET["data2"]."</u>";  ?> |

**Code (jQuery)**

|  |
| --- |
| <html>  <head>  <title>Ajax Get jQuery Tutorials</title>  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $.get("server\_side\_get.php", {  data1: $("#txt1").val(),  data2: $("#txt2").val()},  function(data){  $("#div1").html(data);  }  );  });  });  </script>  </head>  <body>  <input type="text" id="txt1">  <input type="text" id="txt2">  <div id="div1"></div>  <input type="button" id="btn1" value="Load">  </body>  </html> |

**Screenshot**  
  
  
  
**คำอธิบาย**  
จะมีการส่งค่าตัวแปรแบบ $\_GET โดยส่งค่า #txt1 และ #txt2 หรือถ้าต้องการส่งค่ามากกว่านี้ก็สามารถเพิ่มตัวแปรได้แบบง่าย ๆ เช่น

|  |
| --- |
| <script type="text/javascript">  $(document).ready(function(){  $("#btn1").click(function(){  $.get("webpage.php", {  data1: $("#txt1").val(),  data2: $("#txt2").val(),  data3: $("#txt3").val(),  data4: $("#txt4").val(),  data5: $("#txt5").val()},  function(result){  $("#div1").html(result);  }  );  });  });  </script> |