

บทที่ 2

แนวคิด ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

ในบทนี้เป็นการกล่าวถึงแนวคิด ทฤษฎี โครงงานและแอปพลิเคชันต่างๆ ที่เกี่ยวข้องกับระบบนำเสนองาน เพื่อช่วยแนะนำผู้ใช้งานในด้านการค้นหาข้อมูล หรือค้นหาสิ่งที่มีความใกล้เคียงกันมากที่สุด โดยมีข้อหลักที่เกี่ยวข้องทั้งหมด ดังนี้

2.1 การเรียนรู้ของเครื่อง (Machine Learning)

2.2 การแยกตัวประกอบเมตริกซ์ สำหรับระบบแนะนำ (Matrix Factorization For Recommender System)

2.3 โปรแกรม และเทคโนโลยีที่นำมาใช้

2.1 การเรียนรู้ของเครื่อง

2.1.1 ความหมายของการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง มีการพัฒนาเริ่มมาจากวิชาคณิตศาสตร์และสถิติ แต่ในระยะหลังในยุคของ Big Data ที่มีข้อมูลเป็นจำนวนมาก มีความหลากหลายในโครงสร้างและเป็นข้อมูลที่หลังไหลเข้ามาอย่างรวดเร็ว ทำให้ศาสตร์และเทคโนโลยีของการวิเคราะห์ข้อมูลถูกพัฒนาให้ดีขึ้น หลังจากปี 2000 การเรียนรู้ของเครื่อง เริ่มเป็นที่นิยมและมีคำจำกัดความที่ชัดเจนยิ่งขึ้น

การเรียนรู้ของเครื่อง เป็นศาสตร์แขนงหนึ่งของวิทยาศาสตร์คอมพิวเตอร์ (Computer Science) ซึ่งมุ่งเน้นที่จะสร้างองค์ความรู้จากข้อมูล โดยเริ่มจากการสร้างโมเดลทางคณิตศาสตร์ที่มีความยืดหยุ่น และสามารถปรับตัวเองให้เข้ากับข้อมูลที่ได้รับเพื่อที่จะสามารถทำนาย หรือคาดเดาเหตุการณ์ในอนาคตได้

2.1.2 ประเภทของการเรียนรู้ของเครื่อง

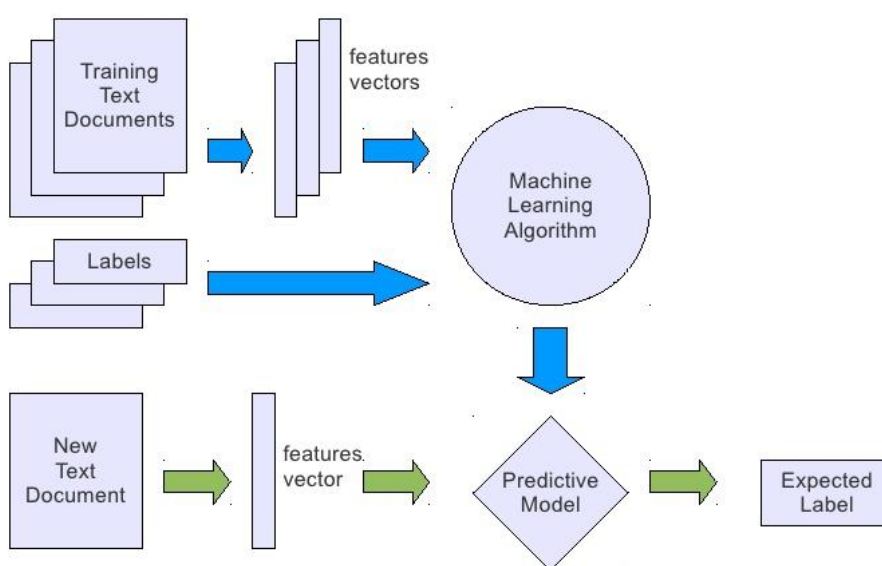
ประเภทของการเรียนรู้ของเครื่อง สามารถจำแนกตามหน้าที่และขอบเขตปัญหาได้หลักๆ คือ การเรียนรู้แบบมีผู้สอน (Supervised Learning), การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) และ การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

2.1.2.1 การเรียนรู้แบบมีผู้สอน

การเรียนรู้แบบมีผู้สอน เป็นเทคนิคหนึ่งของการเรียนรู้ของเครื่อง ซึ่งสร้างฟังก์ชันจากข้อมูลสอน (Training data) ข้อมูลสอนประกอบด้วยอินพุต และผลที่ต้องการ ซึ่งผลจากการเรียนรู้จะเป็นฟังก์ชันอาจจะให้ค่าต่อเนื่อง (เรียกว่าวิธีการถดถอย – Regression) หรือใช้ทำนายประเภทของวัตถุ (เรียกว่า การแบ่งประเภท – Classification)

หน้าที่หลักของการเรียนรู้ของเครื่องแบบมีผู้สอน คือการทำนายค่าฟังก์ชันจากอินพุตที่ถูกต้องโดยใช้ตัวอย่างสอนจำนวนน้อย

แบบจำลองการเรียนรู้แบบมีผู้สอน (Supervised Learning Model)



ภาพที่ 2.1 ภาพโครงสร้างลำดับการทำงานของระบบเรียนรู้แบบมีผู้สอน

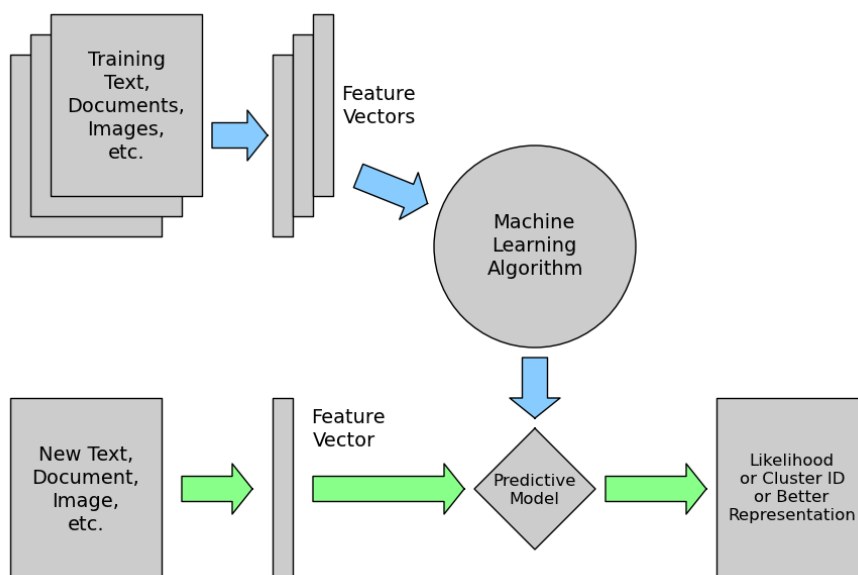
นำข้อมูลสอนที่ต้องการให้เครื่องเรียนรู้ ได้แก่ อินพุตที่ได้จากการแปลงเอกสาร ข้อมูลสอนออกเป็นชุดๆ เพื่อระบุคุณสมบัติของเวกเตอร์ และเอาต์พุตที่ต้องการ เข้าไปแทนในส่วน โครงสร้างของการเรียนรู้ด้วยเครื่อง (Machine Learning Algorithm) เพื่อให้เครื่องได้เรียนรู้ และ วิเคราะห์ความสัมพันธ์ของอินพุตและเอาต์พุต จากนั้นจะเข้าสู่กระบวนการสร้างแบบจำลองทางสถิติของ คุณสมบัติในชุดข้อมูลเทรนนิ่ง และครั้งต่อไปเมื่อมีข้อมูลในลักษณะเดียวกับข้อมูลสอน เครื่องก็จะนำ ข้อมูลดังกล่าวไปทำการพยากรณ์โดยใช้แบบจำลองการพยากรณ์เพื่อหาค่าคุณสมบัติของชุดข้อมูลใหม่

2.1.2.2 การเรียนรู้แบบไม่มีผู้สอน

การเรียนรู้แบบไม่มีผู้สอน เป็นเทคนิคหนึ่งของการเรียนรู้ของเครื่อง โดยการสร้างโมเดลที่เหมาะสมกับข้อมูล การเรียนรู้แบบนี้แตกต่างจากการเรียนรู้แบบมีผู้สอน คือ จะไม่มีการระบุผลที่ต้องการ หรือประเภทไว้ก่อน การเรียนรู้แบบนี้จะพิจารณาวัตถุเป็นเซตของตัวแปรสุ่ม แล้วจึงสร้างแบบจำลองความหนาแน่นร่วมของชุดข้อมูล

การเรียนรู้แบบไม่มีผู้สอนสามารถนำไปใช้ร่วมกับการอนุมานแบบเบย์ (Naïve Bayesian Classifier) เพื่อหาความน่าจะเป็นแบบมีเงื่อนไขของตัวแปรสุ่ม โดยกำหนดตัวแปรที่เกี่ยวข้องให้ นอกจากนี้ยังสามารถนำไปใช้ในการบีบอัดข้อมูล ซึ่งโดยพื้นฐานแล้วขั้นตอนวิธีการบีบอัดข้อมูลจะขึ้นอยู่กับ การแจกแจงความน่าจะเป็นของข้อมูลไม่อย่างชัดเจนก็โดยปริยาย

แบบจำลองการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning Model)

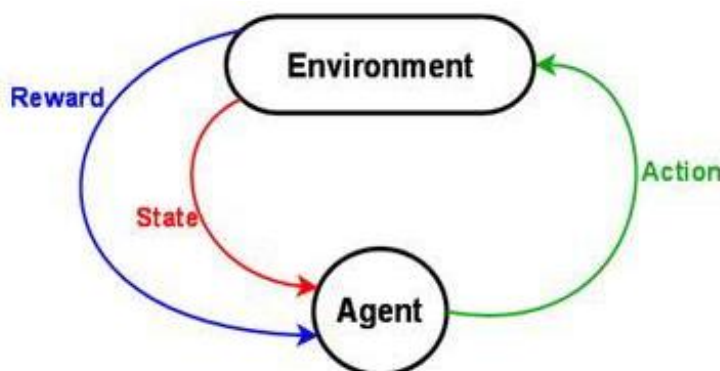


ภาพที่ 2.2 ภาพโครงสร้างลำดับการทำงานของระบบเรียนรู้แบบไม่มีผู้สอน

2.1.2.3 การเรียนรู้แบบเสริมกำลัง

การเรียนรู้แบบเสริมกำลัง คือ การเรียนรู้จากสิ่งแวดล้อมรอบตัว เรียนรู้จากธรรมชาติรอบตัวที่มีอยู่ในชีวิตจริง นำมาดัดแปลงให้กับคอมพิวเตอร์ ตัวอย่างเช่น การเล่นเกมกรุก จะต้องมีการทำนายล่วงหน้าว่าจะสามารถเกิดอะไรขึ้นได้ ซึ่งการเดินแต่ละครั้งอาจไม่เป็นผลดีต่อครั้งนั้นแต่อาจมีผลดีในครั้งต่อไปก็ได้ หรือเกม OX

แบบจำลองการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning Model)



ภาพที่ 2.3 ภาพโครงสร้างลำดับการทำงานของระบบเรียนรู้แบบเสริมกำลัง

ในการเรียนรู้แบบเสริมกำลังจะมีตัวกระทำที่เป็นผู้เรียน และมีอำนาจในการตัดสินใจ ซึ่งจะมีปฏิสัมพันธ์กับสภาพแวดล้อม โดยตัวกระทำนั้นจะเลือกการกระทำ และจากนั้นสภาพแวดล้อมจึงตอบสนองต่อการกระทำนั้นๆ โดยจะสร้างเหตุการณ์ใหม่ให้กับตัวแทน แล้วส่งคืนการตอบแทน

2.2 การแยกตัวประกอบเมตริกซ์ สำหรับระบบแนะนำ

2.2.1 เมตริกซ์

ในคณิตศาสตร์ เมตริกซ์ คือกลุ่มของเลขจำนวนจริง (real numbers) หรือกลุ่มของพารามิเตอร์ หรือกลุ่มของตัวแปรที่เขียนเรียงกันเป็นรูปสี่เหลี่ยมมุมฉากภายในเครื่องหมายวงเล็บ โดยปกติจะใช้วงเล็บ [] หรือวงเล็บ () และใช้อักษรตัวหนาแทนเมตริกซ์ ตัวอย่างเช่น

ขนาดของเมตริกซ์ A เท่ากับ 2x2 ฉะนั้นแบบทั่วไปของเมตริกซ์จึงสามารถเขียนได้ ดังนี้

$$A = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}$$

สมาชิกที่เขียนเรียงกันในแนวนอนเรียกว่าแถว (row) ส่วนสมาชิกที่เขียนในแนวตั้งเรียกว่าสดมภ์ (column) ดังนั้นจำนวนแถวและจำนวนสดมภ์ของเมตริกซ์หนึ่ง จะบ่งบอกถึงขนาด (dimension) ของเมตริกซ์นั้นๆ โดยเอาจำนวนแถวคูณจำนวนสดมภ์ ยกตัวอย่างเช่น

ขนาดของเมตริกซ์ A เท่ากับ 2×2 ฉะนั้นแบบทั่วไปของเมตริกซ์จึงสามารถเขียนได้ ดังนี้

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

หรือ $A = [a_{ij}]$ โดยที่ $i = 1, 2, \dots, m$

$j = 1, 2, \dots, m$

2.2.2 การแยกตัวประกอบของเมตริกซ์

การแยกตัวประกอบของเมตริกซ์ เรียกว่า การแยกเมตริกซ์ (matrix decomposition) การแยกตัวประกอบของเมตริกซ์ เพื่อให้เป็นสองเมตริกซ์หรือมากกว่านั้น และเมื่อคูณตัวประกอบเหล่านั้นเข้าด้วยกันจะได้ผลลัพธ์เดิม ซึ่งมีวิธีการที่เหมาะสมแตกต่างกันไปสำหรับเมตริกซ์นั้นๆ เพื่อค้นหาคุณลักษณะที่แฝงอยู่ในความสัมพันธ์ระหว่างสองเอนทิตีหรือมากกว่านั้น

2.2.2.1 LU Decomposition

LU Decomposition คือ การเปลี่ยนรูปของเมตริกซ์ A ให้กลายเป็น 2 Matrixes เขียนอยู่ในรูป $A = LU$ เมื่อ L คือ Lower Triangular matrix และ U คือ Upper Triangular Matrix หลักการของระเบียบวิธีนี้คือ ต้องการแบ่งเมตริกซ์ A ออกเป็นผลคูณของเมตริกซ์ LU

ขั้นตอนการคำนวณ LU Decomposition

แถวแรกของ U, $U_{1,j}$ เมื่อ $j = 1$ to N หาได้จาก

$$u_{1,j} = a_{1,j} \quad , j = 1 \text{ to } N$$

คอลัมน์แรกของ L, $l_{i,1}$ เมื่อ $i = 2$ to N

$$l_{i,1} = \frac{a_{i,1}}{u_{1,1}} \quad , i = 2 \text{ to } N$$

แถวที่ 2 ของ U

$$u_{2,j} = a_{2,j} - l_{2,1} u_{1,j} \quad , i = 2 \text{ to } N$$

คอลัมน์ที่ 2 ของ L

$$l_{i,2} = \frac{a_{i,2} - l_{i-1} u_{1,2}}{u_{2,2}}, j = 3 \text{ to } N$$

แถวที่ N ของ U

$$u_{n,j} = a_{n,j} - \sum_{k=1}^{n-1} l_{n,k} u_{k,j}, j = n \text{ to } N$$

คอลัมน์ที่ N ของ L

$$l_{i,n} = \frac{a_{i,n} - \sum_{k=1}^{n-1} l_{i,k} u_{k,n}}{u_{n,n}}, j = n + 1 \text{ to } N$$

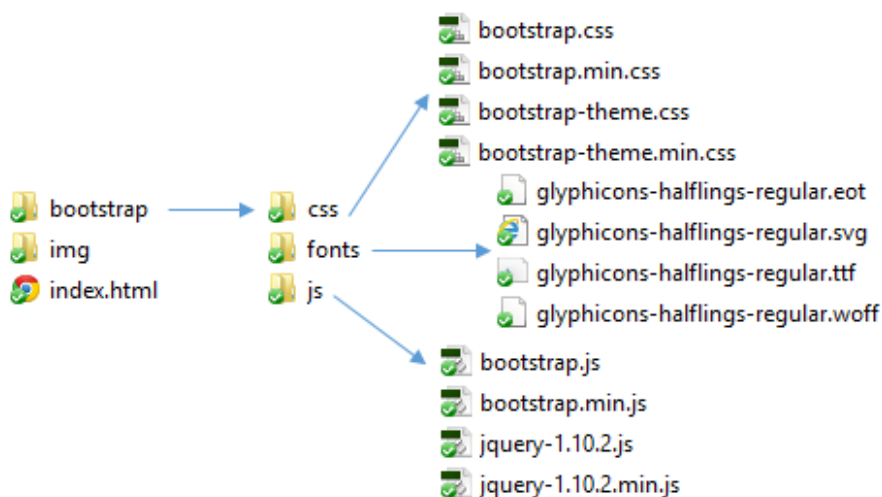
2.3 โปรแกรม และเทคโนโลยีที่นำมาใช้

2.3.1 Bootstrap 3

Bootstrap 3 คือ Frontend Framework ที่รวมเอชทีเอ็มแอล (HTML), ซีเอสเอส (CSS) และ เจเอส (JS) เข้าด้วยกันเพื่อพัฒนาเว็บที่สามารถรองรับทุกอุปกรณ์สมาร์ต หรือ เรียกว่า การออกแบบเว็บไซต์ให้สามารถรองรับการแสดงผลได้กับทุกอุปกรณ์ (Responsive Web) หรือ Mobile First โดยการ ใช้ Bootstrap จะช่วยให้การพัฒนาเว็บไซต์ของเราเร็วขึ้น ง่ายขึ้น และเป็นระบบมากยิ่งขึ้น

ในปัจจุบันมีการพัฒนาขึ้นด้วยกลุ่มนักพัฒนาจากทั่วทุกหนแห่งในโลก มีการอัปเดตแก้ไขข้อผิดพลาดอยู่ตลอดเวลา เพื่อรองรับการทำงานได้อย่างมีประสิทธิภาพและไม่มีปัญหาพร้อม ๆ กับการพัฒนาเทคโนโลยีบนเว็บเบราว์เซอร์ เช่น ซีเอสเอส, เอชทีเอ็มแอล และในปัจจุบันเราจะเห็นว่า Bootstrap มีตัวอย่างให้ดาวน์โหลดมากมาย มีทั้งที่สามารถดาวน์โหลดใช้ได้ฟรี และเสียเงินซื้อ และหลาย ๆ เว็บก็เลือกที่จะใช้ Bootstrap กันมากยิ่งขึ้น

2.3.1.1 โครงสร้างของ Bootstrap Framework



ภาพที่ 2.4 ภาพโครงสร้างพื้นฐานของ Bootstrap Framework

ตัวอย่างการเรียกใช้งาน Bootstrap พื้นฐานทั่วไป

โปรแกรมที่ 2.1 การเรียกใช้ Bootstrap พื้นฐาน

```

1.<head>
2.    <meta charset="utf-8" />
3.    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
4.    <meta name="viewport" content="width=device-width,
5.    initial-scale=1"/>
6.    <title>Bootstrap 101 Template</title>
7.
8.    <!-- Bootstrap -->
9.    <link href="css/bootstrap.min.css" rel="stylesheet">
10.   <script src="js/bootstrap.min.js"></script>
11.</head>
  
```

- กำหนดแท็กเมตาในบรรทัดที่ 3 เพื่อป้องกันอินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet Explorer) แสดงผลแบบ compatible mode โดยการบอกเบราว์เซอร์อินเทอร์เน็ตเอกซ์พลอเรอร์ให้ใช้ engine เวอร์ชันล่าสุดในการแสดงผลเว็บเพจ
- กำหนดแท็กเมตาในบรรทัดที่ 4 เพื่อแจ้งให้เว็บเบราว์เซอร์ทราบว่าเว็บไซต์นี้ถูกออกแบบให้แสดงผลได้อย่างเหมาะสมกับขนาดของอุปกรณ์ต่าง ๆ

- กำหนดแท็กลิงค์ในบรรทัดที่ 8 และสคริปต์ในบรรทัดที่ 9 เพื่อเรียกใช้ซีเอส

เอสของบูทสเตรป

ตัวอย่างการใช้ Bootstrap ในรูปแบบง่ายๆ

ตัวอย่างการใช้แท็ก <button> โดยเรียกใช้งานคลาสของบูทสเตรปเพื่อกำหนดสไตล์ของปุ่ม

โปรแกรมที่ 2.2 การใช้งานแท็ก <button>

```
<body>
  <button type="button" class="btn">Basic</button>
  <button type="button" class="btn btn-
default">Default</button>
  <button type="button" class="btn btn-
primary">Primary</button>
  <button type="button" class="btn btn-
success">Success</button>
  <button type="button" class="btn btn-
info">Info</button>
  <button type="button" class="btn btn-
warning">Warning</button>
  <button type="button" class="btn btn-
danger">Danger</button>
  <button type="button" class="btn btn-
link">Link</button>
</body>
```

ผลที่ได้จากการประมวลผลโปรแกรมที่ 2.2 จะแสดงผลดังนี้



ภาพที่ 2.5 ผลจากการประมวลผลโปรแกรมที่ 2.2

ตัวอย่างการใช้แท็ก <table> โดยเรียกใช้คลาสของบุทแสดรปเพื่อกำหนดสไตล์ของตาราง

โปรแกรมที่ 2.3 การใช้งานแท็ก <table>

```
<table class="table">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Username</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

โปรแกรมที่ 2.4 การใช้งาน <div class="form-group">

```
<form>
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

ผลที่ได้จากการประมวลผลโปรแกรมที่ 2.4 จะแสดงผลดังนี้



The image shows a web form for login. It has two input fields: 'Email address:' and 'Password:'. Below the password field is a checkbox labeled 'Remember me'. At the bottom left of the form is a button labeled 'Submit'.

ภาพที่ 2.7 ผลจากการประมวลผลโปรแกรมที่ 2.4

2.3.2 แองกูลาร์เจเอส (Angular JS 1.5)

แองกูลาร์เจเอส คือ เฟรมเวิร์คจาวาสคริปต์ ตัวหนึ่งที่พัฒนาโดย Google แองกูลาร์เจเอส นั้นจะมีลักษณะพิเศษในการเลือกใช้งานโดยจะต้องเพิ่มแอททริบิวต์พิเศษให้กับเอชทีเอ็มแอลเพื่อใช้งาน เหมาะสำหรับการพัฒนาเว็บโดยใช้เอชทีเอ็มแอลเพียงหน้าเดียว (Single Page Application) และสามารถพัฒนาเว็บแอปพลิเคชันส่วนหน้า (Front-End Web Application) ได้อย่างสะดวกรวดเร็ว แองกูลาร์เจเอสนั้นจะมีลักษณะที่สำคัญ (Features) คือ

1) การผูกข้อมูลแบบสองทาง (Two Way Data Binding) คือ กระบวนการทำงานหนึ่งในแองกูลาร์เจเอส เมื่อข้อมูลมีการเปลี่ยนแปลงเมื่อใดส่วนการแสดงผลที่ถูกผูกไว้กับข้อมูลนั้น จะถูกเปลี่ยนไปให้ตรงกับข้อมูลที่ได้เปลี่ยนไปโดยอัตโนมัติ

2) เทมเพลต (Templates) คือ กลุ่มของภาษาเอชทีเอ็มแอลที่ถูกแปลงให้กลายเป็นดอม (Document Object Model: DOM) ซึ่งก็คือโครงสร้างของหน้าเว็บแอปพลิเคชัน แองกูลาร์เจเอสเฟรมเวิร์คนั้นสามารถควบคุมเทมเพลตให้ทำงานตามที่คุณพัฒนาต้องการได้

3) เอ็มวีซี (Model-View-Controller: MVC) คือ สถาปัตยกรรมรูปแบบหนึ่งที่แยกการทำงานออกเป็น 3 ส่วนหลัก เพื่อให้ นักพัฒนาสามารถแยกการทำงานเป็นอิสระออกจากกัน ดังนี้

โมเดล เป็นส่วนที่ติดต่อกับฐานข้อมูล โดยโมเดลจะรับคำสั่งมาจากคอนโทรลเลอร์ ซึ่งเป็นคำสั่งในการทำงานกับฐานข้อมูล โดยหลังจากโมเดลทำงานตามคำสั่งเรียบร้อยแล้วจะส่งค่ากลับไปยังคอนโทรลเลอร์อีกครั้งเพื่อให้คอนโทรลเลอร์ส่งให้วิวต่อไป

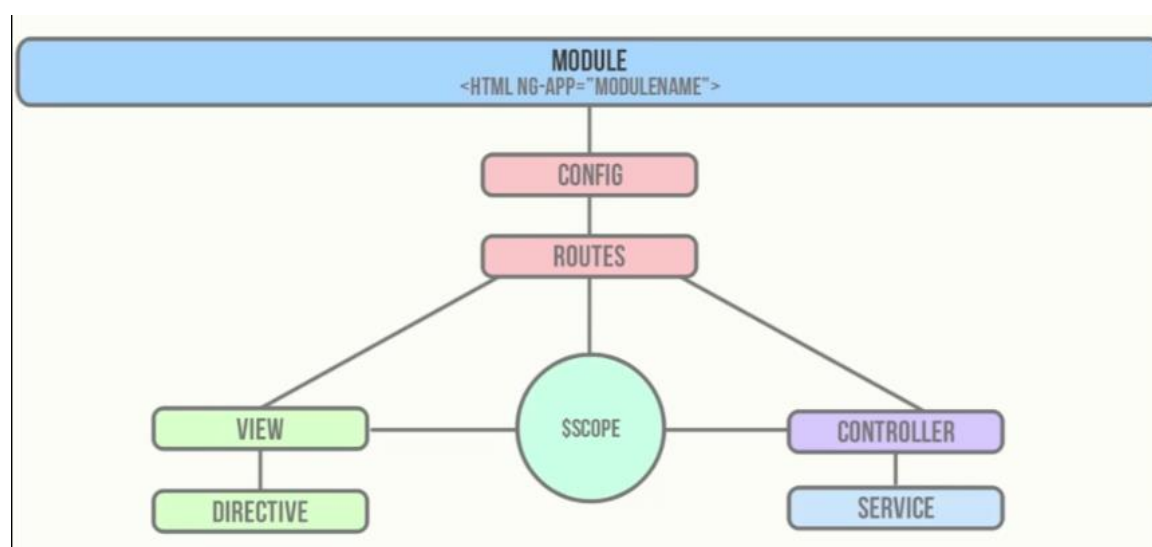
วิว เป็นส่วนหลักของส่วนติดต่อกับผู้ใช้ ซึ่งวิวสามารถมีเทมเพลตของเลเอาท์ที่แตกต่างกันไป โดยข้อมูลต่างๆจะถูกส่งมาจากคอนโทรลเลอร์ แล้วนำมาแสดงผลที่วิว

คอนโทรลเลอร์ เป็นตัวเชื่อมระหว่างวิว และโมเดล เช่นการรับค่าจากฟอร์มในวิว ส่งค่าให้กับคอนโทรลเลอร์แล้ว คอนโทรลเลอร์ก็จะทำการสร้างลอจิกในการบันทึกข้อมูลลงฐานข้อมูลต่อไป

4) ดีเพนเดนซีอินเจกชัน (Dependency Injection) คือการเพิ่มเซอร์วิส (Service) ที่เกี่ยวข้องกับการทำงานของแองกูลาร์เจส เพื่อให้แองกูลาร์เจสทำการโหลดเซอร์วิส ให้เสร็จสิ้นก่อนที่จะทำงานส่วนหลักของไดเรกทีฟ

5) ไดเรกทีฟ (Directives) คือ กลุ่มของเทมเพลตที่ทำงานได้ตามที่นักพัฒนาต้องการ โดยมีการเพิ่มโค้ดเพื่อควบคุมคอม ให้แสดงผลตามที่นักพัฒนาต้องการ

แผนภาพสถาปัตยกรรมและส่วนประกอบ (Angular JS Application Structure)



ภาพที่ 2.8 สถาปัตยกรรมแองกูลาร์เจส

1) โมดูล คือส่วนประกอบต่างๆ ของแอปพลิเคชัน แบ่งออกเป็นโมดูล เพื่อให้ง่ายต่อการจัดการซอฟต์แวร์ ซึ่งจะมีคอนโทรลเลอร์, เซอร์วิส ที่จะนำมาใช้กรอกข้อมูลต่างๆ รวมถึงส่วนของการตั้งค่าคอนฟิก โดยโมดูลหนึ่งสามารถใช้โมดูลอื่นๆได้

2) สโคป ทำหน้าที่เชื่อมโยงระหว่างคอนโทรลเลอร์และวิว

3) ไดเรกทีฟ ทำหน้าที่ในการจัดการเอชทีเอ็มแอลแท็ก ที่สามารถกำหนด และควบคุมการทำงานของแท็กตาที่ผู้พัฒนาต้องการได้

4) เซอร์วิส แองกูลาร์เจสมีเซอร์วิสภายในต่างๆ ที่จะช่วยให้สามารถทำงานได้สะดวกมากยิ่งขึ้น

ขั้นตอนในการเรียกใช้แองกูลาร์เจเอส

ขั้นตอนที่ 1 : จะต้องทำการประกาศการใช้งานแองกูลาร์เจเอส นั้นไว้ในหน้าที่ใช้งานเสมอ

โปรแกรม 2.5 การนำเข้าไฟล์เพื่อเรียกใช้แองกูลาร์เจเอส

```
<script src="/src/angular.min.js"></script>
```

ขั้นตอนที่ 2 : ประกาศ ng-app directive ซึ่งเป็นการใช้ประกาศแอปพลิเคชันแองกูลาร์เจเอส

โปรแกรม 2.6 การประกาศใช้ ng-app directive

```
<div ng-app="" class="ng-scope">
...
</div>
```

ขั้นตอนที่ 3: ประกาศ ng-model directive ซึ่งใช้ผูกค่าต่างๆกับเอชทีเอ็มแอล เพื่อควบคุมข้อมูล

โปรแกรมที่ 2.7 การประกาศใช้ ng-model directive

```
Enter your Name: <input ng-model="name" type="text">
```

ขั้นตอนที่ 4: ประกาศ ng-bind directive ซึ่งใช้ผูกข้อมูลเพื่อแสดงในเอชทีเอ็มแอล

โปรแกรมที่ 2.8 การประกาศใช้ ng-bind directive

```
Hello <span ng-bind="name"></span>
```

ตัวอย่างการใช้แองกูลาร์เจเอส

โปรแกรมที่ 2.9 การเขียน input โดยใช้แองกูลาร์เจเอส

```
<script src="/src/angular.min.js"></script>
<div ng-app="">
    Name: <input ng-model="name" type="text">
    <div ng-bind="name">
</div>
</div>
```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.9

Name:

ภาพที่ 2.9 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.9

โปรแกรมที่ 2.10 การใช้ modules และ controllers

```
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

Try to change the names.

<div ng-app="myApp" ng-controller="myCtrl">
First Name: <input ng-model="firstName" type="text">
Last Name: <input ng-model="lastName" type="text">
Full Name: {{firstName + " " + lastName}}
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.firstName= "Nop";
    $scope.lastName= "Mayoe";
});
</script>
```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.10

Try to change the names.

First Name:

Last Name:

Full Name: Nop Mayoe

ภาพที่ 2.10 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.11

จากโปรแกรมที่ 2.10 จะมีการประกาศโมดูล โดยอ้างอิง ng-app="myApp"

โปรแกรมที่ 2.11 การประกาศโมดูล

```
var app = angular.module('myApp', []);
```

ส่วนการประกาศคอนโทรลเลอร์โดยอ้างอิง ng-controller="myCtrl"

โปรแกรมที่ 2.12 การประกาศคอนโทรลเลอร์

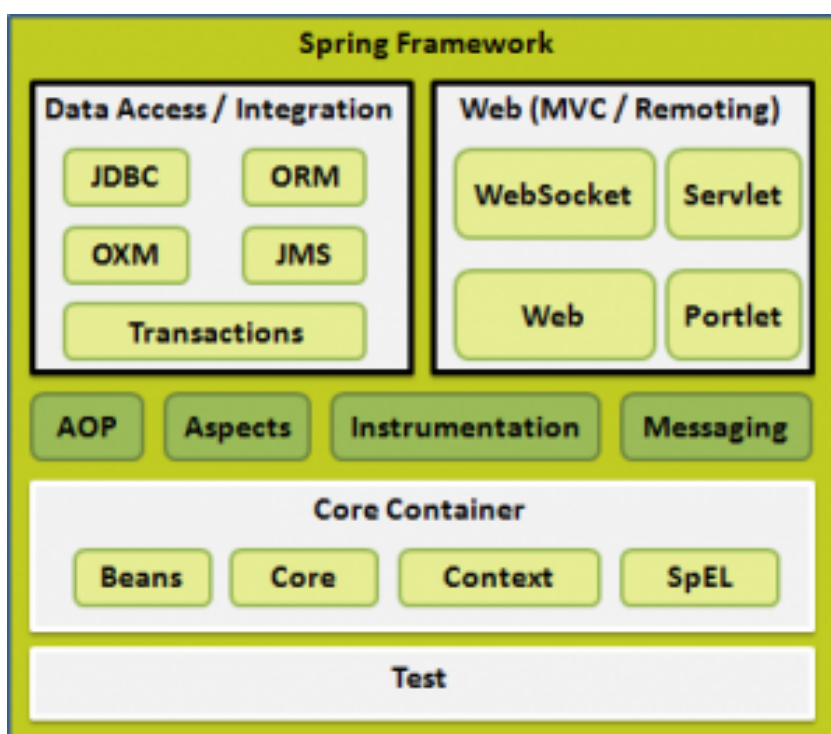
```
app.controller('myCtrl', function($scope) {
    $scope.firstName= "Nop";
    $scope.lastName= "Mayoe";
});
```

2.3.3 สปริงเฟรมเวิร์ค (Spring Framework)

สปริงเฟรมเวิร์ค คือ เฟรมเวิร์คในการสร้างเว็บ ที่รองรับแนวคิดแบบ MVC (Model, View, Controller) โดยหลักการทำงานของ Spring Framework จะออกแบบให้การทำงานทุกอย่างขึ้นอยู่กับ Servlet ที่ชื่อว่า Dispatcher Servlet

ต่อมาเฟรมเวิร์กนี้เป็นที่นิยมมากขึ้น จึงได้พัฒนาต่อโดยเพิ่มโมดูลตั้งแต่ AOP, Spring MVC, Security, Transaction, Webflow เป็นต้น โดยยังคงใช้ปรัชญาเดิมที่ทำให้การพัฒนา J2EE application ง่ายขึ้น และทาง Spring framework นั้นมีเอกสารการอธิบายที่ดี ชี้ให้เห็นถึงปัญหาความซับซ้อนของการพัฒนาแอปพลิเคชันในปัจจุบัน

โครงสร้างของสปริงเฟรมเวิร์ค (Spring Framework Architecture)



ภาพที่ 2.11 โครงสร้างของสปริงเฟรมเวิร์ค

1) คอร์คอนเทนเนอร์ (Core Container) ประกอบด้วย Core, Beans, Context, และ Expression Language modules(SpEL)

- Core module เป็นองค์ประกอบพื้นฐานของ framework รวมไปถึง IoC และ Dependency Injection
- Bean module ให้บริการ BeanFactory ในรูปแบบ factory pattern

- Context module ถูกสร้างโดย Core และ Beans modules และยังถูกใช้เป็นตัวกลางในการเข้าถึง object ต่าง ๆ และใช้ในการกำหนดค่าต่าง ๆ ให้ object
 - SpEL module ให้บริการ expression language ซึ่งใช้สำหรับ query และจัดการกับ object
- 2) การเข้าถึงข้อมูล/การบูรณาการ (Data Access/Integration) ประกอบไปด้วย JDBC, ORM, OXM, JMS และ Transaction modules
- JDBC module ทำให้การเชื่อมต่อ jdbc โดยไม่ต้อง coding
 - ORM module ให้บริการ object-relational mapping APIs ซึ่งประกอบด้วย JPA, JDO, Hibernate, และ iBatis
 - OXM module ให้บริการ Object/XML mapping ซึ่งต้องใช้ใน JAXB, Castor, XMLBeans, JiBX และ XStream
 - Java Messaging ให้บริการ JMS module ประกอบด้วย features สำหรับรับและส่ง message
 - Transaction module สนับสนุน transaction management สำหรับ class ซึ่ง implement special interfaces
- 3) ชั้นของเว็บ (Web layer) ประกอบด้วย Web module ,Web-MVC, Web-Socket, และ Web-Portlet modules
- Web module ให้บริการการทำงานพื้นฐานของเว็บ เช่น multipart file-upload และ Inversion of Control(IoC)
 - Web-MVC module ประกอบด้วย model-view-controller (MVC) ซึ่งใช้สำหรับพัฒนา Web Application
 - Web-Portlet สนับสนุนการใช้งาน MVC บน Portlet(portlet คือ web application แต่ portlet จะเป็นระบบย่อยๆ ที่จะถูกจัดการด้วย Portal กล่าวคือ ถ้าจะเปรียบ Portal เป็น OS ตัว portlet คือ application)
- 4) โมดูลเบ็ดเตล็ด (Miscellaneous Module) มีดังนี้ AOP, Aspects, Instrumentation, Web และ Test modules
- AOP module สนับสนุนแนวความคิด aspect-oriented programming(AOP) (AOP เป็น paradigm หนึ่งของการเขียนโปรแกรมที่เพิ่มความเป็นโมดูลโดยอนุญาตให้แยกคอนเซิร์นแบบตัดขวางได้)

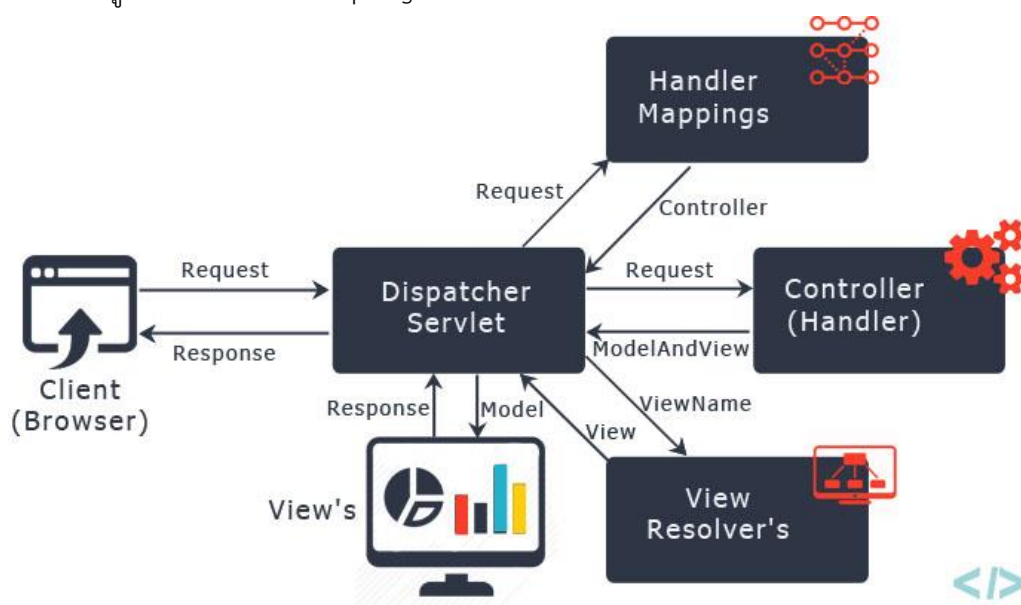
- Aspects module สนับสนุน AspectJ(เป็นการเพิ่มเติมโครงสร้างภาษาใหม่ เข้าไปในภาษา java เพื่อให้สามารถเขียนโปรแกรมแบบ AOP ใน java)

- Instrumentation module สนับสนุนการจัดการเกี่ยวกับ class

- Messaging module สนับสนุน Text Oriented Message Protocol (STOMP) เช่น WebSocket

- Test module สนับสนุนการทดสอบ Spring components ด้วย JUnit หรือ TestNG frameworks

การไหลของข้อมูลของสปริงเอ็มวีซี (Spring MVC Flow)



ภาพที่ 2.12 การไหลของข้อมูลของสปริงเอ็มวีซี

ขั้นตอนที่ 1: Dispatcher Servlet จะได้รับคำขอจากบราวเซอร์

ขั้นตอนที่ 2: Dispatcher Servlet จะส่งคำขอไปยัง Handler Mapping และทำความเข้าใจกับชื่อคลาสคอนโทรลเลอร์ที่เชื่อมโยงกับคำขอที่กำหนด

ขั้นตอนที่ 3: เมื่อได้ชื่อคลาสคอนโทรลเลอร์แล้ว Dispatcher Servlet จะทำส่งคำขอไปยังคอนโทรลเลอร์คลาสนั้น จากนั้นคอนโทรลเลอร์จะประมวลผลคำขอโดยการหาเมธอดที่เหมาะสม และส่งคืนออบเจกต์โมเดล และวิว (ประกอบด้วยข้อมูลโมเดล และชื่อวิว) กลับไปที่ Dispatcher Servlet

ขั้นตอนที่ 4: Dispatcher Servlet ส่งชื่อวิวไปยัง View Resolver เพื่อรับหน้าเว็บ

ขั้นตอนที่ 5: สุดท้าย Dispatcher Servlet จะส่งออบเจกต์โมเดลไปยังหน้า View เพื่อแสดงผลทางบราวเซอร์

2.3.4 จาวา (Java)

จาวา คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ โดยภาษาจาวามีจุดประสงค์เพื่อใช้แทนภาษา C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับ ภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิม ภาษานี้ถูกเรียกว่า “ภาษาโอ๊ก (Oak)” ซึ่งตั้งชื่อตามต้นโอ๊ก ใกล้ที่ทำงานของ เจมส์ กอสลิง ในภายหลังจึงเปลี่ยนชื่อเป็น “จาวา” ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของ ภาษาจาวาอยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ OOP (Oriented Object Programming) มาพัฒนาโปรแกรมของตนด้วยภาษาจาวา

ส่วนประกอบของโปรแกรมภาษาจาวา

1) แพคเกจ (Package) เป็นกลุ่มของคลาสหรือไลบรารีมาตรฐานของภาษาจาวา มีฟังก์ชันต่าง ๆ ให้ใช้มากมาย เช่น ช่วยสนับสนุนงานด้าน Input และ Output งานด้าน Graphic User Interface (GUI) การจัดการข้อความ การคำนวณทางคณิตศาสตร์ และอื่นๆ

2) คลาส (Class) กลุ่มของวัตถุ (Object) ที่มีโครงสร้างพื้นฐานพฤติกรรมเดียวกัน ดังนั้น วัตถุที่มีคุณสมบัติลักษณะเดียวกันนี้ จะรวมกลุ่มอยู่ในคลาสเดียวกัน จึงสามารถสรุปได้ว่าคลาส คือ ต้นแบบข้อมูล ที่มีไว้เพื่อสร้างวัตถุ นอกจากจะมีชื่อคลาสที่บอกคุณสมบัติของคลาสนั้นแล้วยังมี คุณสมบัติ (Attribute) และ การทำงาน (Operation) ต่าง ๆ ซึ่งเป็นตัวอธิบายรายละเอียด และหน้าที่ต่าง ๆ ด้วย ในส่วนของการประกาศคลาส จะต้องประกาศคลาสให้ชื่อตรงกับไฟล์เสมอ นอกจากอินเนอร์ คลาส (Inner Class) ที่อยู่ในคลาสเดียวกัน โดยชื่อคลาสนั้นควรจะขึ้นต้นด้วยตัวใหญ่ และถ้ามีหลายคำให้ใช้ตัวพิมพ์ใหญ่แบ่งดังนี้

โปรแกรมที่ 2.13 การสร้างคลาสด้วยภาษาจาวา

```
public class ClassName {
    ...
}
```

3) เมธอด (Method) เป็นกลุ่มของคำสั่งที่สร้างขึ้นมาเพื่อทำหน้าที่บางอย่าง การสร้างเมธอดจะสามารถทำให้เราใช้ได้นั้นซ้ำๆ โดยที่ไม่ต้องเขียนโปรแกรมใหม่ เมื่อสร้างเมธอดในภาษา Java มันสามารถที่จะเรียกใช้งานได้จากส่วนใดๆ ของโปรแกรม ขึ้นกับขอบเขตและระดับการเข้าถึงที่ได้กำหนดขึ้น โดยรูปแบบในการสร้างเมธอดในภาษาจาวา เป็นดังนี้

- public ใช้นิยามตัวแปร, เมธอดและ คลาสใดๆ เพื่อให้สามารถนำไปใช้กับคลาส หรือ โปรแกรมอื่นๆ ได้
- private ใช้นิยามตัวแปร หรือ เมธอดเพื่อให้เรียกใช้ได้เฉพาะภายในคลาสที่สร้างตัวแปร หรือ เมธอดนั้นๆ ขึ้นมาเท่านั้น

- protected ใช้นิยามตัวแปร หรือ เมธอดที่ใช้ได้เฉพาะคลาสที่สร้างขึ้นมาด้วยวิธีการสืบทอด (Inheritance) เท่านั้นโดยปกติจะใช้ protected กับคลาสที่เป็นคลาสต้นฉบับ
- static ใช้นิยามตัวแปรและ เมธอดที่ต้องการให้มีค่าคงที่ สามารถนำไปใช้ได้กับทุกๆ ส่วนของคลาส โดยค่านั้นจะไม่มีเปลี่ยนแปลงไม่ว่ากรณีใดๆ
- void ใช้นิยาม เมธอดโดยเป็นการกำหนดให้ไม่มีการส่งค่าใดๆ กลับมาให้กับ Method นี้ (ดังนั้นจึงไม่ต้องใช้คำสั่ง return)

โปรแกรมที่ 2.14 การสร้างเมธอดด้วยภาษาจาวา

```
private void methodName() {
    .....
}
```

และในการประมวลผลโปรแกรมได้จะต้องมีเมธอดที่ชื่อว่า Main ดังตัวอย่างในโปรแกรมที่ 2.15 ซึ่งเมธอด Main จะเป็นที่แรกที่โปรแกรมจะเริ่มทำงาน

โปรแกรมที่ 2.15 เมธอด Main

```
public static void main (String[] args) {
    ...
}
```

4) Statements เป็นคำสั่งของโปรแกรมเพื่อให้โปรแกรมทำงานตามต้องการ เช่น `System.out.println("Hello World!");` เป็นการแสดงผลข้อความออกทางหน้าจอ โดยปกติโปรแกรมมักจะมีหลายคำสั่ง

โปรแกรมที่ 2.16 ฟังก์ชัน Hello World ภาษาจาวา

```
// Hello World Program
public class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello World!");
    }
}
```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.16

Hello World!

ภาพที่ 2.13 ภาพที่ผลลัพธ์การประมวลผลโปรแกรมที่ 2.16

2.3.5 ไพทอน (Python)

ไพทอน เป็นภาษาโปรแกรมขั้นสูงชนิดหนึ่ง ที่ถูกออกแบบมาให้โค้ดสามารถอ่านได้ง่าย โดยการใช้ whitespace เป็นตัวแบ่งว่าโค้ดแต่ละบรรทัดอยู่ภายใต้บล็อกใดแทนการใช้ปีกกาแบบภาษาอื่น โครงสร้างภาษาไพทอนมีพื้นฐานการใช้จำนวนบรรทัดให้น้อย ถ้าเทียบกับ C++ หรือ JAVA อีกอย่างหนึ่งภาษาไพทอนนี้เป็น Open Source เหมือนอย่างพีเอชพี ทำให้ทุกคนสามารถนำไพทอนมาพัฒนาโปรแกรมได้โดยไม่ต้องเสียค่าใช้จ่าย

คุณลักษณะเด่นของภาษาไพทอน คือมีการสนับสนุนแนวคิดเชิงวัตถุ โดยโค้ดที่เขียนด้วยไพทอนจะสามารถนำไปประมวลผลได้หลากหลายระบบปฏิบัติการ

โครงสร้างของภาษาไพทอน

1) โมดูล (Module) คือไฟล์ของไพทอน ซึ่งโมดูลจะประกอบไปด้วยคลาส ฟังก์ชัน และตัวแปรต่างๆ และนอกนี้ยังสามารถนำโมดูลอื่นเข้ามาในโปรแกรมได้ ซึ่งโมดูลอาจจะอยู่ภายในแพ็คเกจ ซึ่งเป็นเหมือนไดเรกทอรีของโมดูล

2) คอมเมนต์ (Comment) คอมเมนต์ในภาษาไพทอนนั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย Whitespace หรือโค้ดของโปรแกรม หรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายข้อโค้ดที่เขียนขึ้น และไม่มีผลต่อการทำงานของโปรแกรม

โปรแกรมที่ 2.18 การคอมเมนต์ในภาษาไพทอน

```
# My first Python program

'''
This is a multiline comment
'''

print ('Hello Python.') # Inline comment
```

ในตัวอย่างได้คอมเมนต์สามแบบด้วยกัน แบบแรกเป็นการคอมเมนต์แบบบรรทัดเดียว (Single line) แบบที่สองเป็นการคอมเมนต์แบบหลายบรรทัด (Multi line) และแบบสุดท้ายเป็นการคอมเมนต์แบบ inline หรือการคอมเมนต์ภายในบรรทัดเดียวกันกับโค้ด

3) สเตทเมนต์ (Statement) คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษาไพทอนนั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษาซี และจาวา ซึ่งใช้เครื่องหมายเซมิโคลอนสำหรับการจบคำสั่งการทำงาน แต่อย่างไรก็ตามในภาษาไพทอนนั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมายเซมิโคลอน

โปรแกรมที่ 2.19 ตัวอย่างสเตทเมนต์ของภาษาไพทอน

```
name = input('What is your name?\n')
print ('Hi, %s.' % name);
print ('Welcome to Python.');
```

จากตัวอย่างจะมี 4 คำสั่งในโปรแกรม สองบรรทัดแรกเป็นคำสั่งที่ใช้บรรทัดใหม่ในการจบคำสั่ง ซึ่งเป็นแบบปกติในภาษาไพทอน และบรรทัดสุดท้ายมีสองคำสั่งในบรรทัดเดียวที่คั่นด้วยเครื่องหมายเซมิโคลอนสำหรับการจบคำสั่ง

4) ตำแหน่งการเยื้อง และช่องว่าง (Indentation and Whitespace) ในภาษาไพทอนนั้น จะใช้ช่องว่างและแท็บสำหรับกำหนดบล็อกของโปรแกรม เช่นคำสั่ง if-else for หรือการประกาศฟังก์ชัน ซึ่งคำสั่งเหล่านั้นเป็นคำสั่งแบบบล็อก โดยจำนวนช่องว่างที่ใช้นั้นต้องเท่ากัน

โปรแกรมที่ 2.20 ตัวอย่างบล็อกคำสั่งในภาษาไพทอน

```
n = int(input ('Input an integer: '))

if (n > 0):
    print ('x is positive number')
    print ('Show number from 0 to %d' % (n - 1))

else:
    print ('x isn\'t positive number')

for i in range(n):
    print(i)
```

จากตัวอย่างเป็นบล็อกของโปรแกรมจากทั้ง 3 คำสั่ง ในคำสั่งแรกคือ if ในบล็อกนี้มีสองคำสั่งย่อยอยู่ภายใน ที่หัวของบล็อกนั้นจะต้องมีเครื่องหมาย : กำหนดหลังคำสั่งในการเริ่มต้นบล็อกเสมอ อีกสองบล็อกสุดท้ายนั้นเป็นคำสั่ง else และ for ซึ่งมีหนึ่งคำสั่งย่อยอยู่ภายใน ในภาษาไพทอนนี้ เข้มงวดกับช่องว่างภายในบล็อกมาก นั้นหมายความว่าทุกคำสั่งย่อยภายในบล็อกนั้นต้องมีจำนวนช่องว่างเท่ากันเสมอ

5) Literals คือเครื่องหมายที่ใช้แสดงค่าของค่าคงที่ในโปรแกรมภาษาไพทอน นั้นมี Literal ของข้อมูลประเภทต่างๆ เช่น Integer Floating-point number และ String หรือแม้กระทั่งตัวอักษร และ Boolean

โปรแกรมที่ 2.21 การกำหนด Literal ให้กับตัวแปรในภาษาไพทอน

```
a = 1
b = -1.64E3
c = True
d = "marcuscode.com"
e = 'A'
```

จากตัวอย่างเป็นการกำหนด Literal ประเภทต่างๆ ให้กับตัวแปรในค่าที่เป็นแบบตัวเลขนั้นสามารถกำหนดค่าลงไปโดยตรงได้ทันที และสามารถกำหนดในรูปแบบสั้นได้อย่างในตัวแปร b และสำหรับบูลีนนั้น จะเป็น True ส่วน String หรือ Character นั้นจะต้องอยู่ภายในเครื่องหมาย double quote หรือ single quote เสมอ

6) เอ็กซ์เพรสชัน (Expressions) คือการทำงานร่วมกันระหว่างค่าตั้งแต่หนึ่งไปจนถึงหลายค่า โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงานในภาษาไพทอนนั้น เอ็กซ์เพรสชันจะมีสองแบบ ดังนี้

- บูลีน เอ็กซ์เพรสชัน (Boolean expression) เป็นการกระทำกันของตัวแปร และตัวดำเนินการและจะได้ผลลัพธ์เป็นค่าบูลีน โดยทั่วไปแล้วมักจะเป็นตัวดำเนินการเปรียบเทียบค่า และตัวดำเนินการตรรกศาสตร์

- เอ็กซ์เพรสชันทางคณิตศาสตร์ คือการกระทำกันกับตัวดำเนินการและได้ค่าใหม่ที่ไม่ใช่บูลีน

โปรแกรมที่ 2.22 เอ็กซ์เพรสชันในภาษาไพทอน

```
a = 4
b = 5

# Boolean expressions
print(a == 4)
print(a == 5)
print(a == 4 and b == 5)
print(a == 4 and b == 8)

# Non-boolean expressions
print(a + b)
print(a + 2)
print(a * b)
print(((a * a) + (b * b)) / 2)
print("Python " + "Language")
```

จากตัวอย่างมีตัวแปร a และ b โดยกำหนดค่าให้กับตัวแปรเหล่านี้ และทำงานกับตัวดำเนินการประเภทต่างๆ ที่แสดงเอ็กซ์เพรสชันในรูปแบบของบูลีนเอ็กซ์เพรสชัน ที่จะได้ผลลัพธ์สุดท้ายเป็นเพียงค่า True และ False เท่านั้น ส่วนค่าอื่นๆนั้น สามารถเป็นค่าใดๆที่ไม่ใช่บูลีน

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.22

```
True
False
True
False
9
6
20
20.5
Python Language
```

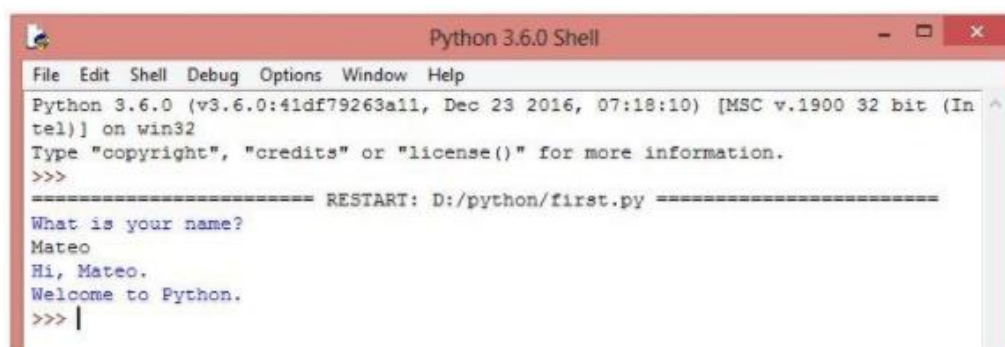
ภาพที่ 2.14 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.22

ตัวอย่างการเขียนโปรแกรมด้วยภาษาไพทอน

โปรแกรมที่ 2.23 โปรแกรมรับชื่อ และแสดงข้อความทางหน้าจอด้วยภาษาไพทอน

```
# My first Python program
name = input('What is your name?\n')
print ('Hi, %s.' % name)
print ('Welcome to Python.')
```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.23 ด้วย Python Shell



ภาพที่ 2.15 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.23

2.3.6 Flask Framework

Flask framework เป็นโครงสร้างการทำงานขนาดเล็ก ของ Python เพื่อใช้ร่วมกัน webserver อีกแบบหนึ่ง ที่ใช้พัฒนา web application บนพื้นฐานของ Werkzeug และโครงสร้างการทำงานของ Jinja2 ภายใต้การรับรองจาก BSD licensed

Flask สนับสนุนส่วนขยายที่สามารถเพิ่มคุณสมบัติของแอปพลิเคชันได้ราวกับถูกนำมาใช้ Flask itself มีส่วนขยายสำหรับ Mapping ระหว่างวัตถุที่ฝั่ง Client กับ Server การตรวจสอบรูปแบบ, จัดการการอัปเดต, เทคโนโลยีการรับรองความถูกต้องแบบเปิดต่างๆและเครื่องมือที่ใช้ร่วมกันหลายรูปแบบ ส่วนขยายได้รับการอัปเดตเป็นประจำสม่ำเสมอมากขึ้นกว่าโปรแกรม Flask core ตัวอย่างโค้ด แสดง แอปพลิเคชันเว็บ (web application)

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

ภาพที่ 2.16 ตัวอย่างโค้ด แสดง แอปพลิเคชันเว็บ (web application)

ความหมายของ Jinja

Jinja 2 เป็นโครงสร้างสำหรับสื่อสารระหว่าง Client กับ Server สำหรับ Python มีการสนับสนุน Unicode เต็มรูปแบบ ที่สามารถใช้งานร่วมกันกับระบบแบบ Sandboxed ใช้กันอย่างแพร่หลายและการรับรองจาก BSD licensed

ได้รับแรงบันดาลใจจากระบบเทมเพลตของ Django แต่ขยายด้วยภาษาที่แสดงออกซึ่งทำให้ผู้เขียนเทมเพลตมีชุดเครื่องมือที่มีประสิทธิภาพมากขึ้น และตรวจสอบอัตโนมัติสำหรับการใช้งานที่มีความสำคัญสูงสุด

คุณสมบัติ Jinja

1) โหมดการทำงานแบบ Sandboxed ทุกแง่มุมของการดำเนินการของเทมเพลตจะได้รับการตรวจสอบและระบุไว้ในรายการที่อนุญาตพิเศษหรืออยู่ในรายการที่ไม่อนุญาต ซึ่งทำให้สามารถเรียกใช้แม่แบบที่ไม่น่าเชื่อถือได้

- 2) มีความสามารถในการสร้าง HTML อัตโนมัติ เพื่อป้องกันการเขียนสคริปต์ข้ามเว็บไซต์
- 3) Template inheritance ทำให้สามารถใช้รูปแบบเดียวกันหรือแบบเดียวกันสำหรับเทมเพลตทั้งหมด
- 4) มีประสิทธิภาพด้านความเร็วในการติดต่อระหว่าง Server กับ Client เพราะผลการกระทำให้ทางฝั่ง Client
- 5) สร้าง Object ก่อนล่วงหน้าก่อนเวลาใช้งาน
- 6) ง่ายต่อการแก้ปัญหาด้วยระบบการหาข้อผิดพลาด แบบคอมไพล์และรันไทม์ ข้อผิดพลาดลงในระบบสืบค้นกลับ Python มาตรฐาน
- 7) ง่ายต่อการกำหนดค่า ตัวอย่างเช่นคุณสามารถปรับแต่ง Jinja2 ให้พอดีกับรูปแบบการแสดงผลเช่น LaTeX หรือ JavaScript
- 8) ผู้ช่วยออกแบบแม่แบบ Jinja2 มาพร้อมกับผู้ช่วยเล็ก ๆ น้อย ๆ ที่เป็นประโยชน์ซึ่งจะช่วยแก้ปัญหาทั่วไปในเทมเพลตต่างๆเช่นการแบ่งลำดับของรายการออกเป็นหลายคอลัมน์และอื่น ๆ

2.3.7 NoSQL

NoSQL ย่อมาจาก Not Only SQL คือแนวทางหนึ่งสำหรับจัดการข้อมูลและการออกแบบฐานข้อมูล สำหรับข้อมูลขนาดใหญ่ ที่ไม่มีโครงสร้างที่ชัดเจน ระบบต้องการการทำงานแบบเรียลไทม์ หรือใช้สำหรับการวิเคราะห์ข้อมูลแบบออฟไลน์ เช่น ระบบการวิเคราะห์ข้อมูลจากโซเชียลเน็ตเวิร์ค เป็นต้น

ข้อดีของ NoSQL

- 1) สามารถขยายระบบได้ง่าย
- 2) รองรับข้อมูลขนาดใหญ่
- 3) รองรับรูปแบบข้อมูลที่หลากหลาย หรือมีความยืดหยุ่นสูงได้

ข้อเสียของ NoSQL

- 1) เป็นเทคโนโลยีใหม่ๆ ซึ่งยังไม่เสถียรมากนัก
- 2) ส่วนสนับสนุนผู้ใช้อย่างไม่เต็มที่เท่าที่ควร ส่วนใหญ่จะมีข้อมูลเฉพาะกลุ่ม
- 3) ขาดผู้มีความรู้และผู้เชี่ยวชาญ

โครงสร้างข้อมูลของ NoSQL Database 4 ชนิดประกอบด้วย

- 1) Column-based databases ฐานข้อมูลที่เก็บข้อมูลในรูปของตารางคล้ายๆกับฐานข้อมูลเชิงสัมพันธ์แต่จะเก็บข้อมูลในแบบคอลัมน์แทนแบบแถว ซึ่งเหมาะกับการทำเหมืองข้อมูล ตัวอย่างของฐานข้อมูลที่เป็นลักษณะนี้ได้แก่ Hbase, Cassandra, Hypertable เป็นต้น

2) Document-based Database เป็นฐานข้อมูล NoSQL ที่ทำการบันทึกข้อมูลเป็น JSON Structure หรือ เป็นเอกสาร ซึ่งเป็นชุดของข้อความยาวๆ ทำให้มีอิสระในการจัดเก็บข้อมูลประเภทใดๆ ก็ได้ไม่ว่าเป็น Integer, String หรือข้อความใดๆ ก็ตาม และไม่ต้องกำหนดประเภท หรือ รูปแบบของข้อมูลล่วงหน้า แต่อย่างไรก็ตามตัวอย่างของ NoSQL กลุ่มนี้ได้แก่ CouchDB และ MongoDB เป็นต้น

3) Key-value Store เป็นฐานข้อมูลที่สามารถจัดเก็บข้อมูลในรูปแบบใดๆ ก็ได้ตามอิสระ โดยข้อมูลเหล่านั้นจะสามารถถูกเข้าถึงได้ด้วยการระบุค่า Key ประจำข้อมูลนั้นๆ ตัวอย่างของฐานข้อมูลกลุ่มนี้เช่น Redis และ Riak

4) Graph-based data-stores เป็นฐานข้อมูลที่มีการระบุความสัมพันธ์ของข้อมูลแต่ละชุดเป็น Network หรือเป็น Graph โดยข้อมูลแต่ละ Node นั้นจะสามารถจัดเก็บข้อมูลได้อย่างอิสระ ตัวอย่างของ Graph Database นี้ก็เช่น Neo4j และ OrientDB

2.3.8 มงโกดีบี (MongoDB)

มงโกดีบี เป็น Open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือไม่มีความสัมพันธ์ (Relation) ของตารางแบบ SQL ทั่วไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทน การบันทึกข้อมูลทุกๆ เรคอร์ดในมงโกดีบีจะเรียกว่า Document ซึ่งจะเก็บค่าเป็น Key และ Value

โปรแกรมที่ 2.24 การเก็บค่าแบบมงโกดีบี

```
{
  "_id": ObjectId("554b8ee746e04bc5503aef47"),
  "name": "Chai"
}
```

การเก็บข้อมูล document ในมงโกดีบีจะถูกเก็บไว้ใน Collections (เปรียบเทียบกับตารางในฐานข้อมูลเชิงสัมพันธ์ทั่วไป) แต่แตกต่างกันที่ collection ไม่จำเป็นที่จะต้องมีการสร้างเหมือนกันก็สามารถบันทึกข้อมูลได้

เปรียบเทียบมองโกดีบี และมายเอชคิวแอลคร่าวๆ ดังนี้

ตารางที่ 2.1 ตารางเปรียบเทียบระหว่างมองโกดีบี และมายเอชคิวแอล

MySQL	MongoDB
Table	Collection
Row	Document
Column	Field
Joins	Embedded documents, linking

คำสั่งและมองโกเชลล์ (Mongo Shell) เบื้องต้น

การเพิ่มข้อมูล (Insert)

โปรแกรมที่ 2.25 การเพิ่มข้อมูล

```
db.things.save({a: 1, b: 2, c: 3})
WriteResult({ "nInserted" : 1 })
```

จากตัวอย่างเป็นการบันทึกข้อมูลลง Collection ชื่อว่า things

การค้นหาข้อมูล (Query)

โปรแกรมที่ 2.26 การค้นหาข้อมูล

```
> db.players.find()
```

จากตัวอย่าง ที่ได้ทำการเพิ่มข้อมูลลงไปใน Collection players ฉะนั้นลอง find() จะได้ผลลัพธ์ดังนี้

```
> db.players.find()

{ "_id" : ObjectId("55cf69a0bda97b92d65ea9de"), "name" : "Lionel Messi" }
{ "_id" : ObjectId("55cf69a0bda97b92d65ea9df"), "name" : "Cristiano Ronaldo" }
```

ภาพที่ 2.17 ผลลัพธ์จากการค้นหาข้อมูลใน Collection players

การค้นหาโดยกำหนดเงื่อนไข

โปรแกรมที่ 2.27 การค้นหาข้อมูลแบบมีเงื่อนไข

```
db.players.find({name: 'Cristiano Ronaldo'})

{ "_id" : ObjectId("55cf69a0bda97b92d65ea9df"), "name" :
"Cristiano Ronaldo" }
```

จากตัวอย่างเป็นการกำหนดเงื่อนไข เช่น ค้นหา Collection players ที่มี name เท่ากับ Lionel Messi และผลลัพธ์จากการค้นหาก็จะแสดงเฉพาะ name ที่มีค่าเท่ากับ Lionel Messi

การปรับปรุงข้อมูล

โปรแกรมที่ 2.28 การปรับปรุงข้อมูลใน Collection

```
> db.players.update(
  {'name': 'Lionel Messi'},
  {
    $set: {'age': 28}
  }
)
```

จากตัวอย่างโปรแกรมที่ 2.28 เป็นการปรับปรุงข้อมูลใน Collection plays ที่มี name เท่ากับ Lionel Messi โดยทำการปรับปรุง age ให้มีค่าเท่ากับ 28

การลบข้อมูล

โปรแกรมที่ 2.28 การลบข้อมูลใน Collection

```
> db.players.remove({'name': 'Lionel Messi'});
```

จากตัวอย่างเป็นการลบข้อมูลใน Collection players ที่มี name เท่ากับ Lionel Messi

2.3.9 VueJS

VueJS เป็นการผสมผสานกันระหว่างแองกูลาร์เจเอส กับรีแอค (React) เพื่อสร้างเป็นรีแอคทีฟคอมโพเนนท์ขึ้น โดยที่ไลบรารีหลักนั้นจะเน้นไปที่เลเยอร์วิวเท่านั้น นอกจากนั้น VueJS ยังมีความสามารถในการใช้งานแอปพลิเคชัน Single-Page ที่มีความซับซ้อนได้ ในส่วนของรีแอคทีฟนั้นคือการโต้ตอบแบบทันที กล่าวคือถ้าหากค่าตัวแปรใดตัวแปรหนึ่งในคอม มีการเปลี่ยนแปลง ตัวแปรนั้นก็จะสามารถเปลี่ยนได้ทันทีโดยไม่ต้องเช็คค่าตัวแปรใดถูกเปลี่ยน

การเรียกใช้งาน VueJS

โปรแกรมที่ 2.29 การเรียกใช้งาน Vue.js

```
<script src="https://unpkg.com/vue"></script>
```

การแสดงผลข้อมูลคอม

โปรแกรมที่ 2.30 การแสดงผลข้อมูลคอม

```
-.html
<div id="app">
  {{ message }}
</div>

-.js
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
})
```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.30

Hello Vue!

ภาพที่ 2.18 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.30

การเชื่อมโยงแอททริบิวต์

โปรแกรมที่ 2.31 การเชื่อมโยงแอททริบิวต์

```
-.html
<div id="app-2">
  <span v-bind:title="message">
    Hover your mouse over me for a few seconds
    to see my dynamically bound title!
  </span>
</div>

-.js
var app2 = new Vue({
  el: '#app-2',
  data: {
    message: 'You loaded this page on ' + new Date()
  }
})
```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.31

Hover your mouse over me for a few seconds to see my dynamically bound title!

ภาพที่ 2.19 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.31

การทำงานแบบมีเงื่อนไขและทำซ้ำ

โปรแกรมที่ 2.32 การทำงานแบบมีเงื่อนไขและทำซ้ำ

```

-.html
<div id="app-3">
  <p v-if="seen">Now you see me</p>
</div>
-.js
var app3 = new Vue({
  el: '#app-3',
  data: {
    seen: true
  }
})

```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.32

Now you see me

ภาพที่ 2.20 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.32

การใช้งาน v-for

โปรแกรมที่ 2.33 การใช้งาน v-for

```

-.html
<div id="app-4">
  <ol>
    <li v-for="todo in todos">
      {{ todo.text }}
    </li>
  </ol>
</div>
-.js
var app4 = new Vue({
  el: '#app-4',
  data: {
    todos: [
      { text: 'Learn JavaScript' },
      { text: 'Learn Vue' },
      { text: 'Build something awesome' }
    ]
  }
})

```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.33

1. Learn JavaScript
2. Learn Vue
3. Build something awesome

ภาพที่ 2.21 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.33

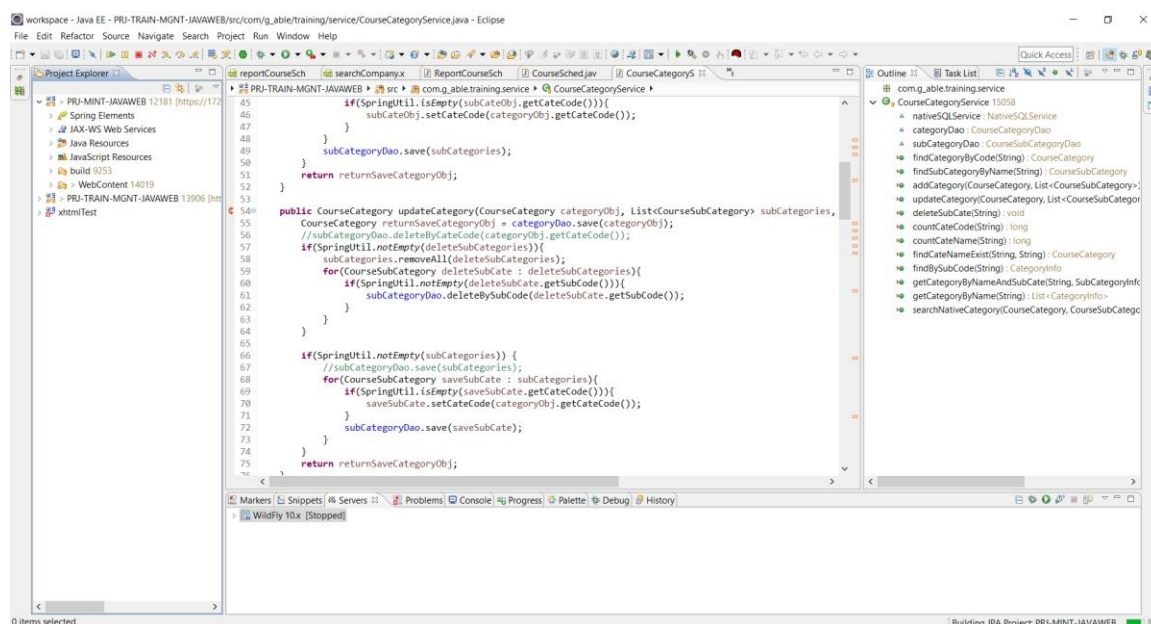
2.3.10 Eclipse EE Luna

Eclipse คือโปรแกรมที่ใช้สำหรับพัฒนาภาษา ซึ่งโปรแกรม Eclipse เป็นโปรแกรมหนึ่งที่ใช้ในการพัฒนาแอปพลิเคชันเซิร์ฟเวอร์ได้อย่างมีประสิทธิภาพ และเนื่องจาก Eclipse เป็นซอฟต์แวร์ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆจากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่ม

ความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug-ins ดังนั้นหากต้องการให้ Eclipse ทำงานใดเพิ่มเติม ก็เพียงแค่พัฒนา สำหรับงานนั้นขึ้นมา และนำปลั๊กอินนั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse Plug-in ที่มีมาพร้อมกับ Eclipse เมื่อ download มาครั้งแรกก็คือองค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียน และโปรแกรมภาษา Java

ข้อดีของโปรแกรม Eclipse คือ ติดตั้งง่าย สามารถใช้ได้กับ J2SDK ได้ทุกเวอร์ชัน รองรับภาษาต่างประเทศอีกหลายภาษา มี ที่ใช้เสริมประสิทธิภาพของโปรแกรม สามารถทำงานได้กับไฟล์หลายชนิด เช่น HTML, Java, C, JSP, EJB, XML และ GIF และที่สำคัญเป็นฟรีแวร์ (ให้ใช้งานได้ 90 วัน ถ้าจะใช้งานเต็มประสิทธิภาพต้องเสียค่าใช้จ่ายภายหลัง) ใช้งานได้กับระบบปฏิบัติการ, และ Mac OS ส่วนประกอบของโปรแกรม Eclipse EE Luna



ภาพที่ 2.22 ส่วนประกอบของโปรแกรม Eclipse

2.3.11 NodeJS

Node.js คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript สรุปล NodeJS ก็คือ Platform ตัวหนึ่งที่เขียนด้วย JavaScript สำหรับเป็น Web Server ทดสอบการทำงาน Node.js

หลักจากติดตั้ง Node.js ให้ทำการทดสอบด้วยการสร้างไฟล์ .js 1 ไฟล์ โดยด้านในมีโค้ดดังนี้

โปรแกรมที่ 2.34 โปรแกรม Hello World

```
//include package http เข้ามาเพื่อใช้สำหรับรัน web server
var http = require('http');

//ตั้งค่า server ให้แสดงผลคำว่า "Hello World" ส่งกลับไปที่ client
var server = http.createServer(function (request, response)
{
    response.writeHead(200, {"Content-Type": "text/plain"});
    response.end("Hello World\n");
});

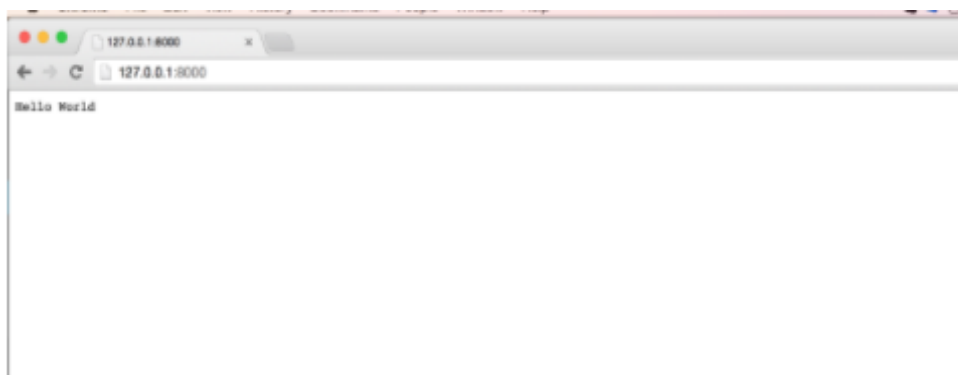
//กำหนด port ให้ server เป็น port 8000
server.listen(8000);
console.log("Server running at http://127.0.0.1:8000/");
```

จากนั้นเปิดเทอร์มินอล หรือซีเอ็มดี (Command Prompt) ใช้คำสั่ง cd เพื่อเข้าไปยังโฟลเดอร์ที่สร้างไฟล์ไว้ และรันไฟล์ด้วยคำสั่ง ต่อไปนี้

โปรแกรมที่ 2.35 การรันไฟล์ด้วยเทอร์มินอล

```
> cd /folder/to/your/js/file
> node app.js // เปลี่ยน app.js เป็นชื่อที่เซฟ กรณีที่เซฟเป็นชื่ออื่น
> Server running at http://127.0.0.1:8000/
```

ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.34



ภาพที่ 2.22 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.34

เมื่อสามารถประมวลผลโปรแกรมที่สร้างแล้ว ดังในตัวอย่างนี้จะประมวลผลเซิร์ฟเวอร์ผ่านทางโมดูลเอชทีทีพี ซึ่งอาจจะใช้งาน และจัดการได้ค่อนข้างยาก จึงมีการพัฒนาแพ็คเกจชื่อเอ็กซ์เพรส (Express) ขึ้นเป็น Web Application Framework สำหรับ Node.js ซึ่งอยู่ในส่วนของ NPM

NPM (Node package manager) คือ Tool สำหรับจัดการ package (บางคนอาจเรียก module) ที่มีการสร้างไว้แล้วมาใช้งานใน application ของเรา (เมื่อมี package ถูกนำมาใช้ใน application จะเรียก package เหล่านั้นว่า dependency ของ application) โดย NPM ถูกติดตั้งมาพร้อมกับ Node.js และสามารถใช้งานได้ผ่านทาง command-line ซึ่งแพ็คเกจของเอ็นพีเอ็มที่ได้รับคความนิยมคือ Express.js

ทดลองใช้เอ็นพีเอ็ม

เมื่อติดตั้งเอ็นพีเอ็มแล้วทดลองใช้ ซึ่งจะต้องการแก้ไขไฟล์ .js ที่สร้างตามโปรแกรมที่ 2.34 โดยจะใช้ express สำหรับประมวลผลเว็บเซิร์ฟเวอร์ ดังนี้

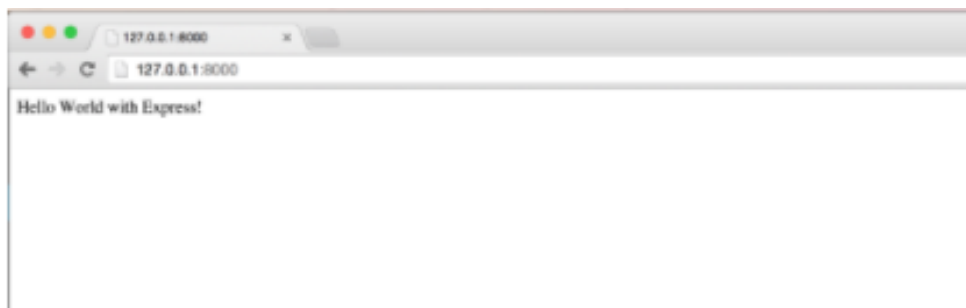
โปรแกรมที่ 3.36 การใช้ express ในการประมวลผลไฟล์

```
var express = require('express');

var app = express();
app.get('/', function (req, res) {
  res.send('Hello World with Express!');
});

var server = app.listen(3000, function () {
  var host = server.address().address;
  var port = server.address().port;
  console.log('Example app listening at http://%s:%s', host,
port);
});
```

เมื่อทำการประมวลผลโปรแกรมที่ 2.36 จะได้ผลลัพธ์ ดังนี้



ภาพที่ 2.23 ผลลัพธ์จากการประมวลผลโปรแกรมที่ 2.36