**Web Programming**

**Unit 3 : Front End Development**

**Editors, IDE**

⊗ Editor คือ software สำหรับการเขียนโปรแกรม

⊗ บางครั้งเรียกว่า IDE : Integrated Development Environment

⊗ IDE ทำหน้าที่ จัดสรรทรัพยากรที่เหมาะสมสำหรับการเขียนโปรแกรมในภาษาต่างๆ

⊗ สำหรับ IDE ที่เหมาะสมสำหรับเขียนภาษาท HTML และภาษาอื่นๆที่เกี่ยวข้องกับการพัฒนาเว็บไซต์ ได้แก่ Visual Studio Code, Eclipse, Notepad++, Netbeans, X code

⊗ IDE ของแต่ละบริษัทมีข้อดีข้อเสีบแตกต่างกัน นักศึกษาสามารถหาข้อมูลความแตกต่างได้จากการค้นคว้าด้วยตนเองบนเว็บไซต์ เนื่องจากแต่ละเวอร์ชั่นของแต่ละ IDE ก็จะมีความแตกต่างกันด้วย

⊗ ดังนั้นจึงสรุปได้ยากว่า IDE ตัวใดดีที่สุดยิ่งไปกว่านั้น IDE แต่ละยี่ห้อก็จะเหมาะสำหรับภาษาหนึ่งแต่อาจจะไม่เหมาะกับอีกภาษาหนึ่ง

⊗ ดังนั้นการเลือก IDE จึงมีปัจจัยหลายด้านเข้ามาพิจารณา เช่น

→ การจัดสรรและใช้ทรัพยากร <resource> ของเครื่องคอมพิวเตอร์มากน้อยเพียงใด

→ มีระบบการแก้ไขจุดบกพร่อง <debug> ที่เข้าใจง่ายและมีความละเอียดมากน้อยแค่ไหน

→ IDE บางตัวอาจเหมาะสำหรับ Java แต่ไม่เหมาะสำหรับ HTML แต่อีกตัวอาจจะเหมาะสำหรับ HTML แต่ไม่เหมาะสำหรับ Java

→ User Interface มีความซับซ้อนในการใช้งานมากน้องเพียงใด สามารถปรับสีของ font ได้หรือไม่

→ รองรับการเขียนโปรแกรมได้กี่ภาษา

→ ปัจจัยอื่นๆ เช่น ความถนัดขึ้นอยู่กับการใช้งานและความต้องการของผู้พัฒนา

**Visual Studio Code**

Text

Description automatically generated

**NetBeans IDE**

Text

Description automatically generated

**Web Development**

⊗ คือ การพัฒนาหรือสร้างเว็บไซต์โดยผ่านกระบวนการเขียนโปรแกรม <Web Programming> โดยพื้นฐานแล้วเราสามารถใช้ภาษา HTML ภาษาเดียวในการสร้างเว็บไซต์แต่ในปัจจุบันมีการใช้งานภาษาโปรแกรมหลายภาษารวมไปถึง framework ต่างๆเพื่อให้เว็บไซต์มีความสามารถมากขึ้นรองรับบริการต่างๆได้ โดยภาษาที่มีการใช้งานเพิ่มเติม เช่น CSS, JavaScript, PHP, SQL, Python เป็นต้น

⊗ การพัฒนาเว็บไซต์ในปัจจุบันจะแบ่งออกเป็น 2 ส่วน คือ Front-End Development,   
Back- End Development

⊗ Front-End Development <หน้าบ้าน> จะเป็นการพัฒนาเว็บไซต์ในส่วน User Interface <UI> หรือหน้าเว็บที่ผู้ใช้ทำการใช้งาน โดยมีหลักการออกแบบที่ต้องคำนึกถึงผู้ใช้เรื่องของ Web Design and UX/UI design โดยจะต้องพัฒนาเว็บไซต์ในส่วนนี้ให้มีความสวยงาม น่าใช้ ใช้งานง่ายและมีประสิทธิภาพ

⊗ Back- End Development <หลังบ้าน> คือ การพัฒนาเว็บไซต์ในส่วนของการจัดการ Server, Database รวมไปถึง Service ต่างๆที่เป็นส่วนหนึ่งของเว็บไซต์เพื่อให้การใช้งานของผู้ใช้นั้นเป็นไปอย่างราบรื่น

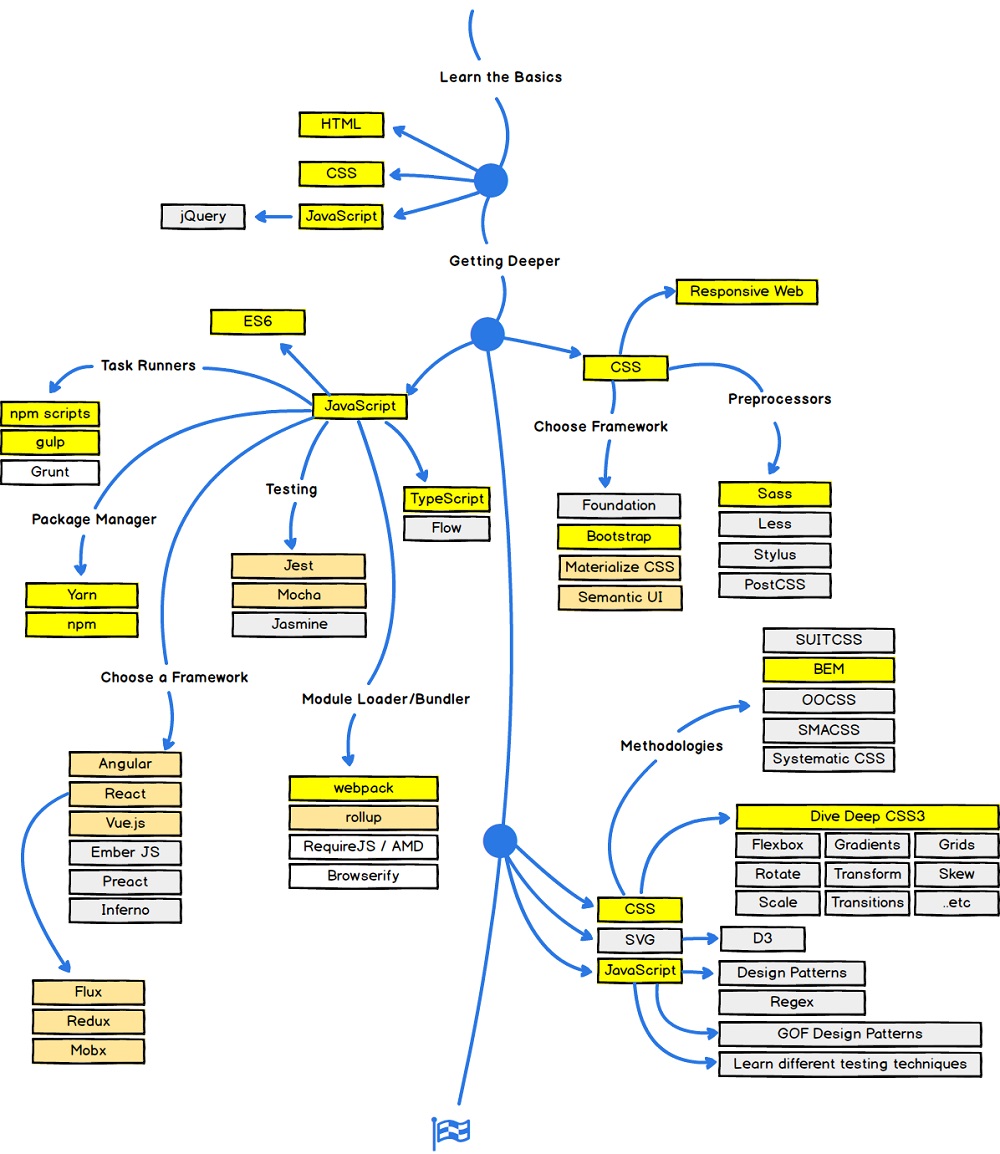
**Framework**

⊗ จะทำให้การเขียนโค้ดนั้นง่ายขึ้น เช่น หากเราใช้ framework ของภาษา JavaScript จะทำให้เราเขียน JavaScript ง่ายขึ้นเนื่องจากตัว framework มีการเขียนคำสั่งบางส่วนเอาไว้แล้ว เราเพียงแค่ต้อง import เข้ามาและยังสามารถปรับได้ตามความต้องการของเราแต่จะอยู่บนกฎระเบียบของ framework นั้นๆหรือพูดง่ายๆคือ framework จะเป็นโค้ดการทำงานที่มีคนเขียนไว้แล้ว เราเพียงแค่เอามาใช้ต่อและปรับแต่งให้เข้ากับความต้องการของเรา

⊗ ตัวอย่าง หากเราใช้ framework เราอาจเขียนโค้ดเพียง 3 บรรทัดแต่หากไม่ใช่ framework เราอาจต้องเขียนโค้ดถึง 6 บรรทัดและอาจต้องมีการใช้คำสั่งที่ซับซ้อนขึ้น

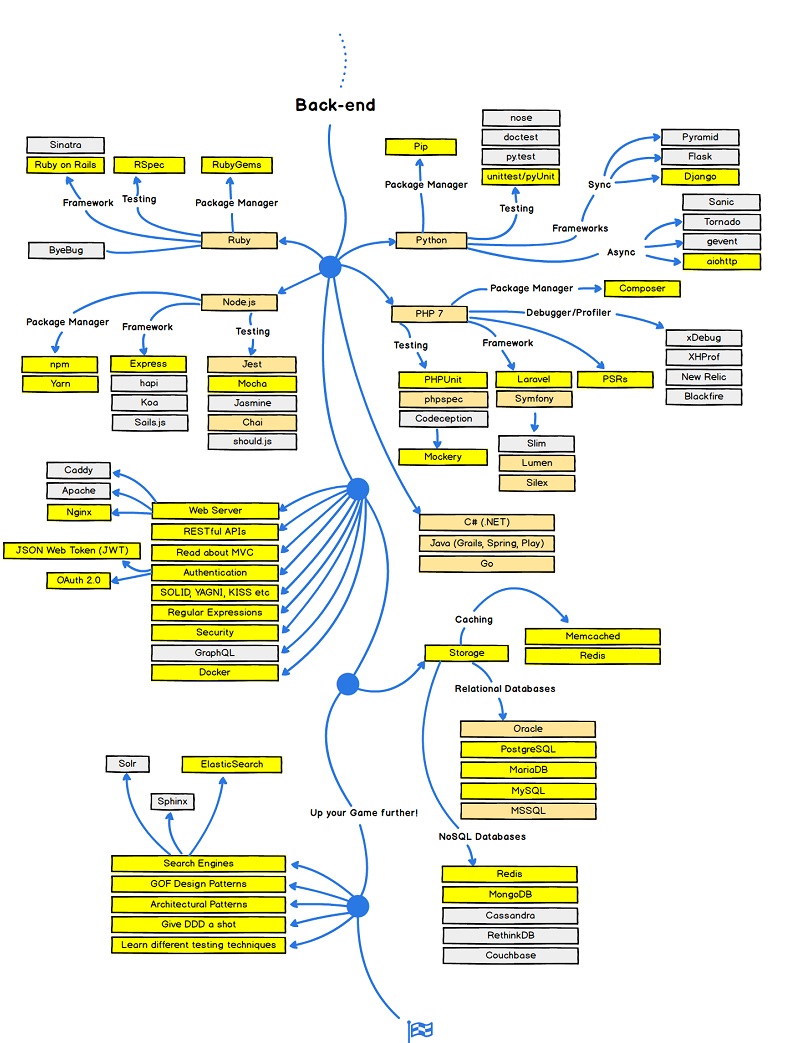
⊗ Front-End Framework

Framework และภาษาที่ใช้ในงานด้าน Front-End Development



⊗ Back-End Framework

Framework และภาษาที่ใช้ในงานด้าน Back-End Development



**Front-End Development**

⊗ สิ่งที่ต้องคำนึกถึงในการสร้างหน้าเว็บไซต์ คือ Web Design และ UX/UI design ซึ่งจะเป็นการคำนึกถึงผู้ใช้และสร้างประสบการณ์ที่ดีต่อการใช้งานเว็บไซต์ซึ่งในส่วนนี้ไม่จำเป็นที่จะต้องยุ่งเกี่ยวกับ coding เลย โดย coding จะอยู่ในขั้นตอนของการพัฒนา

⊗ ในวิชานี้จะสอนการใช้งานภาษา HTML, CSS, JavaScript รวมไปถึงการใช้งาน framework สำหรับ font-end development

**> HTML <**

⊗ Hypertext Markup Language เป็นภาษาสคริปต์ <Script Language> และเป็นภาษาหลักพื้นฐานของการสร้างเว็บเพจข้อมูลข่าสารที่ปรากฎบนหน้าจอผ่าน Web Brower ถูกกำหนดโดยภาษา HTML เป็นหลัก

⊗ โปรแกรมภาษา HTML จะมีไวยากรณ์ <Syntax> และรูปแบบการเขียนเป็นแบบแท็ก <tag> **ตัวอย่าง** <tagname>เนื้อหา…</tagname>

**HTML : ภาษาสคริปต์**

Application

Description automatically generated with medium confidence

⊗ จากตัวอย่างจะเห็นว่าข้อความที่เป็นผลลัพธ์ด้านขวาเกิดจากการกำหนดรูปแบบจาก html ที่เป็นการกำหนด tag ด้านซ้ายได้แก่ h1 และ p1 ที่ให้ผลลัพธ์แตกต่างกันระหว่าง 2 ข้อความ

**HTML Tag**

⊗ Tag ในภาษา HTML แบ่งออกเป็น 2 ประเภท คือ tag ที่มีการเปิดและปิด, tag ที่ไม่มีการปิด

**1. tag ที่มีการเปิดและปิด** ซึ่งข้อข้อมูลที่แสดงบนเว็บจะอยู่ระหว่างแท็กทั้งสอง

**ตัวอย่าง**

<p> แท็กเปิดสำหรับเริ่มย่อหน้าใหม่

</p> แท็กปิดสำหรับสิ้นสุดย่อหน้า

<p> ข้อความในย่อหน้าอยู่ตรงนี้ </p>

Graphical user interface, application, Word

Description automatically generated

⊗ จะสังเกตได้ว่าตัวอย่างข้อความในตัวอย่างด้านบนจะอยู่ในย่อหน้าเดียวกัน ผลลัพธ์ที่ได้จะอยู่ในบรรทัดเดียวกัน แต่ตัวอย่างด้านล่างข้อความทั้ง 2 ไม่ได้อยู่ใน tag เดียวกัน   
ดังนั้นผลลัพธ์ที่ได้จะแสดงออกมาว่าข้อความทั้งสองอยู่คนละย่อหน้า

Graphical user interface, application, Word

Description automatically generated

**2. tag ที่ไม่มีการปิด** ซึ่งข้อมูลที่แสดงบนเว็บจะอยู่ด้านหลังของ tag เช่น <br> คือ tag สำหรับขึ้นบรรทัดใหม่

**ตัวอย่าง**

Diagram, text

Description automatically generated**โครงสร้างพื้นฐานของภาษา HTML**

Graphical user interface, text, application

Description automatically generated

⊗ Web Brower ทำหน้าที่ ในการอ่านและแปลถอดความภาษา HTML จากนั้นจะแสดงผลออกมาเป็นหน้าเว็บเพจโค้ดด้านซ้ายจะแสดงผลดังรูปด้านขวา

Graphical user interface, text, application

Description automatically generated

**HTML Element**

⊗ แท็กที่กำหนดลักษณะของเนื้อหาต่างๆเหล่านี้เราเรียกว่า Element

⊗ เนื้อหาต่างๆของเว็บเพจจะอยู่ในส่วนของแท็ก <body>…</body> และเนื้อหาแยกย่อยจะอยู่ในแท็กต่างๆ เช่น หากเป็นหัวข้อจะใช้แท็ก Heading <h1>…</h1> ซึ่งมีระดับตั้งแต่ h1-h6 หากเป็นเนื้อหาที่เป็นบทความหรือย่อหน้าจะใช้แท็ก Paragraph <p>…</p> หากต้องการแทรกรูปภาพให้ใช้แท็ก Image <img> เป็นต้น

**.html**

⊗ ไฟล์ทุกไฟล์ที่เขียนด้วยภาษษา HTML จะมียามสกุล “.html” สามารถเปิดได้ด้วยโปรแกรม web browser ทุกชนิด เมื่อเขียนภาษา HTML เสร็จให้ทำการบันทึกเป็นไฟล์ .html หรือ .htm จึงจะสามารถนำไปใช้งานได้

Graphical user interface, text, application, email

Description automatically generated

**> CSS : Cascading Style Sheet<**

⊗ เป็นภาษาโปรแกรมที่ใช้ควบคู่กับภาษา HTML เพื่อเป็นการตกแต่งภาพลักษณ์ของเว็บเพจรวมไปถึงสี ตำแหน่ง การวางข้อความ ตาราง รูปภาพ สื่อต่างๆบนหน้าเว็บ

⊗ พูดง่าย คือ เป็นภาษาสำหรับตกแต่งเว็บไซต์ให้มีความสวยงามยิ่งขึ้น

⊗ ก่อนจะมี CSS จะใช้แต่ HTML เท่านั้นซึ่งจะมีความซับซ้อนและทำให้โค้ด HTML ดูยุ่งยากแต่ CSS ช่วยให้การทำงานแยกเป็นส่วนๆโดยแยกเป็นส่วนๆ โดยแยกส่วนเนื้อหากับส่วนตกแต่งออกจากกันเพิ่มความง่ายให้กับผู้พัฒนาอีกทั้งยังมีการจัดวางโค้ดได้อย่างเป็นระเบียบง่ายต่อการบริหารจัดการและการ debug

**การแทรก CSS**

⊗ ก่อนการเขียนภาษา CSS ต้องมีการระบุแท็กเปิดเพื่อบอก IDE ว่าเราต้องการจะพิมพ์ CSS โปรแกรมจะทำการปรับไวยากรณ์ให้เป็นไปตามชนิดของแท็กที่เปิดไว้ ภาษา CSS มีแท็กเปิดและปิด คือ <style>…</style>

<style>

…

…

</style>

**CSS : Syntax**

Graphical user interface, text, application, email

Description automatically generated

**CSS : Selector**

⊗ สามารถระบุได้ 3 รูปแบบ

1. ระบุชื่อแท็ก เช่น p, h1, img โดย style จะมีผลในทุกๆแท็กที่มีชื่อนั้นๆ

2. ระบุ id ของแท็กโดย style จะมีผลเฉพาะแท็กที่มี id ตรงกับที่ระบุ

3. ระบุ class ของแท็กโดย style จะมีผลเฉพาะแท็กที่มี class ตรงกับที่ระบุ

⊗ นอกจากนี้ยังมีการระบุแบบซับซ้อนที่สามารถแท็กย่อยได้ เช่น การกำหนดสไตล์ให้กับ <p> บางแท็กใน <div>

**CSS : id Selector**

⊗ สามารถระบุ id ได้ด้วยการใส่เครื่องหมาย # ตามด้วยชื่อ id เช่น #p1 (…) คือ ระบุแท็กที่มี id คือ “p1” ดังรูปจากรูปจะเห็นว่า <p> ที่มี id คือ p1 เท่านั้นที่จะได้รับผลของ style

A picture containing chart

Description automatically generated

**CSS : การแทรกโค้ด CSS**

⊗ สามารถเขียนได้ 3 แบบ

1. Inline Style เขียนโค้ดลงไปในบรรทัดที่ต้องการบนเอกสาร html เพื่อปรับเปลี่ยนสไตล์ของข้อมูลนั้นโดยเฉพาะ

2. Internal Style เขียนโค้ดลงบนส่วน <head> ของเอกสาร html เพื่อกำหนดสไตล์ให้กับ element ที่มีชื่อเดียวกัน เช่น หากต้องการให้ <p> ทั้งหมดเป็นสีแดงและเป็นตัวหนาสามารถฝังโค้ด CSS เป็นบนส่วน <head> และกำหนด selector เป็น <p> และกำหนดสไตล์ตามต้องการจากนั้นข้อความที่เป็น <p> ทั้งหมดจะมีสไตล์ตามที่กำหนด

3. External Style การสร้างไฟล์ .css แยกออกมาเพื่อกำหนดคุณลักษณะให้กับเอกสาร html หลายไฟล์ เช่น หากต้องการให้เพจแต่ละเพจมีสไตล์ที่เหมือนกันสามารถใช้การเขียนรูปแบบนี้ได้และในแต่ละเพจต้องมีการอ้างอิงถึงไฟล์ .css ที่เขียนไว้ด้านนอก

**CSS : Internal Style**

Text

Description automatically generated

⊗ เขียนโค้ดลงบนส่วน <head> ของไฟล์ html เพื่อกำหนดคุณสมบัติให้กับ element ที่เลือกทั้งหมดบนเพจนี้

⊗ จากรูปจะเห็นว่าพื้นหลังของหน้าเว็บทั้งหมด (body) จะกำหนดสีเป็น linen และข้อความทั้งหมดที่เป็น heading 1 (h1) จะมีสีเป็น maroon และมีระยะห่างจากของทางด้านซ้ายเป็น 40

**CSS : Comment**

⊗ ระบุ comment ด้วยเครื่องหมาย /\*…\*/ โดย /\* คือเปิด \*/ คือปิด

Chart

Description automatically generated with low confidence

**CSS : Properties**

ในภาษา CSS นั้นโดยหลักๆจะมีการใช้งาน property ที่หลากหลายซึ่ง property ที่นักศึกษาควรฝึกและทดลองใช้งานดังนี้

⊗ ตระกูล background เป็นการกำหนดพื้นหลัง

⊗ height, width เป็นการกำหนดขนาดของวัตถุ เช่น รูปภาพ

⊗ ตระกูล font เป็นการกำหนดคุณสมบัติให้กับตัวอักษร

⊗ margin, padding เป็นการกำหนดระยะขอบและการเยื้อง (ย่อหน้า)

⊗ border เป็นการกำหนดเส้นขอบ

⊗ ตระกูล dropdown คือ การทำเมนูแบบ dropdown list

⊗ form คือการสร้างแบบฟอร์มสำหรับกรอกข้อมูล

ความเชี่ยวชาญขึ้นอยู่กับการฝึกฝนและความถี่ในการใช้งาน

**> JavaScript <**

⊗ ภาษา JavaScript เป็นสคริปต์แทรกอยู่ในไฟล์ html และปะปนอยุ่กับโค้ด html

⊗ JavaScript เป็นหนึ่งในภาษาโปรแกรมที่ได้รับความนิยมสูงที่สุดในโลก

⊗ JavaScript เป็นหนึ่งในภาษาหลักที่นำมาพัฒนาเว็บไซต์ในส่วนของ front end ช่วยให้เว็บเพจสามารถโต้ตอบ (interactive) กับผู้ใช้ได้

⊗ JavaScript ยังเป็นภาษาโปรแกรมสำหรับการพัฒนาโมบายแอพแบบ cross platform ที่ได้รับนิยมสูงที่สุดในหลายๆบริษัท เช่น React Native, Ionic, PhoneGap

⊗ JavaScript เป็นภาษาโปรแกรมที่เรียนรู้ได้ง่าย

**JavaScript คืออะไร**

⊗ หากต้องการที่จะเป็น web front end developer จะต้องทราบ 3 ภาษาโปรแกรมดังนี้

1. HTML คือ ภาษาที่กำหนดโครงสร้างของเว็บ การวางเนื้อหา ตำแหน่งของเนื้อหา

2. CSS คือ ภาษาที่กำหนดรูปลักษณ์ สีสันต่างๆของเว็บ

3. JavaScript คือ ภาษาที่กำหนดว่าเว็บสามารถทำอะไรได้บ้างสามารถโต้ตอบกับผู้ใช้อย่างไรได้บ้าง

⊗ JavaScript เป็นภาษาโปรแกรมที่มีลักษณะคล้ายกับภาษา C , Java, Python หรือภาษาอื่นๆที่มีการประกาศตัวแปรมีการกำหนดเงื่อนไข <if-else> มีการใช้งานลูป <for, while loop> หรือมีการกำหนดฟังก์ชันในขณะที่ภาษา HTML, CSS ไม่มีการใช้งานในส่วนนี้

⊗ ดังนั้นจึงมีความซับซ้อนมากกว่าภาษา HTML, CSS

⊗ ภาษา JavaScript สามารถใส่ไว้ในส่วน <head> หรือ <body> ก็ได้โดยต้องกำหนดแท็กสำหรับแทรกภาษา JavaScript คือ <script> … </script>

⊗ จากโค้ดจะเห็นว่ามีการแทรกโค้ด JavaScript ไว้ในส่วนของ <head>

Graphical user interface, text, application

Description automatically generated

⊗ การแทรก JavaScript สามารถกระทำใน element ของ html ได้ดังตังอย่างด้านล่างจะเป็นการแทรกโค้ด JavaScript ไว้ในส่วนของ <button>

Graphical user interface, text, application, email

Description automatically generated

⊗ JavaScript มีความสามารถทำให้เว็บไซต์สามารถตอบโต้หรือมีการรับค่าคำนวณและแสดงผลได้ตามที่ต้องการ โดยหลักๆแล้ว JavaScript จะทำการเปลี่ยนแปลงเนื้อหาของ html ที่มีอยู่ให้เปลี่ยนไปตามที่ผู้ใช้ต้องการ เช่น เปลี่ยนเนื้อหาของ html เปลี่ยนสไตล์ของ CSS ซ่อนหรือแสดง html element

**JavaScript : getElemenyById**

⊗ เรียกวิธีการต่างๆของ JavaScript ว่า method โดยหนึ่งใน method ที่มีการใช้งานกันมากที่สุด คือ getElementById <get element by id> เนื่องจากเป็นการเลือก id ของ html element เพื่อที่จะทำการใส่ action ต่างๆให้กับ element นั้นๆผ่าน id จากโค้ดเป็นการเลือก element ที่มี id ว่า demo ที่อยู่ในเพจนี้ (.innerHTML)

Text

Description automatically generated

⊗ ในส่วน <head> นั้นได้มีการประกาศฟังก์ชันโดยพิมพ์ “function” ตามด้วยชื่อฟังก์ชัน

⊗ ในที่นี้ คือ myFunction( ) จากนั้นตามด้วย { } ซึ่งโค้ดที่อยู่ใน { } จะทำงานก็ต่อเมื่อฟังก์ชันนี้ถูกเรียกใช้

Graphical user interface, text, application, email

Description automatically generated

⊗ ในส่วน <body> จะมีการอ้างถึงชื่อฟังก์ชันที่มีประกาศไว้ในส่วน <head> หมายถึง เมื่อทำการกดปุ่ม “Try it” จะส่งผลให้โค้ดใน { } ของฟังก์ชัน myFunction ในส่วน <head> ทำงาน

Graphical user interface, text, application, email

Description automatically generated

⊗ หรือจะเขียนโค้ดทั้งหมดให้อยู่ในส่วนของ <body> ก็ได้

Graphical user interface, text, application, email

Description automatically generated

**JavaScript : External Files**

⊗ สามารถเขียน JavaScript เป็นไฟล์อื่นแยกได้เช่นนเดียวกัน CSS และทำการใส่ลิงก์เชื่อมโยงไปยังไฟล์ดังกล่าว

⊗ กรณีที่ต้องการใช้โค้ด JavaScript เดียวกันกับหลายๆเว็บเพจโดยกำหนดนามสกุลไฟล์เป็น .js

Graphical user interface, text, application

Description automatically generated

**JavaScript : ตัวแปร**

⊗ JavaScript เป็นภาษาโปรแกรมที่คล้ายภาษาอื่นๆ เช่น Java, C ที่จะมีประกาศตัวแปรและนำตัวแปรไปใช้

⊗ เป็นการนำค่าจากตัวแปร “lastName” มาแสดง

Graphical user interface, text, application

Description automatically generated with medium confidence

⊗ จากโค้ดด้านซ้ายจะเป็นกำหนดตัวแปร x, y, z และกำหนดค่า คือ 5, 6 และตัวแปร z มีค่าคือ “x+y” จากนั้น element ที่มี id demo จะทำการแสดงตัวแปร “z” ซึ่งจะได้ผลลัพธ์ คือ x+y = 5+6 เท่ากับ 11 ดังรูปด้านขวา

Graphical user interface, text, application

Description automatically generated

**> Bootstrap <**

⊗ เป็น framework ภาษา CSS ที่ได้รับความนิยมสูงในเรื่องของการสร้างเว็บที่มีความยืดหยุ่น ตอบสนองและสามารถเข้าถึงได้จากอุปกรณ์แต่ละประเภท เช่น คอมพิวเตอร์หรืออุปกรณ์พกพาโดยเนื้อหาไม่ผิดเพี้ยน มีคำสั่งช่วยในการจัด layout ของหน้าเว็บเพจที่ง่ายและสวยงาม

⊗ Bootstrap ใช้ภาษา HTML, CSS, JavaScript เป็นหลักโดยรองรับ browsers ที่มีการใช้งานทั่วไป ได้แก chrome, Firefox, MS, Edge, Safari, Opera

Graphical user interface, text

Description automatically generated

**Bootstrap 5**

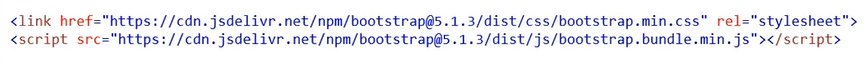
⊗ ปัจจุบันมีการอัพเดทมาจนถึงเวอร์ชั่น 5 โดยความแตกต่างหลักจะอยู่ที่ Bootstrap 5 นั้นใช้ JavaScript แทนการใช้ JQuery และไม่รองรับ Internet Explorer เวอร์ชั่น 11 แบะก่อนหน้า

Graphical user interface, application

Description automatically generated

**Bootstrap : library**

⊗ การใช้งาน Bootstrap ในส่วนของ library ด้วยโดยทำการใส่โค้ดไว้ในส่วน <head>



**Bootstrap : Container**

⊗ การเขียน Bootstrap นั้นต้องมีการกำหนดชนิดของความกว้างของหน้าจอซึ่งจะถูกกำหนดโดบ container มี 2 ประเภท คือ

1. .container การระบุความกว้างตามที่กำหนด

2. .container-fluid การระบุความกว้างตามขนาดของหน้าจออุปกรณ์นั้นๆ

**Bootstrap : Grids**

⊗ จะแบ่งหน้าจอออกเป็นส่วนต่างๆเรียกว่า grid ซึ่งสามารถแบ่งได้สูงสุด 12 column ต่อ 1 เพจและสามารถรวม column ได้ในกรณีต้องการ column ที่กว้างขึ้น

Table

Description automatically generated

**Bootstrap : Grids**

⊗ การกำหนด grid นั้นสามารถระบุขนาดของแต่ละ column ไม่ว่าเนื้อหาในนั้นจะมีขนาดใหญ่เล็กแค่ไหนตามชนิดของอุปกรณ์เพื่อความสวยงามและง่ายต่อการอ่าน

1. xs : for phones

2. sm : for tables

3. md : for desktops

4. lg : for larger desktops

5. xl : extra large

6. xxl : extra extra large

⊗ หากกำหนด sm จะเป็นการบอกว่าหน้าจอจะแสดงผลตามที่ระบุตั้งแต่ tablet และใหญ่ขึ้นไปส่วนอุปกรณ์ที่เล็กกว่า เช่น phones จะมีการแสดงผลอีกรูปแบบหนึ่ง

**Bootstrap : Basic Grids Structure**

Graphical user interface, application

Description automatically generated

Text, letter

Description automatically generated

⊗ จากโค้ดจะเป็นการกำนหด 3 column อยู่ในแถว row การแสดงผลจะเป็นการแสดงผลแชช 3 column ที่อยู่ในแถวเดียวกันในอุปกรณ์ตั้งแต่ tablets และอุปกรณ์ที่มีขนาดใหญ่ขึ้นไป เช่น PCs ส่วนโทรศัพท์มือถือจะแสดงเป็นแบบ 3 ชั้น

⊗ เนื่องจาก -sm- คือการกำหนดอุปกรณ์ขนาดเริ่มต้นที่จะแสดงผลแบบ 3 column ในแถวเดียวกันและอุปกรณ์ที่มีขนาดใหญ่ขึ้นไป ส่วนอุปกรณ์ที่เล็กกว่า sm จะมีการแสดงผลเรียงเป็นชั้น

Text

Description automatically generated

⊗ จากโค้ดและรูปจะเห็นได้ว่ามีการกำหนด -sm-4 และ -sm-8 หมายความว่ามีการกำหนด 2 column ซึ่ง column ทางซ้ายมีขนาด 4 และ ทางขวามีขนาด 8 รวมกันเท่ากับ 12 span



**Bootstrap : Navbar example**

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

⊗ โค้กการสร้างแถบเมนูมีการกำหนดชื่อเว็บไซต์ใน navbar-header ซึ่งอยู่ใน <a> มี class คือ navbar-brand จากนั้นกำหนดลิสต์ <li> ซึ่งมีด้วยกัน 4 เมนู คือ Home, Page1, Page2, Page3 ซึ่ง class = “active” ในเมนู Home แสดงถึงการที่เรากำลังอยู่ในหน้าเว็บนั้นๆ