



รายงานเรื่อง

How to Google

นำเสนอ

ผศ.ดร.มานพ พันธุ์โคกกรวด

จัดทำโดย

61070044	นายณภัทร	พรบุญเรือง
61070065	นาย ทศวรรษ	จันทะพิทักษ์
61070084	นาย ธรรมสรณ์	ตันติยาภินันท์
61070087	นาย ชิตวิฑู	โตสงวน
61070190	นาย รุ่งวรารพร	คุตะนนท์

รายงานประกอบการศึกษารายวิชา

06016312 Web Technology

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำนำ

รายงานฉบับนี้มีจุดประสงค์ในการจัดทำ เพื่อศึกษาเกี่ยวกับวิธีการใช้งาน เทคนิค เคล็ดลับต่างๆของ
การใช้งานของ Google มาเป็นหัวข้อในการศึกษาเพื่อประกอบกับเนื้อหาที่ได้เรียนในรายวิชา 06016312
Web Technology

ผู้จัดทำได้จัดทำหัวข้อนี้เพราะ Google เป็นเว็บไซต์ที่ผู้ใช้บริการอินเทอร์เน็ตส่วนใหญ่ทั่วโลกได้ใช้
บริการแต่ผู้ใช้บริการไม่ทราบวิธีการใช้ที่ถูกต้องหรือเทคนิคต่างๆในการใช้งาน และด้วยความที่หัวข้อนี้น่าสนใจ
คณะผู้จัดทำจึงได้นำมาเป็นหัวข้อในการทำรายงานนี้

คณะผู้จัดทำ

วันที่ 14 พฤษภาคม 2562

สารบัญ

เรื่อง	หน้า
บทนำ	1
ที่มาและความสำคัญ	1
วัตถุประสงค์	1
ประโยชน์ที่คาดว่าจะได้รับ	2
เทคโนโลยีที่ใช้	3
HTML	3
CSS	7
Bootstrap 4	16
JavaScript	24
JQuery	26
ตัวอย่างเว็บไซต์และเทคโนโลยีที่ใช้	30
อ้างอิง	32

บทนำ

ปัจจุบันเทคโนโลยีสารสนเทศและอินเทอร์เน็ตได้เข้ามามีบทบาทต่อการดำเนิน ชีวิตของเรามากขึ้น ซึ่งเราอาจไม่รู้สึกรู้สว่าอินเทอร์เน็ตกลายเป็นปัจจัยที่สำคัญต่อการดำรงชีวิตในยุคข้อมูล ข่าวสารมีความสำคัญ คนหันมาบริโภคข้อมูลข่าวสารกันมากขึ้น นอกจากเทคโนโลยีอินเทอร์เน็ตเปรียบเสมือนถนนสำหรับการเข้าไปถึงข้อมูลที่ต้องการ เรายังต้องการเครื่องมือที่จะสามารถสร้าง เนื้อหาและข้อมูลต่าง ไว้รองรับการเข้าถึง นั่นก็คือเทคโนโลยีเว็บไซต์ ซึ่งเป็นตัวกลางคอยให้ ข้อมูลต่างๆ แก่ผู้ใช้โดยการพัฒนาของเทคโนโลยีอินเทอร์เน็ตและเว็บไซต์ได้ถูกเปลี่ยนแปลงจากเดิมไปมาก

ที่มา และความสำคัญ

ในยุคที่เทคโนโลยีการสื่อสารผ่านอินเทอร์เน็ตกำลังเป็นที่นิยมและมีผลกระทบในทุกๆ ด้านในปัจจุบัน ทำให้ทุกคน ทุกสังคมต้องมีการปรับตัว และพัฒนาให้ทันต่อการเปลี่ยนแปลงในโลกของการสื่อสาร และการพัฒนาของโลกเวิลด์ไวด์เว็บ (World Wide Web; WWW) จากยุคแรก คือ Web 1.0 ซึ่งมีลักษณะเป็น Static Web คือมีการนำเสนอข้อมูลทางเดียว (one-way communication) ด้วยการแปลงข้อมูลข่าวสารที่มีอยู่รอบตัวเราให้อยู่ในรูปของดิจิทัล (Digital) เช่น หนังสือพิมพ์ นิตยสาร หรือการโฆษณาตามหน้าเว็บไซต์ โดยผู้ใช้สามารถอ่านได้แต่ ไม่สามารถเข้าร่วมในการสร้าง ข้อมูลได้ แต่เมื่อก้าวเข้าสู่ยุคที่ 2 ของเทคโนโลยีคือ Web 2.0 เป็นยุคที่ทำให้อินเทอร์เน็ตมีศักยภาพในการใช้งานมากขึ้น เน้นให้ผู้ใช้มีส่วนร่วมในการ สร้างสรรค์ Web 2.0 ทำให้ความนิยมของ Social Media มีการเติบโตอย่างไม่หยุดยั้งและมีแนวโน้มของผู้ใช้บริการทั่วโลก ปัจจุบัน Social Network Website ต่าง ๆ ก็มีการพัฒนาและเปิดโอกาสให้ผู้ใช้ได้เข้าไปมีส่วนร่วมในการใช้ประโยชน์ต่างๆ

เนื่องด้วย Google นั้นเป็นสิ่งที่ผู้ใช้งานกันอย่างแพร่หลายทั่วโลก โดยผู้คนส่วนใหญ่นิยมใช้ในการค้นคว้าหาในสิ่งที่ตัวเองต้องการ ทั้งการหาความรู้ งานวิจัย การทำงาน หรือเพื่อความบันเทิง เป็นต้น ผู้จัดทำเล็งเห็นว่า Google เปรียบเสมือนเส้นทางที่ผู้ใช้เลือกใช้เพื่อเข้าถึงข้อมูลที่ตนต้องการและเว็บไซต์เป็นสิ่งที่ผู้คนสามารถเข้าถึงได้โดยง่ายและสะดวก คณะผู้จัดทำจึงมีความต้องการในการสร้างเว็บไซต์เพื่อให้ผู้ใช้สามารถเข้ามาศึกษาหาความรู้เกี่ยวกับการใช้กูเกิ้ลและเทคนิคการใช้งานกูเกิ้ลเพื่อความสะดวกสบายของผู้ใช้งานมากขึ้น

วัตถุประสงค์

- เพื่อศึกษาและพัฒนาเว็บไซต์เรื่อง How To google
- เพื่อศึกษาค้นคว้าเรื่องที่สนใจเกี่ยวกับ Google
- เพื่อให้ผู้เรียนสามารถพัฒนาเว็บไซต์ได้ด้วยตนเองและนำมาประยุกต์ใช้ให้เข้ากับปัจจุบันมากขึ้น
- เพื่อให้ความรู้แก่ผู้ที่มีความสนใจเกี่ยวกับ Google

ประโยชน์ที่คาดว่าจะได้รับ

- ได้ความรู้ในเรื่องการพัฒนาเว็บไซต์
- เว็บไซต์ให้ความรู้การเทคนิคการค้นหากูเกิ้ล
- เว็บไซต์ให้ความรู้การเทคนิคการให้คีย์ลัดต่างๆ
- เว็บไซต์ให้ความรู้ที่มาประวัติผลิตภัณฑ์ของกูเกิ้ล
- สามารถพัฒนาเว็บไซต์ และนำมาประยุกต์ใช้งานในชีวิตจริงได้

เทคโนโลยีที่ใช้ในการทำโครงงาน

1. HTML (Hypertext Markup Language)

HTML คือ ภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล HTML ย่อมาจากคำว่า Hypertext Markup Language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิงก์ (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่างๆที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink

ความเป็นมาของ HTML เริ่มขึ้นเมื่อปี 1980 เมื่อ Tim Berners Lee เสนอต้นแบบสำหรับนักวิจัยใน CERN เพื่อแลกเปลี่ยนเอกสาร ข้อมูลด้านการวิจัย โดยใช้ชื่อว่า Enquire ในปี 1990 เค้าได้เขียนโปรแกรมเบราเซอร์ และทดลองรันบนเซิร์ฟเวอร์ที่เค้าพัฒนาขึ้น HTML ได้รับการรู้จักจาก HTML Tag ซึ่งมีอยู่ 18 Tag ในปี 1991

HTML ถูกพัฒนาจาก SGML และ Tim ก็คิดเสมือนว่า HTML เป็นโปรแกรมน้อยของ SGML อยู่ในตอนนั้น ต่อมาในปี 1996 เพื่อกำหนดมาตรฐานให้ตรงกัน W3C World Wide Web Consortium จึงเป็นผู้กำหนดสเปกทั้งหมดของ HTML และปี 1999 HTML 4.01 ก็ถือกำเนิดขึ้น โดยมี HTML 5 ซึ่งเป็น Web Hypertext Application ถูกพัฒนาต่อมาในปี 2004 นอกจากนี้ยังมีการพัฒนาไปเป็น XHTML ซึ่ง คือ Extended HTMLซึ่งมีความสามารถและมาตรฐานที่รัดกุมกว่าอีกด้วย โดยอยู่ภายใต้การควบคุมของ W3C (World Wide Web Consortium)

ไทม์ไลน์ของ HTML

- ปี 1980 นักฟิสิกส์ Tim Berner-Lee เสนอระบบ ENQUIRE เพื่อแลกเปลี่ยนเอกสารใน CERN
- ปี 1989 Tim Berner-Lee เขียน Memo ถึง ระบบ Internet Based Hypertext System
- ปี 1990 Tim Berner-Lee เขียนเบราเซอร์และซอฟต์แวร์ฝั่งเซิร์ฟเวอร์
- ปี 1991 เอกสารเผยแพร่ HTML ชื่อ HTML Tag เผยแพร่ครั้งแรก มี Tag ทั้งหมด 18 Tag และมีการใช้งาน 11 Tag จวบจน HTML 4
- กลางปี 1993 Tim และ Dan Conolly เสนอข้อกำหนดของ HTML เป็นครั้งแรกต่อสาธารณะ
- ปี 1994 HTML + HTML draft หมดอายุ HTML WORKING GROUP ดำเนินการต่อ
- ปี 1995 HTML 2.0 เสร็จสิ้น เป็นข้อกำหนด HTML แรกที่ถูกนำไปใช้เป็นมาตรฐานต่อไปในภายหน้า
- 24 พฤศจิกายน 1995: HTML 2.0 เผยแพร่เป็น RFC 1866และบรรจุ RFC 2070 (สากล) 14 มกราคม 1997

- 12 กันยายน 1996: HTML 3.2 เผยแพร่เป็นคำแนะนำของ W3C เป็นรุ่นแรกที่ได้รับการพัฒนาและได้รับมาตรฐานโดย W3C เนื่องจาก IETF ได้ปิดการทำงานของ HTML Working Group แล้ว
- 18 ธันวาคม 1997: HTML 4.0 ได้รับการเผยแพร่เป็นคำแนะนำ W3C ตอนแรกชื่อรหัสว่า "Cougar"
- พฤษภาคม 2000: HTML 4.01 เผยแพร่เป็นคำแนะนำ W3C มีรูปแบบ 3 รูปแบบเช่นเดียวกับ HTML 4.0
- วันที่ 28 ตุลาคม 2014: HTML5 ได้รับการเผยแพร่เป็นคำแนะนำ W3C
- 1 พฤศจิกายน 2016: HTML 5.1 ได้รับการเผยแพร่เป็นคำแนะนำ W3C
- 14 ธันวาคม 2017: HTML 5.2 ได้รับการเผยแพร่เป็นคำแนะนำ W3C

ปัจจัยพื้นฐานในการเขียนโฮมเพจโดยใช้ภาษา HTML ในการพัฒนาโฮมเพจด้วยภาษา HTML นั้นต้องปัจจัยพื้นฐานที่จำเป็น ดังนี้

1. โปรแกรมแก้ไขข้อความ (Text Editor) เช่น โปรแกรม Emacs, PICO หรือ vi บนระบบปฏิบัติการ UNIX, โปรแกรม Simple Text บนระบบปฏิบัติการ Macintosh หรือ โปรแกรม Notepad บนระบบปฏิบัติการ Windows และในกรณีที่ต้องการใช้โปรแกรมประยุกต์ประเภท WYSIWYG (What You See Is What You Get) ก็มีโปรแกรมให้เลือกใช้ได้มากมาย เช่น FrontPage, DreamWeaver, Adobe PageMill ฯลฯ

2. โปรแกรมตกแต่งรูปภาพ (Graphics Design) เช่น โปรแกรม PaintShop pro, Ulead Photo Impact ฯลฯ

3. โปรแกรมเว็บเบราว์เซอร์ เช่น โปรแกรม Netscape Navigator หรือโปรแกรม Internet Explorer เพื่อใช้ในการตรวจสอบผลลัพธ์ของการเขียน HTML

รูปแบบของการเขียน HTML

ในการเขียน HTML นั้นเราจะต้องจัดวางรูปแบบของแท็กต่าง ๆ ให้ถูกต้องโดยแท็กพื้นฐานที่ต้องมีการเขียน HTML ได้แก่

TAG	รายละเอียด
<HTML>...</HTML>	เป็นแท็กกำหนดถึงจุดเริ่มต้นและจุดสิ้นสุดของเอกสาร HTML
<HEAD>...</HEAD>	เป็นแท็กกำหนดจุดเริ่มต้นและจุดสิ้นสุดของส่วนกำหนดค่าเริ่มต้นของเอกสาร HTML เช่น ชื่อของเอกสาร

<TITLE>...</TITLE> เป็นแท็กกำหนดชื่อของเอกสาร

<BODY>...</BODY> เป็นแท็กกำหนดจุดเริ่มต้นและจุดสิ้นสุดของส่วนแสดงข้อมูลของเอกสาร

ซึ่งเราจะต้องจัดวางตำแหน่งของแท็กต่าง ๆ ดังต่อไปนี้

ตัวอย่าง 1.1 การจัดวางตำแหน่งของแท็ก

```
<HTML>
  <HEAD>
    <TITLE>ชื่อเอกสาร</TITLE>
  </HEAD>
  <BODY>
    //ข้อมูลเอกสาร
  </BODY>
</HTML>
```

ขั้นตอนการเขียนโฮมเพจด้วย HTML ในการเขียนโฮมเพจด้วย HTML โดยใช้โปรแกรมแก้ไขข้อความ (Text Editor) Notepad มีขั้นตอนดังนี้

1. เปิดโปรแกรมแก้ไขข้อความ (Text Editor) Notepad
2. พิมพ์แท็ก HTML
3. บันทึกแฟ้มข้อมูล โดยมีนามสกุลเป็น .htm หรือ .html
4. เปิดโปรแกรมเว็บเบราว์เซอร์ (Internet Explorer หรือ Netscape Navigator) แล้วทำการเปิดแฟ้มข้อมูลที่บันทึกไว้จากข้อ 3. เพื่อตรวจสอบผลลัพธ์บนจอภาพว่าแสดงผลได้ตามที่ต้องการหรือไม่

การกำหนดชื่อเอกสาร

ในการเขียน HTML นั้นจำเป็นจะต้องกำหนดชื่อเอกสารด้วยเสมอ เพราะการกำหนดชื่อเอกสารจะทำให้ผู้เข้าชมสามารถรู้ถึงชื่อเรื่องของเอกสารชุดนั้นได้ทันที และสะดวกต่อการสืบค้นข้อมูล การกำหนดชื่อของเอกสารทำได้โดยการพิมพ์ชื่อของเอกสารที่ต้องการไว้ระหว่างแท็ก <TITLE > กับแท็ก </TITLE>

ตัวอย่าง 1.2 การกำหนดชื่อเอกสาร

```
<HTML>
<HEAD>
<TITLE>My First Page</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

เมื่อเราเปิดแฟ้ม HTML นี้ด้วยโปรแกรมเว็บเบราว์เซอร์ จะเห็นข้อความว่า My First Page ปรากฏที่ ส่วนของ Title Bar ของโปรแกรมเว็บเบราว์เซอร์

การใส่ข้อมูลในเอกสาร

ข้อความ รูปภาพ เสียง หรือ คลิปวิดีโอ (Video Clip) ที่ต้องการให้แสดงบนพื้นที่แสดงผลของ โปรแกรมเบราว์เซอร์ ทำได้โดยการพิมพ์ที่ต้องการไว้ระหว่างแท็ก <BODY> กับ </BODY>

ตัวอย่าง 1.3 การใส่ข้อมูลในเอกสาร

```
<HTML>
<HEAD>
<TITLE> My First Page </TITLE>
</HEAD>
<BODY>
welcome to my first home page,
</BODY>
</HTML>
```

เมื่อเราเปิดแฟ้ม HTML นี้ด้วยโปรแกรมเว็บเบราว์เซอร์ ข้อความ My First page ปรากฏที่ส่วนของ Title Bar ข้อความ Welcome to my first home page. จะแสดงที่พื้นที่แสดงข้อมูลของโปรแกรมเว็บเบราว์เซอร์

2. CSS (Cascading Style Sheets)

CSS ย่อมาจาก Cascading Style Sheet มักเรียกโดยย่อว่า "สไตร์ชีต" คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ "Style") ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลล์พ์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดยองค์กร World Wide Web Consortium หรือ W3C

ประโยชน์ของ CSS

1.CSS มีคุณสมบัติมากกว่า tag ของ html เช่น การกำหนดกรอบให้ข้อความ รวมทั้งสี รูปแบบของข้อความที่กล่าวมาแล้ว

2.CSS นั้นกำหนดที่ต้นของไฟล์ html หรือตำแหน่งอื่น ๆ ก็ได้ และสามารถมีผล กับเอกสารทั้งหมด หมายถึงกำหนด ครั้งเดียวจุดเดียวก็มีผลกับการแสดงผลทั้งหมด ทำให้เวลาแก้ไขหรือปรับปรุงทำได้สะดวก ไม่ต้องไล่ตามแก้ tag ต่างๆ ทั่วทั้งเอกสาร

3.CSS สามารถกำหนดแยกไว้ต่างหากจาก ไฟล์เอกสาร html และสามารถนำมาใช้ร่วม กับเอกสารหลายไฟล์ได้ การแก้ไขก็แก้เพียง จุดเดียวก็มีผลกับเอกสารทั้งหมด

CSS กับ HTML / XHTML นั้นทำหน้าที่คนละอย่างกัน โดย HTML / XHTML จะทำหน้าที่ในการวางโครงร่างเอกสารอย่างเป็นรูปแบบ ถูกต้อง เข้าใจง่าย ไม่เกี่ยวข้องกับการแสดงผล ส่วน CSS จะทำหน้าที่ในการตกแต่งเอกสารให้สวยงาม เรียกได้ว่า HTML /XHTML คือส่วน coding ส่วน CSS คือส่วน design CSS การแก้สไตร์ กำหนดสไตร์กันอีกครั้งแต่จะแทนที่จะกำหนดสไตร์ในโค้ด HTML เราจะใช้ CSS แทน โดยในไฟล์ CSS ของเว็บเรา (ยังไม่ต้องสนใจว่ามันอยู่ที่ไหน) เราจะเพิ่มโค้ด CSS สำหรับสไตร์ชื่อ sidenote ลงไปแบบด้านล่าง

```
.sidenote {
    background-color: #ffffb6;
    padding: 12px;
}
```

ชื่อ sidenote เป็นชื่อที่ตั้งขึ้นมาเองให้สื่อความหมาย (side note แปลหยาบๆ ได้ว่า เกร็ดความรู้) แต่จริงๆ จะตั้งชื่อสไตล์อะไรก็ได้ พอกำหนดสไตล์ขึ้นมาแล้ว หากเราต้องการให้ย่อหน้าไหนมีพื้นหลังสีเหลืองดังโค้ดด้านล่าง

`<p class="sidenote">ย่อหน้านี้มีพื้นหลังสีเหลือง ใช้เพื่อแสดงเกร็ดความรู้เพิ่มเติม</p>` โค้ดด้านบนเป็นการระบุให้แท็ก `<p>` (ย่อหน้า) ใช้สไตล์ CSS ชื่อ sidenote ผลลัพธ์ที่ได้คือ ย่อหน้าดังกล่าวจะมีพื้นหลังสีเหลือง (ให้ผลเหมือนกับวิธีระบุสไตล์ใน HTML)

ย่อหน้านี้มีพื้นหลังสีเหลือง ใช้เพื่อแสดงเกร็ดความรู้เพิ่มเติม

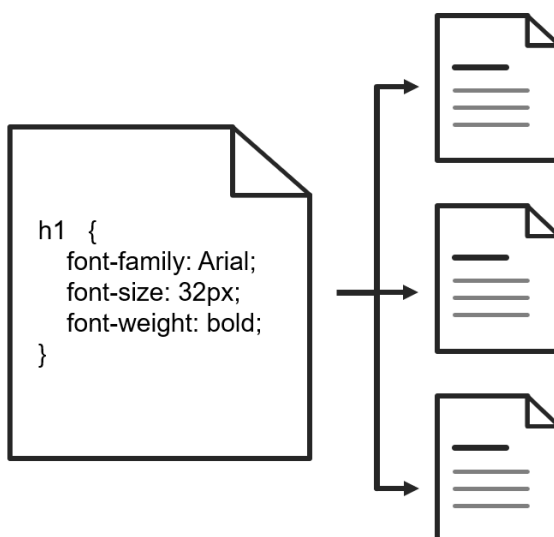
แต่ประโยชน์ของใช้ CSS กำหนดสไตล์คือ หากวันหนึ่งเราอยากเปลี่ยนพื้นหลังเป็นสีเขียว (รหัสสี #a3e4d7) เราก็แค่แก้ค่าสไตล์ของ sidenote ในไฟล์ CSS แค่อันเดียวเท่านั้น

```
.sidenote {
    background-color: #a3e4d7;
    padding: 12px;
}
```

เพียงเท่านี้สีพื้นหลังก็จะเป็นแบบด้านล่าง

ย่อหน้านี้มีพื้นหลังสีเขียว ใช้เพื่อแสดงเกร็ดความรู้เพิ่มเติม

ข้อดีของวิธีนี้คือ ไม่ว่าเว็บคุณจะมีย่อหน้าที่ใช้โค้ด `<p class=sidenote>...</p>` พื้นหลังของย่อหน้าพวกนี้จะเปลี่ยนเป็นสีเขียวทั้งหมด จากการแก้โค้ด CSS แค่อันเดียวซึ่งเป็นที่มาของชื่อ Cascading Style Sheet ครับ ความหมายก็คือ คุณแก้ค่าจุดเดียว (ในไฟล์ CSS) แล้วค่าที่แก้จะ “กระจาย (cascade)” ไปทั้งเว็บไซต์



การใช้ CSS Selector เลือกแท็ก HTML

แท็ก HTML ก็คือพวก `<h1>`, `<div>`, `<p>`, `` พวกนี้ครับ ซึ่งสามารถใช้ CSS selector กับแท็กเหล่านี้ได้ อย่างโค้ด CSS ด้านล่าง ผมกำหนดให้แท็ก `<h1>` มีตัวอักษรสีม่วง

```
h1 {
    color: purple;
}
```

ให้สังเกตว่าคราวนี้ไม่มีจุดนำหน้าชื่อ ตรงนี้มีความสำคัญ

- ถ้า selector มีจุดนำหน้า (.h1) แปลว่าเราต้องการเลือก HTML element ที่มีคลาสชื่อ h1
- ถ้าไม่มีจุดนำหน้า (h1) แปลว่าเราต้องการเลือกแท็ก h1 หมายความว่าถ้าโค้ด HTML เป็นแบบนี้

```
<h1>หัวข้อหลัก</h1>
<p class="h1">ย่อหน้าแรก</p>
<h2>หัวข้อย่อย</h2>
```

จะมีเพียงหัวข้อหลักตัวเดียวเท่านั้นที่จะมีตัวอักษรสีม่วง ส่วนย่อหน้า `<p class="h1">` นั้นไม่เกี่ยว เพราะในโค้ด CSS ตัว selector ของเราคือ h1 (ไม่มีจุด) ไม่ใช่ .h1 หมายความว่าเราเลือกแท็ก h1 ไม่ได้เลือก HTML element ที่มีคลาสชื่อ h1 (หวังว่าจะไม่งงนะครับ)

การเขียน Selector แบบขั้นสูง

ตัวอย่างที่ผ่านๆ มา เป็นการเขียน selector แบบง่ายๆ แต่ความจริง selector อาจมีโครงสร้างซับซ้อน

ตัวอย่างเช่น ถ้ามีเครื่องหมาย comma (,) มาคั่นแบบนี้

```
h1, h2, p {
  color: purple;
}
```

แปลว่าเราต้องการให้แท็ก h1, h2, และ p มีตัวอักษรสีม่วง ประโยชน์ของการใช้ comma แบบนี้คือ เราไม่ต้องเขียนสไตล์เดิมซ้ำๆ กันแบบนี้

```
h1 {
  color: purple;
}
h2 {
  color: purple;
}
p {
  color: purple;
}
```

Selector สามารถเขียนแบบนี้ได้ด้วย

```
div img {
  border: #cccccc solid 4px;
}
```

ในโค้ดด้านบน เราต้องการกำหนดสไตล์ให้แท็ก img ที่อยู่ภายใต้แท็ก div หมายความว่าถ้าโค้ด HTML เป็นแบบนี้

```

<div></div>

```

สไตล์ที่กำหนดจะถูกเช็ตให้กับ image_2.jpg เท่านั้น เนื่องจากเป็นแท็ก img เดียวที่อยู่ภายใต้แท็ก div สามารถผสม class selector กับ type selector ได้ด้วย แบบนี้

```
.border-image img {
    border: #cccccc solid 4px;
}
```

โค้ดด้านบนเป็นการกำหนดสไตล์ให้แท็ก `img` ที่อยู่ภายใต้ HTML element ที่มีคลาสชื่อ `border-image` หมายความว่าถ้าโค้ด HTML เป็นแบบนี้

```
<div class="border-image">
  
</div>
<div>
  
</div>

```

สไตล์ที่กำหนดจะถูกใช้ได้กับ `image_1.jpg` เท่านั้น เนื่องจากเป็นแท็ก `img` เดียวที่อยู่ภายใต้ HTML element ที่มี class ชื่อ `border-image` Selector อีกแบบที่เราใช้บ่อยๆ ก็คือแบบนี้

```
div.embed-container {
    margin-bottom: 24px;
}
```

ในโค้ดด้านบน เราต้องการกำหนดสไตล์ให้แท็ก `div` ที่มีคลาสชื่อ `embed-container` หมายความว่าถ้าโค้ด HTML เป็นแบบนี้

```
<div>
  <p class="embed-container">...</p>
</div>
<div class="border-image">
  ....
</div>
<div class="embed-container">
```

```
....
</div>
```

สไตล์ที่กำหนดจะถูกเซตให้กับแท็ก div ตัวสุดท้ายเท่านั้น (ส่วนแท็ก p ที่มีคลาส embed-container ไม่เกี่ยว เพราะไม่ใช่แท็ก div) เป็นเพียงตัวอย่างการเขียน CSS selector ส่วนหนึ่งเท่านั้น ความจริง สามารถเขียน selector ได้อีกหลายรูปแบบ ซึ่งจริงๆ syntax พวกนี้ไม่ได้ (และไม่คิดจะจำด้วย) ปกติเวลาผมจะแก้ CSS จะเปิดดู CSS reference ในเน็ตประกอบไปด้วยเสมอ (แล้วลองแก้โดยใช้ WordPress live preview)

CSS Property

```
selector {
    property1: value1;
    property2: value2;
    ...
}
```

หลังจากที่เขียน selector เพื่อเลือก element ที่ต้องการแก้สไตล์เรียบร้อยแล้ว ก็ต้องมาเขียน property เพื่อระบุว่าจะอยากแก้ค่าอะไร (ความกว้าง, ความสูง, สีตัวอักษร, สีพื้นหลัง, ฯลฯ) ซึ่ง property นี้มีเป็นร้อยๆ ค่าเลยครับ แต่ property ที่ผมใช้บ่อยๆ มีอยู่ประมาณ 8 ตัว เลยจะขอสอนแค่นี้เท่านั้น

Margin ใช้เวลาต้องการสร้างช่องว่างระหว่าง HTML element

```
.custom-style {
    margin: 25px;
}
```

สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้มี margin = 25 พิกเซล

Padding ใช้เวลาต้องการสร้างช่องว่างข้างใน HTML element

```
.custom-style {
    padding: 30px;
}
```

สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้มี padding = 30 พิกเซล

background-color ใช้เวลาต้องการเซตค่าสีพื้นหลัง

```
.custom-style {
    background-color: AliceBlue;
}
```

สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้มีสีพื้นหลัง = AliceBlue

Color ใช้เวลาต้องการเซตสีตัวอักษร

```
.custom-style {
    color: DarkBlue;
}
```


สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้มีสีตัวอักษร = DarkBlue

font-size ใช้เพื่อกำหนดขนาดตัวอักษร

```
.custom-style {
    font-size: 30px;
}
```

สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้มีตัวอักษรขนาด 30 พิกเซล

font-weight ใช้ font-weight เพื่อกำหนดความหนาของตัวอักษร ค่าที่ใช้ได้มีตั้งแต่ 100, 200, ..., 900 โดยค่าความหนา 400 จะเป็นความหนาของตัวอักษรปกติ

```
.custom-style {
    font-weight: 600;
}
```

สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้เป็นตัวหนา = 600

Border ใช้ border เพื่อสร้างกรอบให้กับ HTML element โดยสามารถกำหนด ขนาด, สี, และเลือกได้ว่าจะให้เป็นเส้นทึบ (solid) หรือเส้นประ (dotted)

```
.custom-style {
    border: 5px solid red;
}
```

สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้มีกรอบเป็นเส้นทึบสีแดงขนาด 5 พิกเซล

text-align ใช้ text-align เพื่อจัดตำแหน่งของตัวอักษร (left, right, center)

```
.custom-style {  
    text-align: center;  
}
```

สไตล์ด้านบนให้ผลลัพธ์ด้านล่าง

ย่อหน้านี้อยู่ตรงกลาง

3. Bootstrap 4

Bootstrap 4 คือ Front-end Framework ที่ใช้ในการพัฒนาเว็บไซต์ส่วนการแสดงผลให้กับผู้ใช้ ซึ่งสามารถเข้ามาช่วยกำหนดกรอบการทำงานของกลุ่มผู้พัฒนาเว็บไซต์ให้เป็นไปในแนวทางเดียวกัน ซึ่งสามารถรองรับได้ทุก Smart Device หรือเรียกว่า Responsive Web หรือ Mobile First โดยคำว่าคำว่า Bootstrap ในภาษาอังกฤษมักจะหมายถึง “สิ่งที่ช่วยทำให้ง่ายขึ้น” หรือ “สิ่งที่ทำได้ด้วยตัวของมันเอง” ซึ่งในที่นี้จะหมายความว่า หากใช้ Bootstrap แล้วก็ไม่จำเป็นต้องไปหาอะไรมาเพิ่มอีก ซึ่งช่วยให้ง่ายต่อการพัฒนา ก่อนหน้านี้ Bootstrap ได้ออกมาแล้ว 3 เวอร์ชัน ปัจจุบันได้ออกมาเป็นเวอร์ชันที่ 4 แล้ว นั่นก็คือ Bootstrap 4

Bootstrap เป็นขั้นตอนในการพัฒนาเว็บหนึ่งในขั้นตอนที่สำคัญนั่นก็คือ การออกแบบ โดยจะออกแบบหน้าตาของ web application ให้ดูง่ายต่อการใช้งาน

ย้อนกลับไปในสมัยที่ในโลกนี้ยังไม่มีสิ่งที่เรียกว่า Bootstrap การออกแบบเว็บต้องร่างแบบในโปรแกรม Photo Edit ซึ่งอาจจะใช้โปรแกรมยอดนิยมอย่าง Photoshop หรือโปรแกรมอื่น ๆ ขึ้นอยู่กับความถนัดของผู้ออกแบบ จากนั้นนำแบบร่างที่ได้มาตัดเป็นภาพต่าง ๆ ก่อนนำไป coding เป็น web application ถือว่าเป็นขั้นตอนที่กินเวลาเป็นอย่างมาก

ในปัจจุบันคนส่วนใหญ่นิยมเปิดดูเว็บจาก mobile เป็นหลัก (Mobile First) ดังนั้นผู้พัฒนาจึงต้องออกแบบเว็บให้ตอบสนองกับทุกขนาดหน้าจออีกด้วย

Mark Otto และ Jacob Thornton ทีมพัฒนาของ Twitter Inc. เล็งเห็นปัญหาที่เกิดขึ้น จึงพัฒนา Bootstrap ขึ้นมาเพื่อตอบโจทย์ในเรื่องของออกแบบและ responsive web design ซึ่ง Bootstrap มีระบบ Grid มาช่วยในจัดรูปแบบหน้าตาของเว็บ และการคำนวณค่าหน้าจอให้ด้วย

ส่วนประกอบของ Bootstrap

- Layout จะประกอบไปด้วย containers, grid, media object และ responsive เพื่อใช้ในการตกแต่งรูปร่างหน้าตาของเว็บ โดย containers เป็นการกำหนดขอบเขตวัตถุหรือข้อมูล , grid ใช้เพื่อแบ่ง containers เป็นคอลัมน์ย่อย ซึ่งสามารถทำได้ถึง 12 คอลัมน์ , media object เป็นการจัดการ media ต่างๆที่อยู่บนหน้าเว็บ ส่วน responsive จะเป็นฟังก์ชันที่ช่วยทำให้หน้าเว็บสามารถเป็นรูปแบบให้มีความเหมาะสมตามจอแสดงผลของผู้ใช้
- Base CSS เป็นการกำหนด style ของ display ซึ่งจะใช้ร่วมกับ HTML elements พื้นฐาน เช่น typography (ลักษณะของตัวอักษร ไม่ว่าจะเป็น ความหนา,ขนาดตัวอักษร และ ลักษณะอื่น) , tables (รูปแบบของตาราง ไม่ว่าจะเป็นสี การแบ่งแถวหรือคอลัมน์) และ images (ขนาดของภาพ, ลักษณะของขอบรูปภาพ)
- Components : เป็นการรวบรวม สำหรับสิ่งที่เราต้องใช้บ่อยๆ ไม่ว่าจะเป็น buttons, dropdowns , input groups , navigation , alerts(การแจ้งเตือน), navbar , form control (รูปแบบการกรอกแบบชุดข้อมูล) และ อื่นๆอีกมากมาย

- JavaScript : jQuery plugins ต่างๆ ไม่ว่าจะเป็น modal , carousel หรือ tooltip เพื่อที่จะทำให้เว็บของเรามีลูกเล่นและมีความน่าใช้งานมากขึ้น

ความแตกต่างระหว่าง Bootstrap 4 และ Bootstrap 3

1. หน่วยที่ใช้ในการกำหนดขนาด

- Px เป็นการกำหนดขนาดตัวอักษรซึ่งเมื่อมีการขยายหรือหดจอ ตัวอักษรจะมีขนาดเท่าเดิม
- Rem เพิ่มมาใน bootstrap4 เป็นการกำหนดขนาดตัวอักษรแบบ responsive คือ เมื่อขยายจอตัวอักษรจะขยายตาม และเมื่อหดจอตัวอักษรก็จะเล็กลงตามไปด้วย

Grid

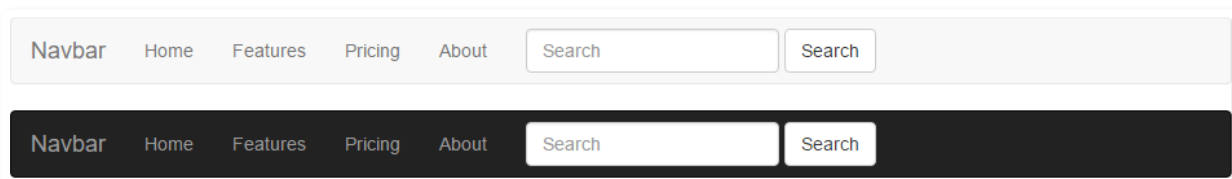
- Grid tiers เป็นการกำหนดขนาด ตามขนาดของจอที่เราจะใช้ให้เหมาะสม โดย bootstrap 4 จะมีการเพิ่มขนาด xl มาเพื่อใช้รองรับจอที่มีขนาดใหญ่ขึ้นโดยขนาดจอที่เหมาะสมคือ 1140 px
- Offsetting Columns ใน bootstrap3 ใช้คำสั่ง “ Col-(ขนาดคอลัมน์)-offset-(จำนวนคอลัมน์ที่ต้องการขยับไป) ” ส่วนใน bootstrap4 ใช้คำสั่ง “ Offset-(ขนาดคอลัมน์)- (จำนวนคอลัมน์ที่ต้องการขยับไป) ” ซึ่งผลลัพธ์ที่ได้จะเหมือนกัน เพียงแต่เปลี่ยนรูปแบบของคำสั่งที่ใช้

2. Table

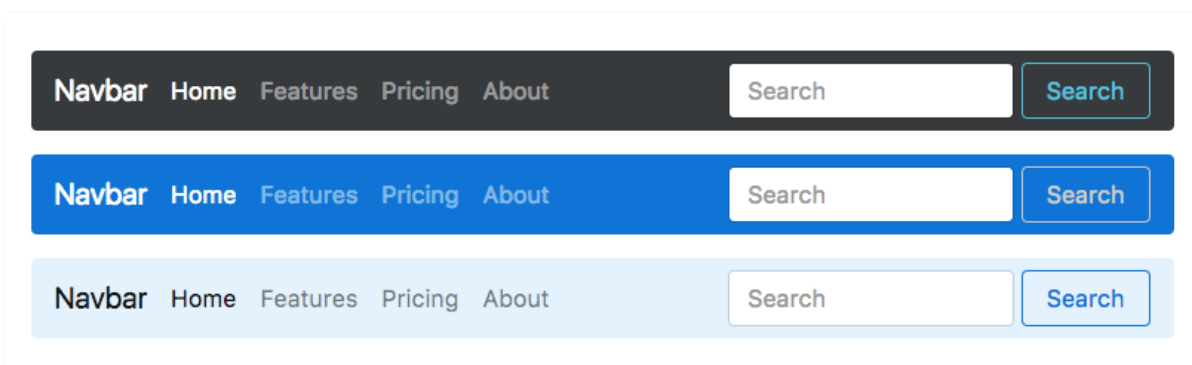
- Condensed table รูปแบบเดิมจาก bootstrap3 จะใช้ class table-condensed แต่ใน bootstrap4 จะใช้ class table-sm ซึ่งมีรูปแบบการเขียนที่ง่ายขึ้น แต่ผลลัพธ์ที่ได้จะยังคงเหมือนเดิม
- Responsive table ใน bootstrap3 จะเขียน class table-responsive ใน <div> ส่วนใน bootstrap4 สามารถเขียน class table-responsive ในตารางได้เลย

3. Navbars

เป็นแถบบาร์เมนู ซึ่งในBootstrap 3จะมีแค่แถบบาร์เมนูสีเทาซีดๆและสีดำให้ใช้ ดูธรรมดา แต่ถ้าเป็นในBootstrap 4ก็จะมีสีอื่นๆให้เลือกใช้เพิ่มขึ้น ดูมีสีสันมากขึ้น ส่วนคำสั่งที่ใช้ก็จะมีบางส่วนที่แตกต่างกันเช่น คำสั่งจัดตำแหน่งถ้าเป็นในBootstrap 3จะใช้คำสั่ง .navbar-right , .navbar-left ถ้าเป็นBootstrap 4จะใช้คำสั่ง .pull()-right , .pull()-left , .pull()-none นอกจากนั้นยังมีคำสั่งในการสร้างแถบแสดงหมายเลขหน้า (Pagination)ที่ในBootstrap 3จะใช้ .pagination ใน ส่วนBootstrap 4จะมีการใช้ .page-item ในและ ใช้ .page-link ใน <a>



NavbarsของBootstrap 3



NavbarsของBootstrap 4

4. Form จะเป็นในส่วนของฟอร์มการกรอกข้อมูล
- Horizontal Forms (ฟอร์มแบบแนวนอน) ในส่วนของ Bootstrap 3 จะใช้คำสั่ง `.form-horizontal`, `.form-control-label` และ Bootstrap 4 จะใช้คำสั่ง `.row` กับ `grid` `.col-form-label`

Horizontal Forms Bootstrap 3

Horizontal Forms Bootstrap 4

- Check boxes and Radio Buttons ในส่วนของ Bootstrap 3 จะใช้คำสั่ง `.radio` , `.radio-inline` , `checkbox` , `checkbox-inline` และ Bootstrap 4 จะใช้คำสั่ง `.form-check` , `.form-check-label` , `.form-check-input` , `.form-check-inline`

- ☐ Option one is this and that—be sure to include why it's great
- ☐ Option two is disabled
- ☒ Option one is this and that—be sure to include why it's great
- ☐ Option two can be something else and selecting it will deselect option one
- ☐ Option three is disabled

Check boxes and Radio Buttons Bootstrap 3

Radios

- ☒ Option one is this and that—be sure to include why it's great
- ☐ Option two can be something else and selecting it will deselect option one
- ☐ Option three is disabled

Checkbox ☐ Check me out

Check boxes and Radio Buttons Bootstrap 4

- Form Control Size (ปรับขนาดของฟอร์ม) ในส่วนของ Bootstrap 3 จะใช้คำสั่ง `.input-(Grid Tier)` และ Bootstrap 4 จะใช้คำสั่ง `.form-control-(Grid Tier)`
- Help text (แสดงข้อความเพื่อช่วยเหลือ) ในส่วนของ Bootstrap 3 จะใช้คำสั่ง `.help-block` และ Bootstrap 4 จะใช้คำสั่ง `.form-text`

Input with help text

A block of help text that breaks onto a new line and may extend beyond one line.

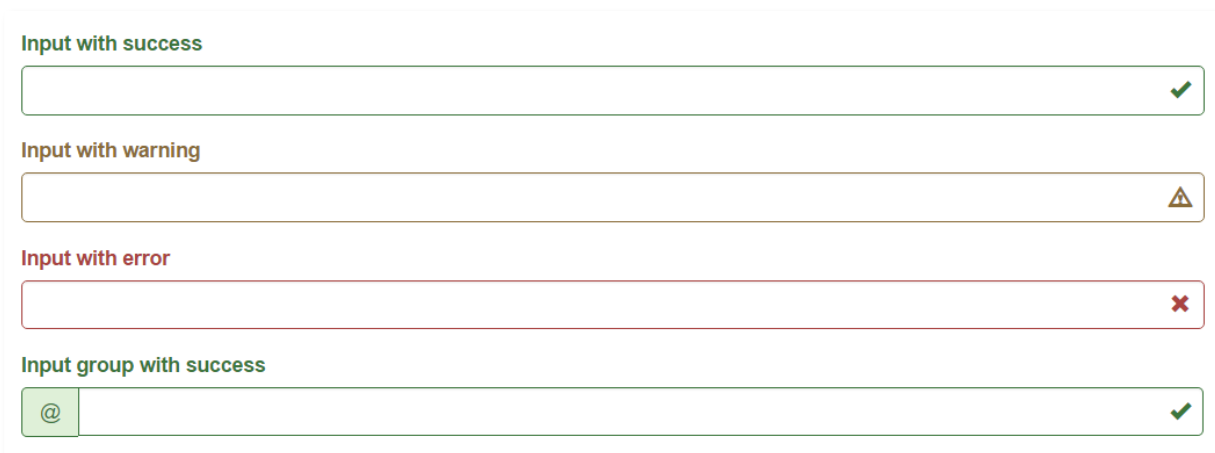
Help text Bootstrap 3

Password

Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.

Help text Bootstrap 4

- Validation and Feedback Icons (ตรวจสอบความถูกต้อง การตอบกลับของไอคอน) ในส่วนของ Bootstrap 3 จะไม่มีคลาสเฉพาะ แต่ใช้ `.form-control-(glyphicons)` ทำได้ แต่ Bootstrap 4 จะใช้คำสั่ง `.has-feedback`



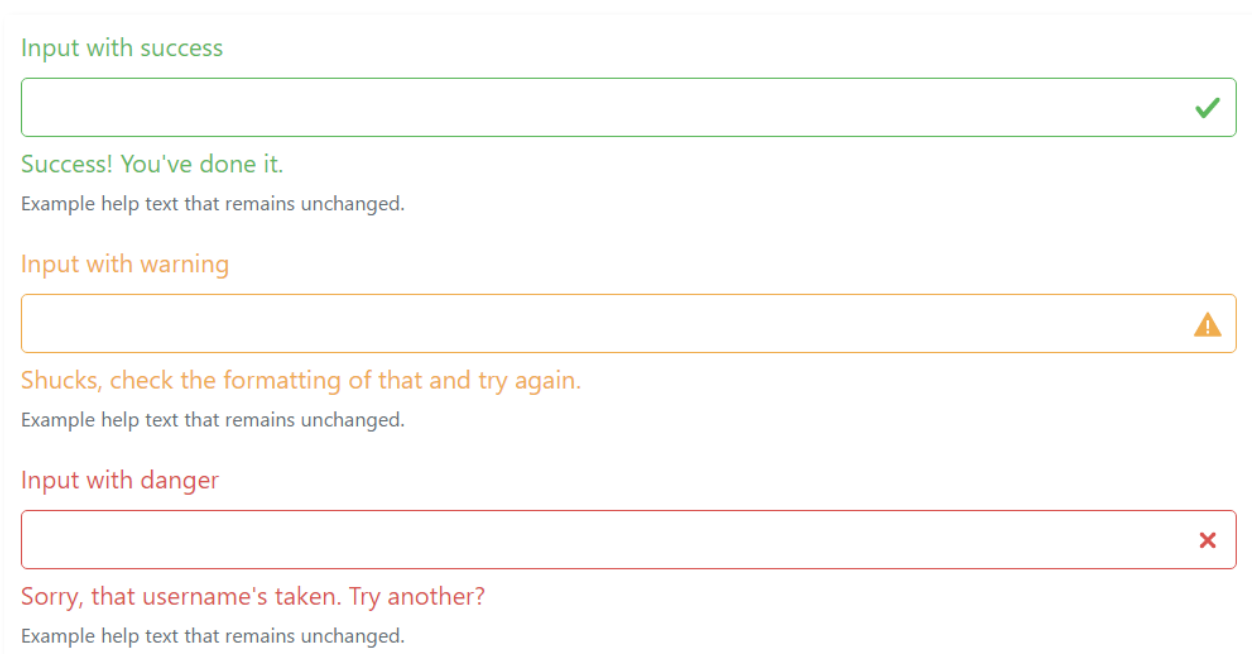
Input with success

Input with warning

Input with error

Input group with success

Validation and Feedback Icons Bootstrap 3



Input with success

Success! You've done it.
Example help text that remains unchanged.

Input with warning

Shucks, check the formatting of that and try again.
Example help text that remains unchanged.

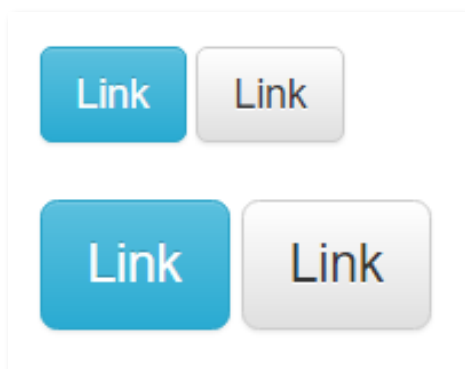
Input with danger

Sorry, that username's taken. Try another?
Example help text that remains unchanged.

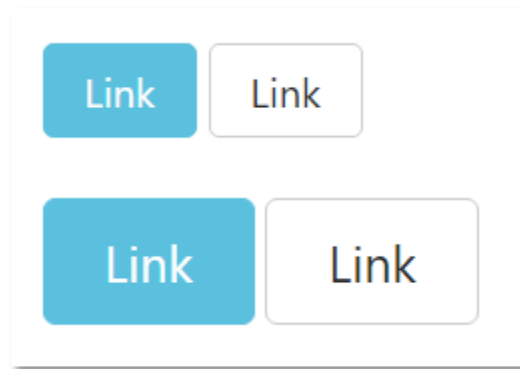
Validation and Feedback Icons Bootstrap 4

5. Button

Semantic Styles ของตัว button ใน Bootstrap 3 สามารถระบุลักษณะได้ด้วยคำสั่ง `.btn-default` , `.btn-info` และ ใน Bootstrap 4 สามารถใช้คำสั่ง `.btn-secondary` , `.btn-info` ซึ่งลักษณะของ button ระหว่าง Bootstrap 3 กับ 4 ก็มีความแตกต่างกันค่อนข้างชัดเจน



Semantic Styles Bootstrap 3



Semantic Styles Bootstrap 4

6. Image

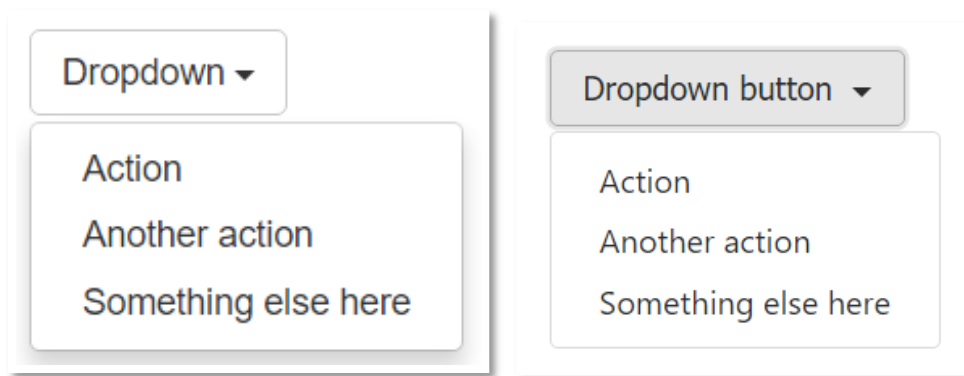
- Responsive Images (การปรับขนาดอัตโนมัติตามขนาดหน้าจอ) ในส่วนของ Bootstrap 3 จะใช้คำสั่ง `.img-responsive` และในส่วนของ Bootstrap 4 จะใช้คำสั่ง `.img-fluid`
- Image Alignment (ตำแหน่งของรูปภาพ) ในส่วนของ Bootstrap 3 จะใช้คำสั่ง `.pull-right` (จัดภาพให้อยู่ด้านขวา), `.pull-left` (จัดภาพให้อยู่ด้านซ้าย), `.center-block` (จัดภาพให้อยู่ตรงกลาง) และในส่วนของ Bootstrap 4 จะใช้ `.m-x-auto` แทน `center-block` โดย ทั้ง 2 แบบนี้จะให้ผลที่ไม่แตกต่างกัน



Image Alignment Bootstrap 3 & 4

7. Drop-down

- Structure ใน Bootstrap 3 ใช้ ``, `` ส่วน Bootstrap 4 ใช้ `.dropdown-menu` รวม `.dropdown-item` ใน `<div>`
- Menu Headers ใน Bootstrap 3 ใช้ `.dropdown-header` ใน `` ส่วน Bootstrap 4 ใช้ `.dropdown-header` ใน `<h1>`, `<h2>`
- Dividers ใน Bootstrap 3 ใช้ `.divider` ส่วน Bootstrap 4 ใน `` ใช้ `.dropdown-divider` ใน `<div>`
- Disabled Menu Items ใน Bootstrap 3 ใช้ `.disabled` ใน `` ส่วน Bootstrap 4 ใช้ `.disabled` ใน `<a>`



Drop-down Bootstrap 3

Drop-down Bootstrap

8. Blockquotes

Blockquotesของทั้งBootstrap 4และ3มีลักษณะที่เหมือนกันแต่คำสั่งในการใช้จะไม่เหมือนกัน ถ้าเป็นBootstrap 3จะใช้<blockquote>ได้เลย แต่ถ้าเป็นBootstrap 4จะต้องใช้<blockquote class="blockquote">

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

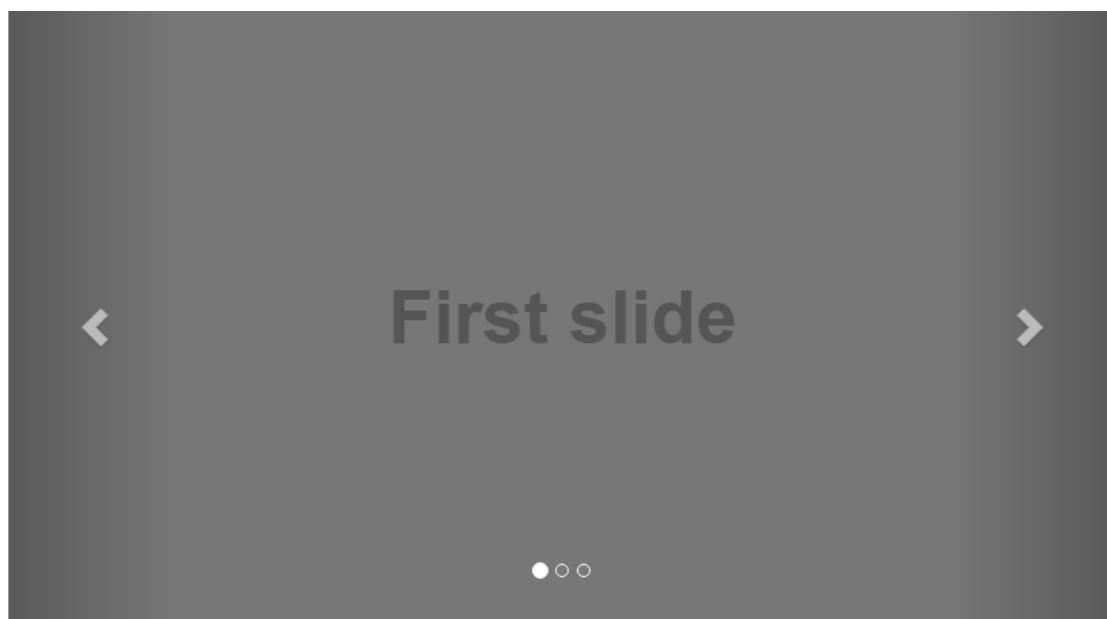
9. Breadcrumbs

Breadcrumbsของทั้ง2เวอร์ชันจะต่างกันตรงที่คำสั่งในการใช้งานตรงที่เวลาที่เราจะเขียนหัวข้อย่อยๆ ในBreadcrumbs ซึ่งถ้าเป็นของBootstrap 4 จะต้องใส่class="breadcrumb-item"ลงไปในด้วย แต่ถ้าเป็นในBootstrap3ไม่จำเป็นต้องใส่

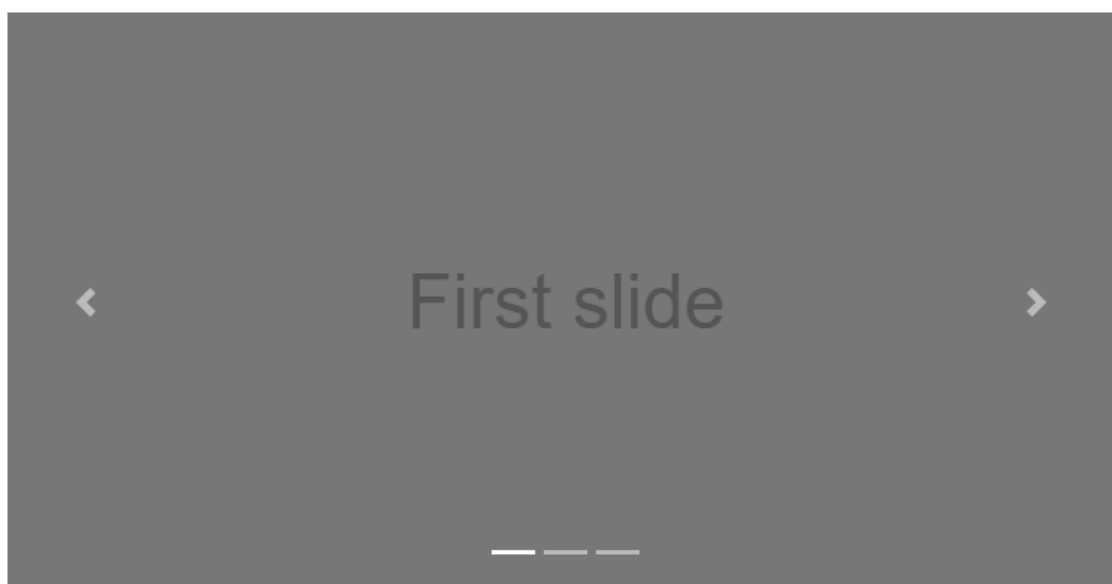
[Home](#) / [Library](#) / [Data](#) / [Bootstrap](#)

10. Carousels

เป็นการโชว์ภาพแบบที่เลื่อนไปซ้าย-ขวาได้ ซึ่งถ้าเป็นในBootstrap 3ภาพแต่ละอันจะถูกใส่ไว้ในclass item แต่ถ้าเป็นBootstrap 4จะใส่ไว้ในclass carousels-item



CarouselsของBootstrap 3



CarouselsของBootstrap 4

4. JavaScript

JavaScript คือ

ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์โดยใช้ร่วมกับ HTML เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น

ประวัติ JavaScript

เริ่มพัฒนาโดย Brendan Eich พนักงานบริษัทเน็ตสเคป โดยขณะนั้นจาวาสคริปต์ใช้ชื่อว่า โมคา (Mocha) และภายหลังได้เปลี่ยนชื่อมาเป็น ไคฟสคริปต์ และเป็น จาวาสคริปต์ในปัจจุบัน รูปแบบการเขียน

ภาษาที่ใช้ คล้ายคลึงกับภาษาซี รุ่นล่าสุดของจาวาสคริปต์คือ 2.0 ซึ่งตรงกับมาตรฐานของ ECMAScript ภาษาจาวาสคริปต์ไม่มีความสัมพันธ์กับ ภาษาจาวา (Java) และ เจสคริปต์ (JScript) แต่อย่างใด ยกเว้นแต่โครงสร้างภาษาที่มีลักษณะคล้ายคลึงกัน เนื่องมาจากได้รับการพัฒนาต่อมาจากภาษาซีเหมือน ๆ กัน และมีชื่อที่คล้ายคลึงกันเท่านั้น สำหรับเจสคริปต์ (JScript) หลังจากที่จาวาสคริปต์ประสบความสำเร็จ โดยมีเว็บเบราว์เซอร์จากหลายๆ บริษัทนำมาใช้งาน ทางไมโครซอฟท์จึงได้พัฒนาภาษาโปรแกรมที่ทำงานในลักษณะคล้ายคลึงกับจาวาสคริปต์ขึ้น และตั้งชื่อว่าเจสคริปต์ ซึ่งทำงานได้กับเบราว์เซอร์อินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet Explorer) เท่านั้น เริ่มใช้ครั้งแรกใน อินเทอร์เน็ตเอกซ์พลอเรอร์ 3.0 เมื่อ สิงหาคม พ.ศ. 2539 โดยสร้างตามมาตรฐาน ECMA 262

JavaScript เป็นมีลักษณะดังนี้

1. เป็นภาษาสคริปต์แบบ lightweight programming language (ภาษาสคริปต์แบบสั้นๆ)
2. JavaScript สามารถใช้ร่วมกับ HTML และ CSS เพื่อสร้างและพัฒนาเว็บไซต์ ที่ใช้ได้กับ web browser รุ่นใหม่ๆ และใช้ได้ทั้ง PCs, laptops, tablets, smart phones, ด้วย

ประโยชน์ของ Javascript

- ทำให้สามารถเขียนโปรแกรมแบบง่ายๆ ได้ โดยไม่ต้องพึ่งภาษาอื่น
- มีคำสั่งที่ตอบสนองต่อผู้ใช้งาน
- ใช้ตรวจสอบข้อมูลได้ เช่น เตือนเมื่อกรอกรูปแบบข้อมูลผิดพลาด
- ใช้ตรวจสอบผู้ใช้ได้ เช่น ตรวจสอบว่าผู้ใช้งานใช้ Browser อะไร
- สร้าง Cookies ได้
- สร้างลูกเล่นให้เว็บไซต์ได้

การใช้งานภาษา JavaScript

จาวาสคริปต์ เป็นภาษาในรูปแบบของภาษาโปรแกรมแบบโพรโทไทป์ โดยมีโครงสร้างของภาษาและไวยากรณ์อยู่บนพื้นฐานของภาษาซี ปัจจุบันมีการใช้จาวาสคริปต์ที่ฝังอยู่ในเว็บเบราว์เซอร์ในหลายรูปแบบ เช่น ใช้เพื่อสร้างเนื้อหาที่เปลี่ยนแปลงเสมอภายในเว็บเพจ ใช้เพื่อตรวจสอบความถูกต้องของข้อมูลที่ใช้กรอกก่อนนำเข้าระบบ ใช้เพื่อเข้าถึงข้อมูลที่อยู่ภายใต้โครงสร้างแบบ Document Object Model (DOM) เป็นต้น

นอกจากนี้จาวาสคริปต์ยังถูกฝังอยู่ในแอปพลิเคชันต่าง ๆ นอกเหนือจากเว็บเบราว์เซอร์ได้อีกด้วย เช่น widget ของ ยาฮู(YaHoo) เป็นต้น โดยรวมแล้วจาวาสคริปต์ถูกใช้เพื่อให้นักพัฒนาโปรแกรม สามารถเขียนสคริปต์เพื่อสร้างคุณสมบัติพิเศษต่างๆ เพิ่มเติมจากที่มีอยู่บนแอปพลิเคชันดั้งเดิม

โปรแกรมใด ๆ ที่สนับสนุนจาวาสคริปต์จะมีตัวขับเคลื่อนจาวาสคริปต์ (JavaScript Engine) ของตัวเอง เพื่อเรียกใช้งานโครงสร้างเชิงวัตถุของโปรแกรมหรือแอปพลิเคชันนั้น ๆ

ตัวอย่างการประกาศใช้งานตัวแปร

ตัวแปร ในจาวาสคริปต์ สามารถประกาศใช้งานได้ด้วยการระบุคำสำคัญ `var` นำหน้าชื่อตัวแปร

```
var x; //ประกาศตัวแปร x, โดยที่ยังไม่มีการใส่ค่า
```

```
var y = 2; //ประกาศตัวแปร y ให้มีค่าเท่ากับ 2
```

ตัวอย่างด้านบน มีการใส่หมายเหตุ ตามหลังการประกาศใช้งานตัวแปร โดยการใส่เครื่องหมายทับ สองตัว (forward slashes)

ตัวอย่างการใช้คอนโซล

คำสั่งที่ใช้ในการตรวจสอบและแสดงผลค่าต่างๆ โดยการเรียกใช้ อ็อบเจกต์ console

```
console.log("Hello world!");
```

ในการใช้งาน JavaScript นั้น จำเป็นต้องใส่ Code ให้อยู่ระหว่างแท็ก `<script>` และ `</script>` โดยตัวคำสั่ง JavaScript นี้จะอยู่ในส่วนแท็ก `<head>` และ `<body>` ของเอกสาร HTML เช่น

JavaScript ในแท็ก `<body>` ตัวอย่างเช่น

```
<html>
<body>
  <script>
    document.write("<h1>Welcome To </h1>");
    document.write("<p>king mongkut's institute of technology ladkrabang</p >");
  </script>
</body>
</html>
```

ผลลัพธ์คือ

Welcome To

king mongkut's institute of technology ladkrabang

การใส่คำสั่ง Javascript นั้นอาจใส่ในส่วนแท็ก `<head>` และ `<body>` แท็กใดแท็กหนึ่งแล้ว หรือทั้งใน `<head>` และ `<body>` พร้อมๆ กันก็ได้

5. jQuery

jQuery คือ JavaScript Library ซึ่งถูกออกแบบมาเพื่อให้การเขียน JavaScript นั้นมีความสะดวกและง่ายขึ้น เพราะว่าการนำ JavaScript เอาไปประยุกต์กับงานจำพวกเว็บ (Client-side JavaScript) นั้นเป็นสิ่งที่ยุ่งยาก ไม่ว่าจะเป็นเรื่องความไม่เข้ากันของ **Web Browser** แต่ละค่าย, DOM หรือ API เป็นต้น ดังนั้น jQuery จึงรวมเอา Object และ Function ต่างๆ ที่จำเป็นมารวบรวมไว้ในรูปแบบของ Library พอเป็นเช่นนี้แล้ว ไม่ว่าโค้ดที่คุณเขียนจะใช้ JavaScript หลายบรรทัดขนาดไหน ก็สามารถทำให้สั้นลงได้ อาจทำให้เหลือสั้นเพียงแค่บรรทัดเดียวเท่านั้น

jQuery ประกอบด้วยฟีเจอร์ต่างๆ ดังนี้

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

ประโยชน์ของ JQuery

- 1.ความสามารถในการทำงานแบบ ajax ก็คือ ทำงานได้ในรูปแบบที่ไม่ต้องโหลดหน้าใหม่ให้คนใช้เห็น
- 2.การสร้าง animation ได้แบบง่ายๆเลย ไม่ว่าจะเป็นทำรูปให้เคลื่อนที่ หรือ DIV
- 3.สามารถจัดการกับ css (style sheet) ของ element นั้นๆได้
- 4.ค้นหา element ที่เราต้องการและจัดการ เพิ่มหรือลบ Attributes ที่เราต้องการได้
- 5.ทำ Effect ต่างๆกับ Element ที่เราต้องการ เช่นการ hide DIV ที่เราต้องการ
- 6.การดักเหตุการณ์ต่างๆ
- 7.รองรับกับภาษาได้ทุกรูปแบบ (asp, php, .net , html ,.....)
- 8.รองรับได้ทุก ๆ Browser ดีกว่าเขียน Javascript แบบเดิม ๆ
- 9.ไม่ต้องเขียนโค้ดใหม่เพราะมีให้ใช้อยู่แล้ว
- 10.ทำให้นักพัฒนาเขียนโค้ดน้อยลงแต่ทำงานได้มากขึ้น
- 11.ความเร็วในการทำงานขนาดเล็ก

การใช้งาน JQuery

การจะเริ่มเขียน JQuery ได้ เราต้องเตรียมอุปกรณ์ให้พร้อมก่อน สิ่งที่เราจำเป็นต้องมีประกอบด้วย

- Text editor เครื่องมือสำหรับเขียน JQuery เราแนะนำ Eclipse สามารถใช้งานได้ทั้ง Window, MAC
- Browser Internet Explorer 8+, Mozilla Firefox, Google Chrome, Safari และอื่น
- Webserver เครื่องคอมพิวเตอร์ที่ติดตั้ง PHP, MySQL, Apache ทำหน้าที่เก็บ Website
- JQuery Library โค้ดและชุดคำสั่งของ JQuery วิธีการดาวน์โหลด สามารถดูได้จากด้านล่าง

ก่อนที่จะเริ่มต้นเรียน jquery นั้นผู้เรียนจำเป็นต้องมีความรู้เรื่อง ,HTML5 , CSS3 , JavaScript เพื่อให้สามารถทำความเข้าใจเกี่ยวกับ jquery ได้ง่ายขึ้น โดยเราต้องดาวน์โหลดและติดตั้งมันเสียก่อน

หลังจากดาวน์โหลดเสร็จแล้วก็เอาไฟล์ jquery.js ไปวางไว้ที่ Root Document หรือจะเอาไปรวมไว้กับไลบรารีอื่นๆ ก็ได้ ขอให้เรารู้ก็แล้วกันว่ามันอยู่ที่ไหนจะได้อ้างอิงไปหาถูกที่ (ตามตัวอย่างใช้ jquery เวอร์ชัน 3.4.1) โดยการใส่โค้ดให้ยึดตามชื่อของไฟล์ jquery ที่เราโหลดมา เช่น

- หากโหลดมาชื่อไฟล์เป็น jquery-3.4.1 ก็ให้ใส่โค้ดเป็น `<script src="/jquery-3.4.1.js">`
 - หากโหลดมาชื่อไฟล์เป็น jquery-3.4.1.min ก็ให้ใส่โค้ดเป็น `<script src="jquery-3.4.1.min.js"></script>`
- โดยมีรูปแบบ ดังนี้

```
< script src="jquery-3.4.1.min.js"></script>
```

โดยเขียนโค้ดข้างต้นไว้ในแท็ก <head> ตัวอย่างเช่น

```
<head>
< script src="jquery-3.4.1.min.js"></script>
<script>
    // เราจะเขียนโค้ดคำสั่ง javascript ต่างๆของเราในส่วนนี้จะ
</script>
</head>
<body>
    <!-- ส่วนตรงนี้เราก็จะใช้เขียนโค้ด html -->
</body>
</html>
```

สำหรับบางคนที่ไม่สะดวกจะดาวน์โหลดไฟล์ jQuery อาจใช้วิธีอ้างอิงจากเซิร์ฟเวอร์อื่นๆได้ดังนี้

1.อ้างอิงจากเซิร์ฟเวอร์ของ google จะใช้ syntax ดังนี้

```
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
</head>
```

2.อ้างอิงจากเซิร์ฟเวอร์ของ Microsoft จะใช้ syntax ดังนี้

```
<head>
<script src="//ajax.aspnetcdn.com/ajax/jquery/jquery-3.4.1.min.js">
</script>
</head>
```

jQuery syntax

jQuery ออกแบบมาให้เลือก HTML element และกำหนด action ให้กับอิลิเมนต์นั้น มี syntax ดังนี้

`$(selector).action()`

selector คือส่วนที่เลือก เช่น element , CSS

action() คือ เมธอดที่ใช้กำหนดการทำงานให้กับ selector

ตัวอย่าง เช่น

- `$(this).hide()` เป็นการซ่อนองค์ประกอบปัจจุบัน
- `$("p").hide()` เป็นการซ่อนองค์ประกอบ "p" ทั้งหมด
- `$(".test").hide()` เป็นการซ่อนองค์ประกอบที่มี class= "test"
- `$("#test").hide()` เป็นการซ่อนองค์ประกอบที่มี id="test"

ลักษณะของการใช้งาน jQuery

การใช้งาน jQuery Library จะใช้งานผ่านฟังก์ชัน `jQuery()` เป็นหลัก ยกตัวอย่างเช่น หากต้องการเข้าถึงแท็ก `<title>` เราสามารถเขียนได้ดังนี้ `jQuery("title")`

เนื่องจากมันเป็นฟังก์ชันหลักที่เราต้องเรียกใช้อยู่ตลอด ผู้คิดค้นจึงเตรียมฟังก์ชัน `$()` ไว้เป็น Shortcut ให้เราเรียกใช้ได้โดยสะดวก ดังนั้นตัวอย่างข้างต้นจึงเขียนได้ใหม่ดังนี้

`$("title")`

จากตัวอย่างที่ยกมาข้างต้นนี้เมื่อเราเข้าถึง Elements ต่างๆได้แล้ว เราก็สามารถเรียกใช้ฟังก์ชันต่างๆ เพื่อดัดแปลงแก้ไข Elements นั้นได้ ไม่ว่าจะเป็น Text, Attribute, หรือ CSS เป็นต้น ตัวอย่างเช่น

```
$("#title").text("Hello jQuery");
```

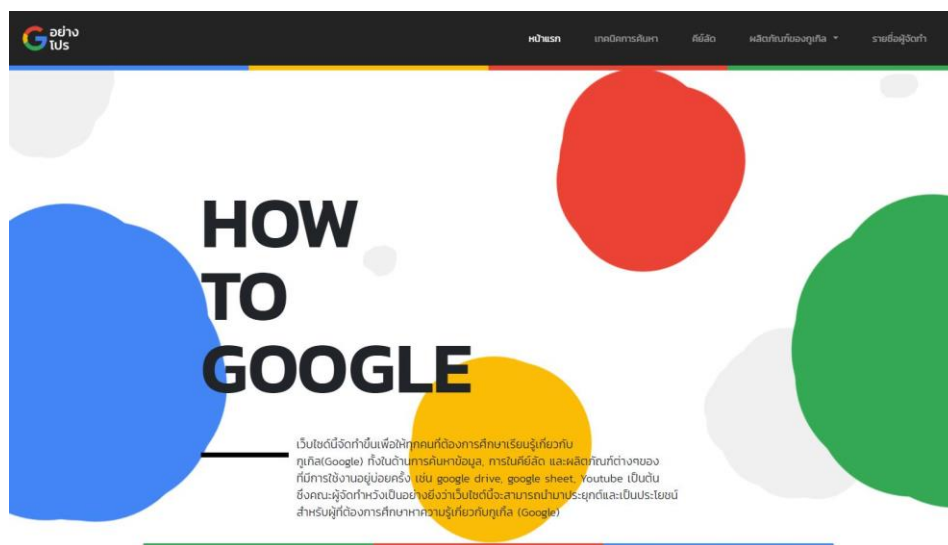
ตัวอย่างข้างต้น ?title? คือ Selector หรือตัวกำหนดว่าจะเข้าถึง Elements ใด โดย Syntax จะใช้รูปแบบเดียวกับ Selector ใน CSS ส่วนเมธอด text() คือเมธอดสำหรับแก้ไขข้อความใน Element นั้นๆ (เมื่อเรา Select อะไรได้แล้ว ทุกอย่างคือออบเจกต์ ดังนั้นฟังก์ชันภายในออบเจกต์ก็คือเมธอดนั่นเอง)

นอกจากนี้ jQuery ยังนำเสนอรูปแบบการเขียนที่เรียกว่า Method Chaining กล่าวคือเมื่อเราได้ ออบเจกต์ (Elements) ที่ต้องการแล้ว เราสามารถเรียกใช้หลายๆเมธอดเป็นลำดับขั้นได้ (และเขียนมันภายในบรรทัดเดียว) ตัวอย่างเช่น `$("#h1").text("One more?").css("font-style","italic");`

ในตัวอย่างข้างต้นนี้เราเข้าถึงแท็ก `<h1>` (สมมุติว่ามีเพียงแท็กเดียวในเอกสาร HTML) และเปลี่ยนแปลงข้อความในแท็กให้เป็น "One more?" ด้วยเมธอด text() จากนั้นเราจึงเรียกเมธอด css() เพื่อกำหนดสไตล์ให้กับแท็ก `<h1>` อีกที นี่แหละเทคนิคที่เรียกว่า Method Chaining และจะ Chain มากกว่าสองครั้งก็ได้

ตัวอย่างเว็บไซต์และเทคโนโลยีที่ใช้

HTML



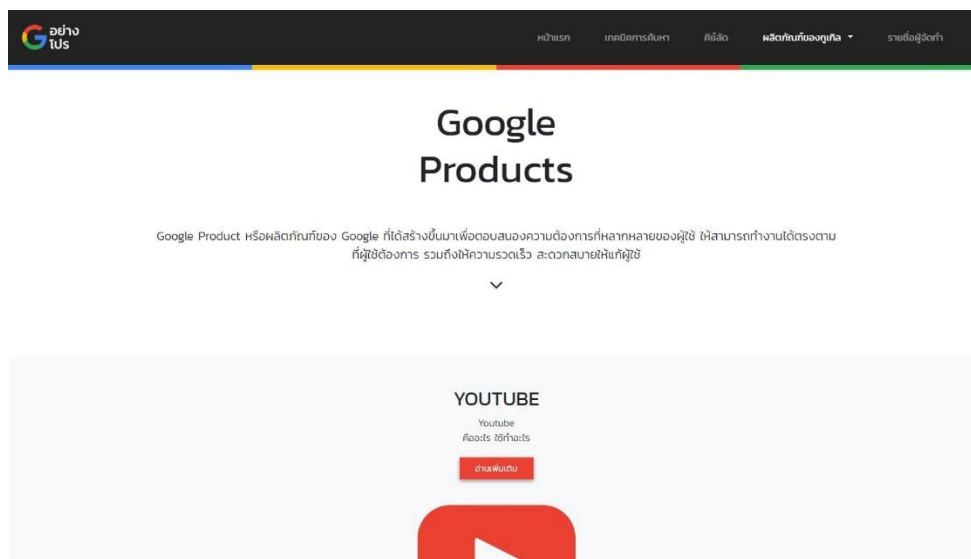
HTML ถูกนำมาใช้ในแทบทุกส่วนของเว็บไซต์ เปรียบเสมือนโครงร่างของเว็บไซต์

CSS



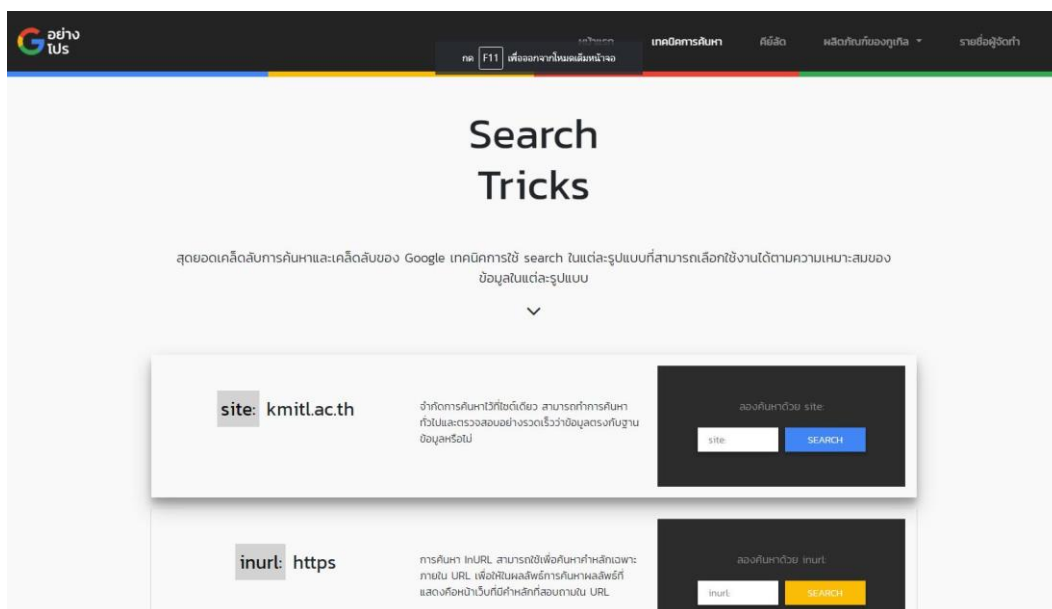
CSS ถูกนำมาใช้ในการตกแต่งควบคุมขนาด และสีของเว็บไซต์โดยทั่วไป และ นำมาใช้ทำ Animation ในส่วนของหน้า Keyboard Shortcut

Bootstrap



Bootstrap เป็นส่วนสำคัญที่ถูกนำมาใช้ในการควบคุมกำหนดรูปแบบของเว็บไซต์ ถูกนำมาใช้ในการใส่สี จัดเรียงหน้า และจัดรูปแบบหน้าให้ดูสวยงาม อีกทั้งใช้ทำส่วนของ Navigation bar ด้านบนของเว็บไซต์ ตลอดจนใช้ในการทำให้เว็บไซต์กลายเป็น Responsive Web

JavaScript และ JQuery



JQuery เป็น library หนึ่งของ JavaScript โดยถูกนำมาใช้หลักๆในสร้างส่วนของการ search ใน หน้า Search Tricks

อ้างอิง

1. CSS

<http://www.web-thai.com>

<http://www.enjoyday.net>

<http://www.xvlnw.com>

<https://noobmarketer.com/css-tutorial/>

2. Bootstrap

<https://v4-alpha.getbootstrap.com/>

<http://www.siamhtml.com/bootstrap-/>

<http://getbootstrap.com/>

3. html

<http://www.codingbasic.com/html.html>

<https://thongkred.wordpress.com/>

4. JavaScript

https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript

<https://www.mindphp.com/>

5. jQuery

<http://www.kontentblue.com>

<https://blog.sogoodweb.com/Article/Detail/52388>

<http://sci.udru.ac.th>