

บทที่ 1 แผนงานในการพัฒนาเว็บ

Web Development Roadmaps

ผู้สอน อ.ปวิม ชฎารัตนจิตติ

แผนงานสร้างเว็บส่วนเบื้องหน้า (Front-End Roadmap)

▶ ขั้นตอนที่ 1 เรียนรู้พื้นฐาน (Learn the Basics)

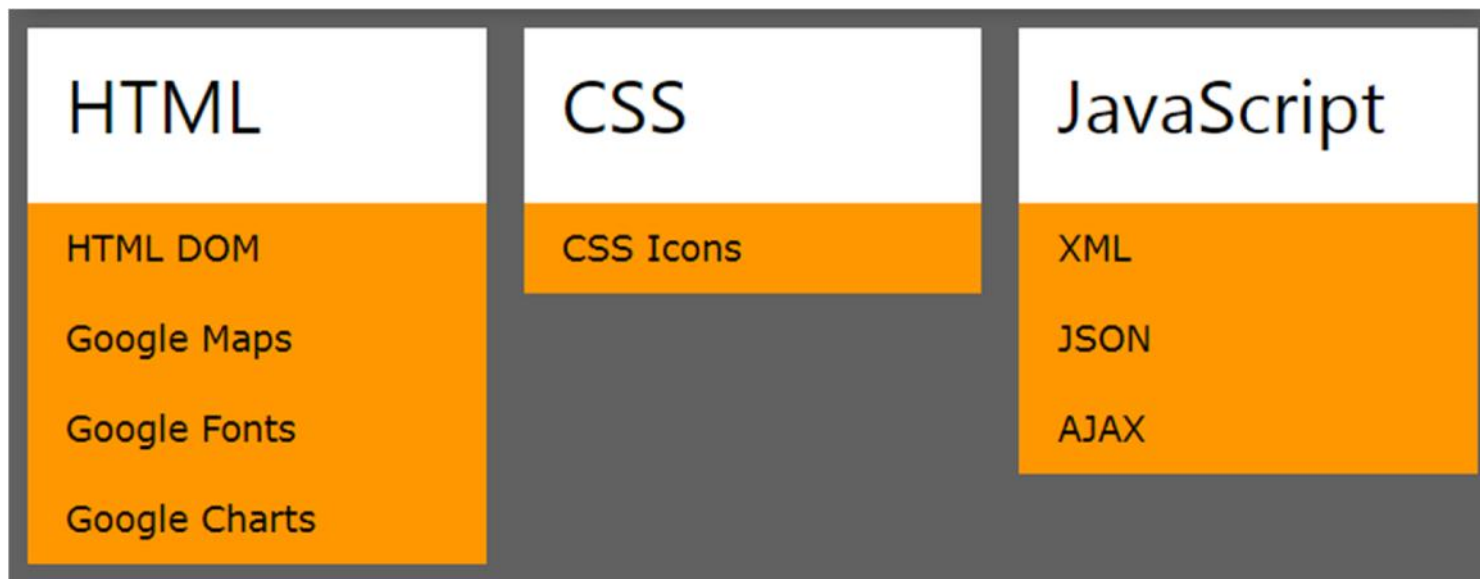
- ▶ นักพัฒนาเว็บไซต์ควรจะต้องมีความเข้าใจพื้นฐานใน HTML, CSS และ JavaScript
- ▶ เว็บไซต์ใหม่จะมีการพัฒนาให้เป็นแบบ Responsive Web Design
- ▶ JavaScript 5 (ECMAScript 5) เป็นภาษาที่รองรับกับโปรแกรมเว็บเบราว์เซอร์ในปัจจุบัน โดยจะต่างกับ JavaScript ปกติในเรื่องของการใช้อะเรย์ฟังก์ชัน



แผนงานสร้างเว็บส่วนเบื้องหน้า (Front-End Roadmap)

▶ ขั้นตอนที่ 2 เรียนรู้แบบเจาะลึก (Dig Deeper)

- ▶ เมื่อมีความเข้าใจและมั่นใจใน HTML และ CSS ถึงเวลาเรียนรู้ให้ลึกลงไป
- ▶ ควรเรียนรู้หลักในการสร้างสิ่งต่างๆ ในเว็บ เช่น แผนที่ การใช้ฟอนต์ การสร้างไอคอนในภาษา HTML การใช้ HTML DOM
- ▶ เรียนรู้วิธีใช้ AJAX และ JSON เพื่อเชื่อมต่อไปยังการรับส่งข้อมูลกับเซิร์ฟเวอร์

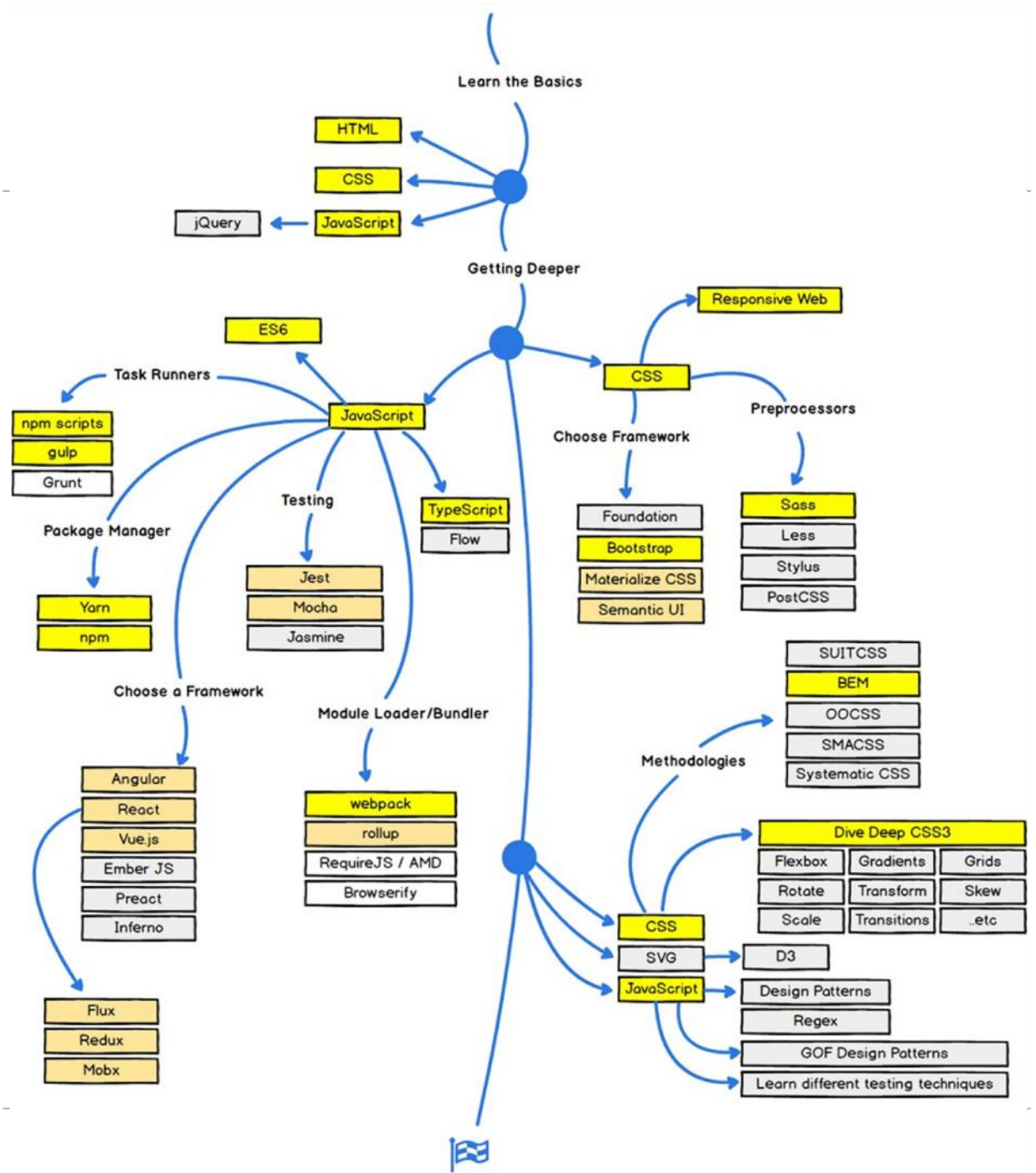


แผนงานสร้างเว็บส่วนเบื้องหน้า (Front-End Roadmap)

▶ ขั้นตอนที่ 3 เลือกใช้เฟรมเวิร์ค (Choose Frameworks)

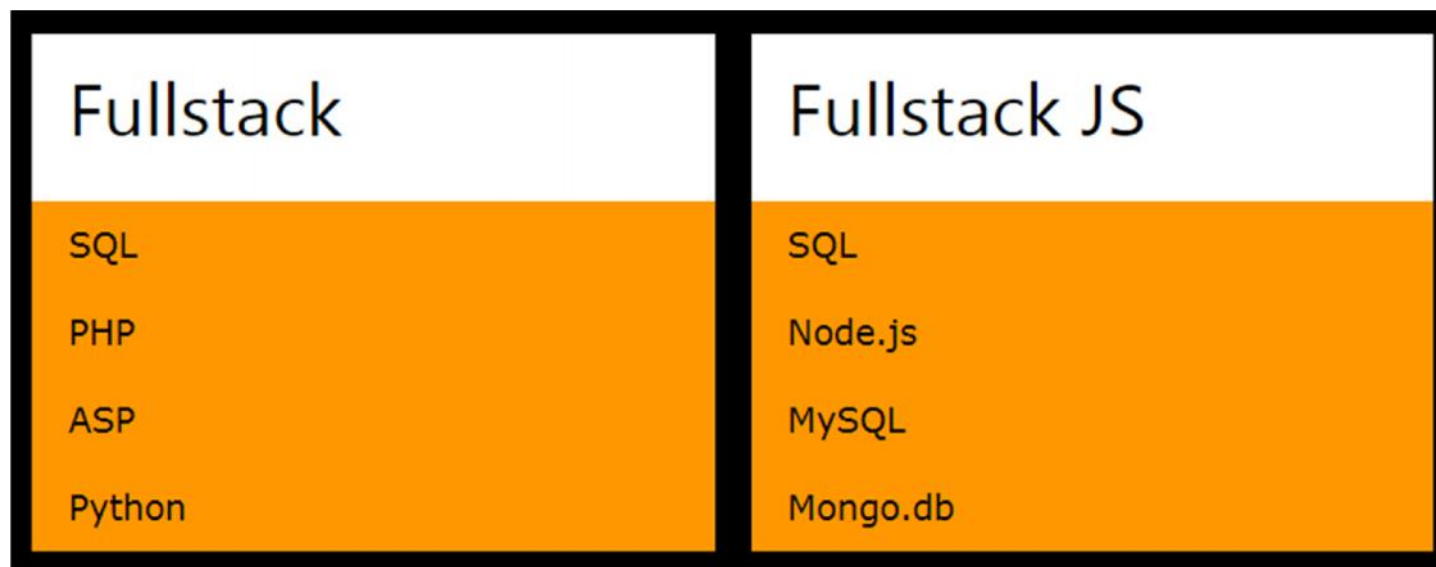
- ▶ ในส่วนของ CSS ควรเลือกเพื่อมาใช้ทำ Responsive Web Design ได้แก่ Bootstrap, Material Design หรือ W3.CSS
- ▶ ในส่วนของ JavaScript ควรเรียนรู้อย่างน้อย 1 ตัว ที่เป็นเฟรมเวิร์คยุคปัจจุบัน ได้แก่ React.js, Angular.js , Vue.js , W3.JS
- ▶ ถึงแม้ปัจจุบันจะมีความนิยมในการใช้ JQuery แต่คนส่วนใหญ่ก็ยังใช้ JavaScript เฟรมเวิร์คกันอยู่



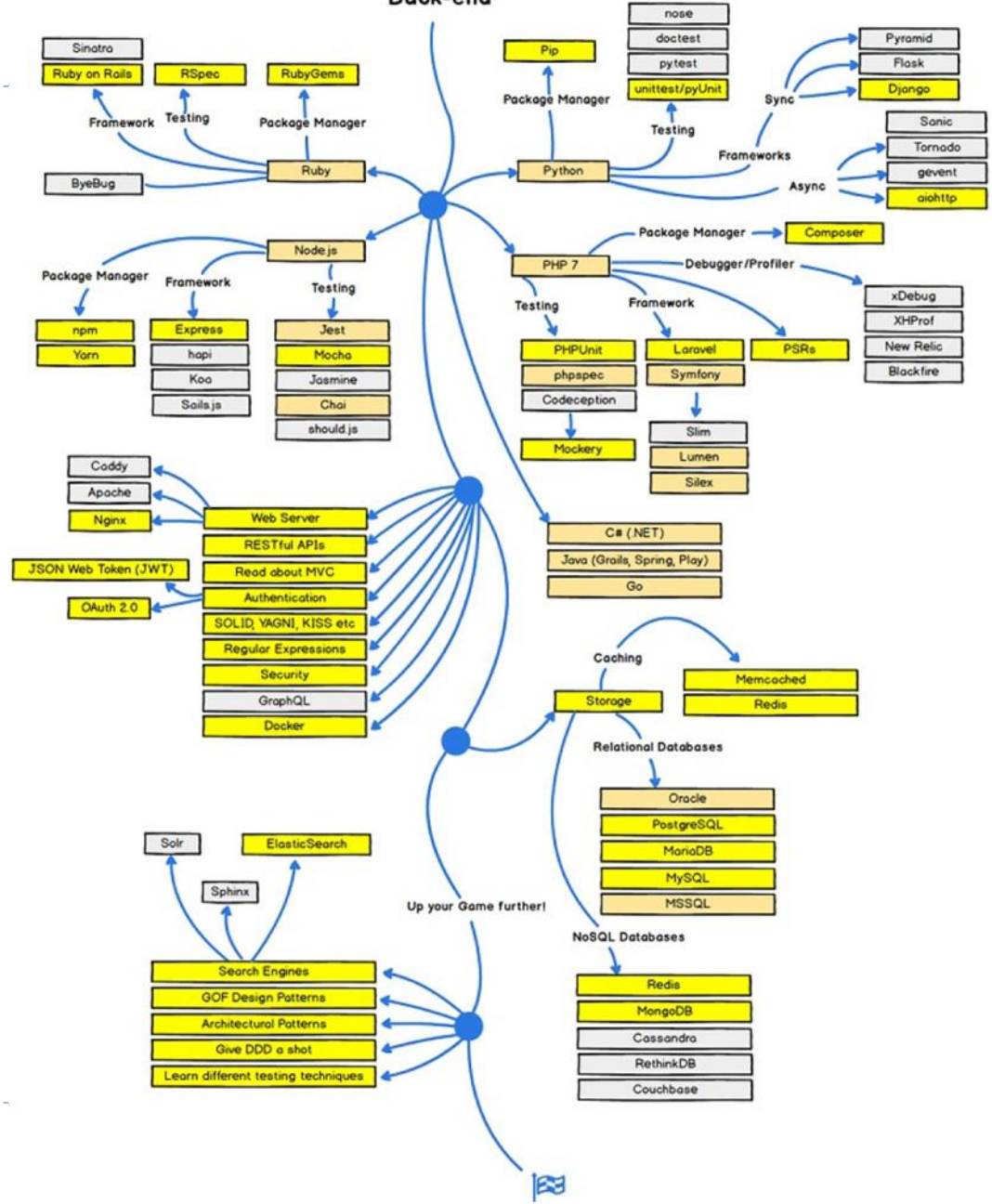


แผนงานสร้างเว็บส่วนเบื้องหลัง (Back-End Roadmaps)


- ▶ **ขั้นตอนที่ 4 สร้างเว็บส่วนเบื้องหลัง (Back-End Roadmaps)**
- ▶ วางระบบด้านหลังทั้งหมด ข้อมูลจะวิ่งจากไหนไปไหน ด้วยวิธีอะไร โครงสร้างเป็นอย่างไร ระบบความปลอดภัย ชนิดของฐานข้อมูลที่ใช้เก็บข้อมูล
 - ▶ **Fullstack** แบบสายทั่วไป ศึกษาเกี่ยวกับ SQL, PHP, ASP, Python
 - ▶ **Fullstack Javascript** สายจาวา ศึกษาเกี่ยวกับ SQL, Node.js, MySQL, Mongo.db



Back-end

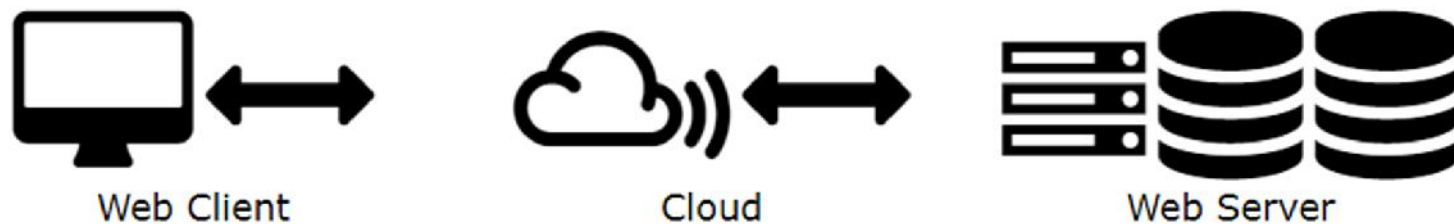


ความเชี่ยวชาญของนักพัฒนาเว็บไซต์

- ▶ เนื่องจากเว็บไซต์ และแอปพลิเคชันนั้นมีการเติบโตที่ค่อนข้างซับซ้อน และจำเป็นต้องพัฒนาอยู่ตลอดเวลา ผู้พัฒนาจึงต้องมีความเชี่ยวชาญให้มากขึ้นอยู่เสมอ โดยขึ้นอยู่กับสิ่งที่นักพัฒนาเว็บไซต์มีความสนใจแบ่งเป็น 3 ประเภทหลักได้แก่
 - ▶ **Front End Development** การพัฒนาโปรแกรมในส่วนหน้าบ้าน (ส่วนที่ทุกคนสามารถมองเห็นได้ของเว็บไซต์ หรือแอปพลิเคชัน)
 - ▶ **Back End Development** การพัฒนาโปรแกรมในส่วนหลังบ้าน (ส่วนของการทำงานเบื้องหลังจำพวก ฐานข้อมูล และ โครงสร้างพื้นฐาน)
 - ▶ **Full Stack Development** การพัฒนาอย่างเต็มรูปแบบทั้งหน้าบ้าน และหลังบ้าน ถือเป็นความเชี่ยวชาญที่ลงตัวของทั้งสองรูปแบบ โดยสามารถนำไปใช้กับ web stack , mobile stack หรือ native application stack แอปพลิเคชันดั้งเดิม เช่น โปรแกรมซอฟต์แวร์สำหรับอุปกรณ์เฉพาะ เป็นต้น
-
- 

Hyper Text Transfer Protocol (HTTP)

- ▶ **WWW** เป็นการติดต่อสื่อสารระหว่างเครื่องของผู้ใช้ กับเซิร์ฟเวอร์
- ▶ การติดต่อสื่อสารระหว่างเครื่องของผู้ใช้ กับเซิร์ฟเวอร์ทำได้โดยการส่ง HTTP Requests และได้รับข้อมูลตอบกลับในรูปแบบ HTTP Responses
- ▶ **เครื่องผู้ใช้(Clients)** ส่วนใหญ่เชื่อมต่อผ่านเบราว์เซอร์ เช่น Chrome, Edge, Safari, Firefox ซึ่งก็ขึ้นกับผู้ใช้เองว่าจะใช้โปรแกรมหรือ อุปกรณ์แบบไหนในการเชื่อมต่อ
- ▶ **เครื่องผู้ให้บริการ(Servers)** จะให้บริการอยู่บนระบบอินเทอร์เน็ตหรือผ่านระบบคลาวด์



HTTP Request / Response

- ▶ การติดต่อระหว่างผู้ใช้กับผู้ใช้บริการจะสมบูรณ์เมื่อมี การร้องขอ (Request) และการตอบสนอง (Response) ขั้นตอนคือ
 - ▶ 1. ผู้ใช้ส่งข้อมูล HTTP request ไปยังเว็บผ่านเบราว์เซอร์
 - ▶ 2. เว็บเซิร์ฟเวอร์ได้รับ HTTP request
 - ▶ 3. เซิร์ฟเวอร์รันโปรแกรม เพื่อประมวลผล HTTP request ที่ได้รับ
 - ▶ 4. เซิร์ฟเวอร์ส่ง HTTP response (ผลลัพธ์) กลับไปยังเบราว์เซอร์
 - ▶ 5. เบราว์เซอร์ของผู้ใช้ได้รับ HTTP response (ผลลัพธ์) และแสดงผล
-



วงจรในการร้องขอข้อมูลผ่าน HTTP (The HTTP Request Circle)

- ▶ รูปแบบทั่วไปที่มีการร้องขอผ่าน HTTP ได้แก่
 - ▶ เบราวเซอร์ร้องขอ HTML page เซิร์ฟเวอร์ตอบกลับในรูปแบบของ HTML file
 - ▶ เบราวเซอร์ร้องขอ style sheet เซิร์ฟเวอร์ตอบกลับในรูปแบบของ CSS file
 - ▶ เบราวเซอร์ร้องขอไฟล์ภาพ JPG เซิร์ฟเวอร์ตอบกลับในรูปแบบของ JPG file
 - ▶ เบราวเซอร์ร้องขอโค้ด JavaScript เซิร์ฟเวอร์ตอบกลับในรูปแบบของ JS file
 - ▶ เบราวเซอร์ร้องขอข้อมูล เซิร์ฟเวอร์ตอบกลับในรูปแบบของข้อมูล (XML หรือ JSON).



XHR - XML Http Request

- ▶ เบราวส์เซอร์ในปัจจุบันทั้งหมดมีการติดตั้ง XMLHttpRequest Object (XHR).
- ▶ XHR เป็นสิ่งที่อยู่ในรูปของ Javascript ถูกนำมาใช้ในการส่งถ่ายข้อมูลระหว่างเบราว์เซอร์กับเว็บเซิร์ฟเวอร์ เพื่อการแก้ไขข้อมูลที่แสดงบนหน้าเว็บ(web page)
- ▶ ถูกนำมาใช้กับโปรโตคอลส่งข้อมูลหลายประเภทไม่ใช่แต่เพียง HTTP ข้อมูลมีได้หลายรูปแบบไม่ว่าจะเป็น HTML, CSS, XML, JSON หรือ plain text



XHR - XML Http Request

- ▶ ข้อมูลที่ส่งในรูปวัตถุของ XHR มีข้อดีคือ
 - ▶ อัปเดตข้อมูลในเว็บได้โดยไม่ต้องโหลดหน้าเว็บใหม่
 - ▶ ขอข้อมูลใหม่ๆ จากเซิร์ฟเวอร์ได้ หลังโหลดหน้าเว็บไปแล้ว
 - ▶ รับข้อมูลอัปเดตใหม่จากเซิร์ฟเวอร์ได้ หลังโหลดหน้าเว็บไปแล้ว
 - ▶ ส่งข้อมูลไปยังเซิร์ฟเวอร์แบบเบื้องหลัง
- ▶ ข้อมูลที่ส่งในรูปวัตถุของ XHR อยู่ภายใต้แนวคิดของ AJAX และ JSON

