

#โมเดิร์น

# JavaScript

เก่งได้ใน 30 วัน

ฉบับ

Node.js  
+  
MongoDB



- พื้นฐาน JavaScript ครบทุกแง่มุม สำหรับนักเรียน นักศึกษา และโปรแกรมเมอร์
- การใช้ JavaScript จัดการกับ DOM
- สร้างเว็บแอปพลิเคชันฝั่งเซิร์ฟเวอร์ด้วย Node.js
- พื้นฐานการใช้งานฐานข้อมูล MongoDB พร้อมตัวอย่างแอปพลิเคชัน
- วิธีใช้งานแพ็คเกจยอดนิยม เช่น express, mongoose, jsonwebtoken, ejs และอื่น ๆ

**พิเศษ !**

ดาวน์โหลดโค้ดตัวอย่าง  
เพื่อใช้ประกอบการเรียน

จรัล วารินทร์

# คำนำ

JavaScript คือ ภาษาคอมพิวเตอร์ที่ถูกออกแบบมาให้ใช้งานบนเบราว์เซอร์ นักพัฒนาสามารถเขียนโค้ด JavaScript เพื่อแก้ไขเนื้อหาบนเว็บเพจ ใช้ JavaScript เพื่อสร้างแอนิเมชัน หรือใช้ JavaScript เพื่อปรับเปลี่ยนหน้าตาหรือสีสันทบนเว็บเพจก็ได้

แม้ว่า JavaScript จะถูกออกแบบมาให้ใช้งานบนเบราว์เซอร์ แต่นักพัฒนายังสามารถใช้ JavaScript นอกเบราว์เซอร์ผ่านทาง Node.js ได้อีกด้วย นั่นหมายความว่า การเรียนรู้ JavaScript เพียงภาษาเดียวก็เพียงพอสำหรับการสร้างเว็บแอปพลิเคชันทั้งในฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์

เพื่อให้ผู้อ่านสามารถเข้าใจ JavaScript ได้ดียิ่งขึ้น หนังสือเล่มนี้จึงถูกแบ่งออกเป็น 3 ส่วน ดังนี้

ส่วนที่ 1 เป็นการปูพื้นฐาน JavaScript ที่ควรทราบทั้งหมด เช่น การประกาศตัวแปร, การเลือกชนิดข้อมูลที่เหมาะสม, วิธีคำนวณ, การวนลูป, การตัดสไลซ์, วิธีสร้างและใช้ฟังก์ชัน, การกำหนดคลาส, การสร้างออบเจกต์ และอื่น ๆ อีกมากมาย หลังจากอ่านส่วนนี้จบผู้อ่านจะมีความรู้พื้นฐานเกี่ยวกับ JavaScript ครบทุกแง่มุมแล้ว

ส่วนที่ 2 เป็นการนำ JavaScript ไปใช้กับ Node.js ซึ่งช่วยให้นักพัฒนาสามารถสร้างเว็บแอปพลิเคชันในฝั่งเซิร์ฟเวอร์ได้ เช่น การสร้างเว็บเซิร์ฟเวอร์ และการสร้าง REST API

ส่วนที่ 3 เป็นการนำ JavaScript ไปสร้างเว็บแอปพลิเคชันที่ต้องติดต่อกับฐานข้อมูล MongoDB เช่น การเพิ่มเอกสารไปยังฐานข้อมูล การคิวรีเพื่อเลือกเอกสารจากฐานข้อมูล การแก้ไขเอกสาร การลบเอกสาร ฯลฯ และในตอนท้ายยังได้แสดงตัวอย่างแอปพลิเคชันที่ต้องอ่านและเขียนข้อมูลไปยังฐานข้อมูลอีกด้วย

จากเนื้อหาทั้งหมดของหนังสือเล่มนี้ ที่เน้นการเรียนรู้ JavaScript จากโค้ดตัวอย่างจริงจำนวนมาก ผู้เขียนมั่นใจเป็นอย่างยิ่งว่า ผู้อ่านจะมีความรู้เกี่ยวกับ JavaScript ครอบคลุมทุกด้านในระดับที่สามารถนำไปใช้งานจริงได้อย่างแน่นอน

จีราวุธ วารินทร์

jeerawuth@me.com

## Part 1 JavaScript Basic

บทที่ 1	แนะนำ JavaScript	6
บทที่ 2	ติดตั้งเครื่องมือ และทดสอบโค้ด JavaScript	24
บทที่ 3	ตัวแปรและชนิดข้อมูล	48
บทที่ 4	สตริง	68
บทที่ 5	ชนิดข้อมูลแบบ Number	86
บทที่ 6	ฟังก์ชัน	103
บทที่ 7	การตัดสินใจ	131
บทที่ 8	การวนซ้ำ	156
บทที่ 9	พื้นฐานการใช้งานออบเจกต์	177
บทที่ 10	อาร์เรย์	209
บทที่ 11	อาร์เรย์เมธอด	234
บทที่ 12	Scope และ Closure	254
บทที่ 13	คอนสตรัคเตอร์ฟังก์ชันและโปรโตไทป์	276
บทที่ 14	คลาส	287
บทที่ 15	การสืบทอดคลาส (Class Inheritance)	313
บทที่ 16	พื้นฐานการใช้งาน Error Handling	328
บทที่ 17	พื้นฐานเกี่ยวกับ Asynchronous Operation	336
บทที่ 18	การใช้ JavaScript จัดการกับ DOM	354

## Part 2 Essential Node.js

บทที่ 19	สร้างแอปพลิเคชันฝั่ง Backend ด้วย Node.js	384
บทที่ 20	พื้นฐานการใช้งาน Express	409
บทที่ 21	สร้าง REST API โดยใช้ Express	435
บทที่ 22	Authentication และ Authorization	458
บทที่ 23	การใช้งาน EJS	489

## Part 3 MongoDB Briefly

บทที่ 24	ฐานข้อมูล MongoDB	504
บทที่ 25	การควิรีและจัดการฐานข้อมูล MongoDB	529
บทที่ 26	แอปพลิเคชันข้อมูลสินค้า	552

**Part 1**

# JavaScript Basic

# บทที่ 1

## แนะนำ JavaScript

JavaScript (จาวาสคริปต์) เป็นภาษาสคริปต์ที่ใช้กับเว็บเบราว์เซอร์ นักพัฒนาสามารถนำ JavaScript มาสร้างเว็บแอปพลิเคชันทั้งในฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์

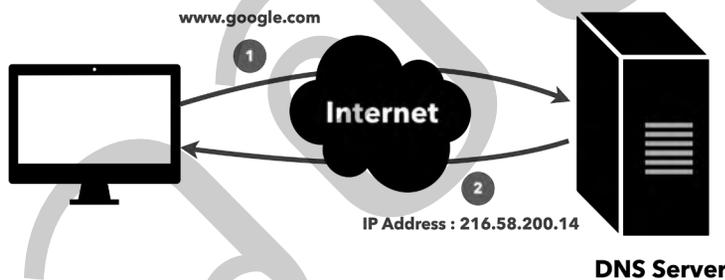
ในบทแรกนี้จะมาทำความรู้จักกับภาษา JavaScript และวิธีการทดสอบโค้ด JavaScript ผ่านทางอินเทอร์เน็ต วิธีนี้มีข้อดีคือ ไม่ต้องติดตั้งอะไรลงไปในเครื่องคอมพิวเตอร์ก็สามารถทดสอบ หรือเรียนรู้การใช้ JavaScript ได้ทันที

### บทวนเกี่ยวกับอินเทอร์เน็ต (Internet)

อินเทอร์เน็ต คือ คอมพิวเตอร์ที่เชื่อมต่อกันเป็นระบบเครือข่ายขนาดใหญ่ที่เชื่อมต่อกันทั่วโลก โดยเรียกคอมพิวเตอร์ที่ทำหน้าที่ให้บริการข้อมูล หรือบริการด้านต่าง ๆ บนอินเทอร์เน็ตว่า Server (เซิร์ฟเวอร์) และเรียกคอมพิวเตอร์ที่เชื่อมต่อไปยังอินเทอร์เน็ต และใช้บริการต่าง ๆ จากอินเทอร์เน็ต ว่า Client (ไคลเอนต์)

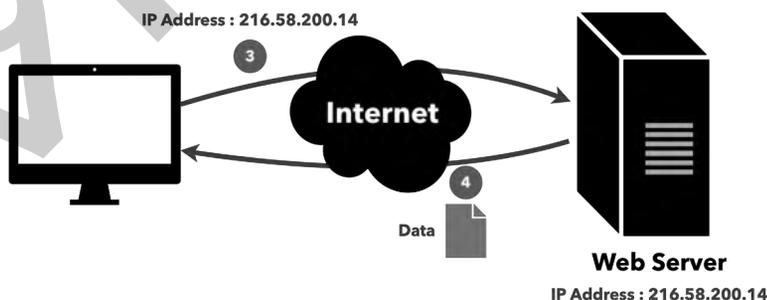
คอมพิวเตอร์และอุปกรณ์ต่าง ๆ ที่เชื่อมต่ออินเทอร์เน็ต จะมีหมายเลข IP Address ที่ไม่ซ้ำกัน การสื่อสารระหว่างคอมพิวเตอร์ที่อยู่ในอินเทอร์เน็ต จะใช้ IP Address ระบุตำแหน่งของผู้รับและผู้ส่งข้อมูลในอินเทอร์เน็ต

เมื่อผู้ใช้ต้องการเปิดดูหน้าเว็บเพจที่อยู่ในอินเทอร์เน็ต จะกรอก URL เช่น thinkbeyondbook.com เพื่อส่งที่อยู่ที่ต้องการติดต่อไปให้กับ DNS Server (Domain Name System Server) เพื่อแปลง URL ให้กลายเป็นหมายเลข IP Address และส่งกลับมายังเบราว์เซอร์



▲ รูปแสดงส่งข้อมูลระหว่างเบราว์เซอร์ และ DNS Server

หลังจากที่เบราว์เซอร์ทราบหมายเลข IP Address ก็ส่งคำร้องขอไปยังเว็บเซิร์ฟเวอร์ตามหมายเลข IP Address ดังกล่าว เซิร์ฟเวอร์ก็จะส่งข้อมูลกลับมาแสดงผลยังเบราว์เซอร์ตามต้องการ



▲ รูปแสดงส่งข้อมูลระหว่างเบราว์เซอร์ และ Web Server

## เบราว์เซอร์ (Web Browser)

เบราว์เซอร์ (Browser) คือ ซอฟต์แวร์สำหรับใช้ท่องอินเทอร์เน็ต เช่น ใช้เบราว์เซอร์เพื่อดูข้อมูลจากเว็บไซต์ ดูหนังฟังเพลงบนเว็บ ตลอดจนเล่นเกมผ่านเว็บ เบราร์เวอร์ถือว่าเป็นซอฟต์แวร์หลักที่มีอยู่ในคอมพิวเตอร์, แท็บเล็ต, สมาร์ทโฟน สมาร์ททีวี และอื่น ๆ

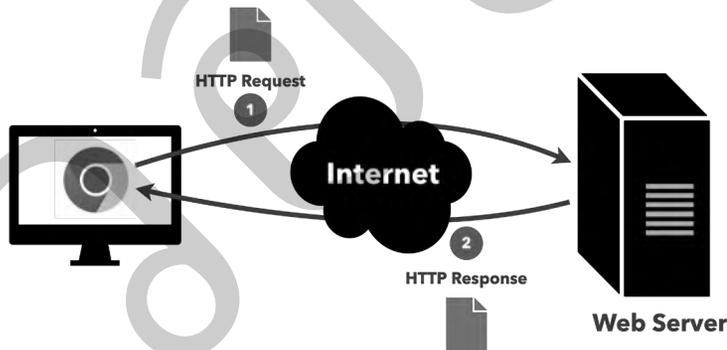
ผู้ใช้สามารถกรอก URL หรือหมายเลข IP Address หรือคลิกลิงก์ในเบราว์เซอร์ เพื่อเปิดเว็บเพจที่ต้องการ โดยเบราว์เซอร์จะใช้หมายเลข IP Address เพื่อส่งการร้องขอแบบต่าง ๆ ไปยังอินเทอร์เน็ต และนำข้อมูลที่ได้รับจากอินเทอร์เน็ตมาใช้งาน

เบราว์เซอร์ที่นิยมใช้งานในปัจจุบัน มีอยู่ด้วยการหลายตัว เช่น Google Chrome, Microsoft Edge, Firefox, Opera, Brave และ Safari

## เว็บเพจ

เว็บเพจ (Webpage หรือ Web page) คือ เอกสารที่ถูกใช้งานบนอินเทอร์เน็ต และถูกแสดงผลบนเบราว์เซอร์ เมื่อผู้ใช้ต้องการเปิดดูข้อมูลจากอินเทอร์เน็ต จะใช้เว็บเบราว์เซอร์ไปยังที่อยู่ของเว็บเพจ (ใช้ IP Address) เพื่อส่งการร้องขอข้อมูลไปยังเว็บเซิร์ฟเวอร์ (เรียกว่า HTTP Request)

เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอ ก็จะส่งผลลัพธ์กลับมายังเบราว์เซอร์ (เรียกว่า HTTP Response) เพื่อนำข้อมูลที่ได้มาใช้ หรือนำมาแสดงผลต่อไป



▲ รูปแสดง HTTP Request และ HTTP Response

เนื้อหาที่อยู่ในเว็บเพจอาจประกอบด้วยข้อความ, ลิงก์, รูปภาพ หรือมีเดียประเภทต่าง ๆ ดังนั้น ข้อมูลเว็บเพจที่ถูกนำมาแสดงผลบนเบราว์เซอร์จึงประกอบไปด้วยไฟล์ประเภทต่าง ๆ มากมาย แต่โดยพื้นฐานแล้ว เว็บเพจจะประกอบด้วยไฟล์หลักอยู่ 3 ประเภท ได้แก่ HTML, CSS และ JavaScript

- HTML คือ ไฟล์ที่ใช้กำหนดโครงสร้างของเว็บเพจ บอกให้ทราบว่าภายในเว็บเพจจะมีเนื้อหา และมีจัดเรียงลำดับอย่างไร ไฟล์ HTML จะมีนามสกุลไฟล์เป็น .htm หรือ .html
- CSS คือ ไฟล์ที่ใช้ตกแต่งหน้าตาเว็บเพจ เช่น กำหนดขนาดและสีตัวอักษร, กำหนดขนาดรูปภาพ, จัดตำแหน่งการแสดงผลบนอุปกรณ์ที่มีขนาดหน้าจอต่างกัน ฯลฯ ไฟล์ CSS จะมีนามสกุลไฟล์เป็น .css

- JavaScript คือ ไฟล์ซึ่งทำให้เว็บเพจสามารถโต้ตอบกับผู้ใช้ได้ หรือทำให้เว็บเพจมีความสามารถบางอย่างเพิ่มขึ้นมา ซึ่งไฟล์ JavaScript จะมีนามสกุลไฟล์เป็น .js

การเขียนโค้ด JavaScript จะมีความสัมพันธ์โดยตรงกับโค้ด HTML และ CSS ที่อยู่ในเว็บเพจ เช่น เขียนโค้ด JavaScript เพื่อแสดงหรือซ่อนเนื้อหาที่อยู่ใน HTML หรือเขียนโค้ด JavaScript เพื่อเลือกใช้งาน CSS สำหรับปรับแต่งหน้าตาเว็บเพจในแบบที่ต้องการ

## ภาษา JavaScript

JavaScript เป็นภาษาคอมพิวเตอร์ ที่ถูกออกแบบมาให้ทำงานบนเบราว์เซอร์ จุดประสงค์เพื่อให้เว็บเพจสามารถโต้ตอบกับผู้ใช้ได้ หรือมีความสามารถบางอย่างเพิ่มขึ้น เช่น สามารถใช้ JavaScript เพื่อขยายรูปภาพเมื่อผู้ใช้เลื่อนเมาส์ไปที่รูปภาพ, ใช้ JavaScript เพื่อสร้างแอนิเมชันเลื่อนข้อความจากซ้ายไปขวา, ใช้ JavaScript เพื่อนำเนื้อหาใหม่มาแทนที่เนื้อหาเดิมโดยไม่ต้องโหลดเว็บเพจใหม่ เป็นต้น

JavaScript เป็นภาษาคอมพิวเตอร์ที่ออกแบบมาสำหรับใช้กับเว็บเพจ ดังนั้น โค้ด JavaScript จึงถูกรันเพื่อใช้งานบนเบราว์เซอร์เป็นหลัก นอกจากนั้น JavaScript ยังสามารถใช้กับสภาพแวดล้อมอื่น ๆ ที่ไม่ใช่เบราว์เซอร์ได้อีกด้วย เช่น ใช้ JavaScript กับ Node.js และ Apache CouchDB เป็นต้น



▲ รูปแสดงตัวอย่างโค้ดคำสั่ง JavaScript

## ECMAScript, ES6, ES2015 และ ES2025

ในอดีตเบราว์เซอร์แต่ละตัว ไม่ว่าจะเป็น Internet Explorer, Firefox, Safari, Opera หรือ Google Chrome จะใช้ JavaScript เพื่อทำให้เบราว์เซอร์ของคุณมีความสามารถเพิ่มขึ้น โดยได้เพิ่มฟีเจอร์ใหม่ ๆ ลงไป (เช่น ตรวจสอบข้อมูลที่กรอกลงในแบบฟอร์ม) แต่ด้วยการที่แต่ละเบราว์เซอร์มีการใช้ JavaScript แยกจากกัน จึงทำให้คุณสมบัติของเบราว์เซอร์ของแต่ละรุ่นจึงแตกต่างกันด้วย

ปัญหาที่เกิดขึ้น คือ เมื่อแต่ละเบราว์เซอร์มีคุณสมบัติต่างกัน นักพัฒนาที่ต้องการสร้างแอปพลิเคชันต้องตรวจสอบชนิดและเวอร์ชันของเบราว์เซอร์ และเขียนโค้ด JavaScript ด้วยรูปแบบที่แตกต่างกัน เพื่อให้สามารถทำงานได้กับเบราว์เซอร์ที่มีอยู่ในตลาด ณ ขณะนั้นได้ทุกตัว

เมื่อไม่มีมาตรฐานกลางที่แน่นอน การใช้งาน JavaScript จึงพบปัญหามากมาย ดังนั้น เพื่อแก้ปัญหาดังกล่าว จึงต้องกำหนดมาตรฐานกลางขึ้น เพื่อให้เบราว์เซอร์แต่ละรายนำ JavaScript ไปใช้งานด้วยแนวทางเดียวกัน

ECMA (เอ็กม่า) คือ องค์กรที่ถูกจัดตั้งขึ้น เพื่อกำหนดมาตรฐานให้กับเทคโนโลยีด้านต่าง ๆ โดยหนึ่งในมาตรฐานสำคัญที่ ECMA ได้กำหนดมาตรฐานขึ้นมา คือ ECMAScript (เลขที่ ECMA-262)

ECMAScript (เอ็กม่าสคริปต์) เป็นมาตรฐานสำหรับใช้กับภาษาสคริปต์ โดยทาง ECMA ได้กำหนดเอกสารเพื่อบอกรายละเอียดว่าภาษาคอมพิวเตอร์ที่จะทำตามข้อกำหนดของ ECMAScript นั้นมีอะไรบ้าง เช่น การประกาศตัวแปรต้องใช้คีย์เวิร์ดใด มีชนิดข้อมูลที่ประเภท หรือมีโครงสร้างทางภาษาเป็นอย่างไร



▲ รูปแสดงเอกสารซึ่งอธิบายข้อกำหนดของ ECMAScript

ECMAScript เวอร์ชันแรก (ECMAScript 1 หรือเรียกย่อ ๆ ว่า ES1) ได้ถูกประกาศออกมาในปี ค.ศ. 1997 โดย JavaScript ถือเป็นภาษาคอมพิวเตอร์ที่พัฒนาขึ้นตามข้อกำหนดที่ได้รับไว้ใน ECMAScript

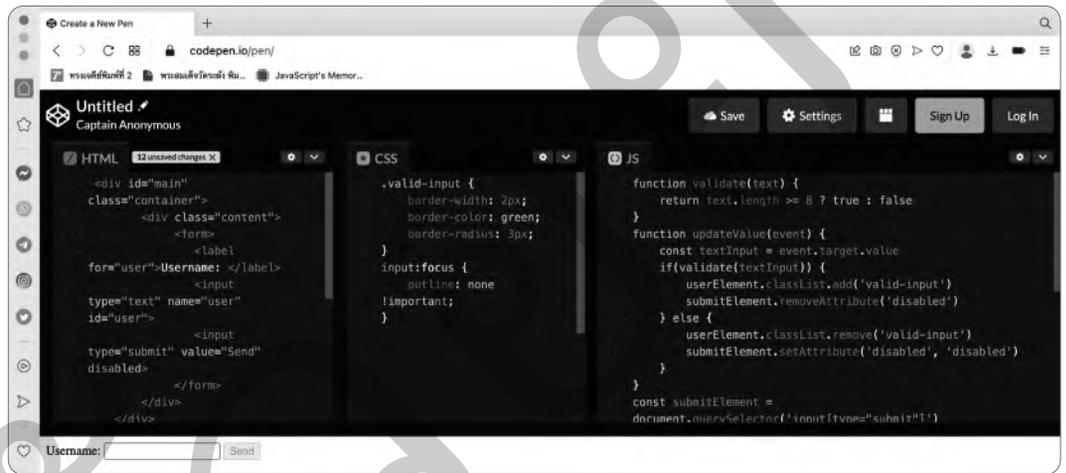
ด้วยการที่เทคโนโลยีมีการเปลี่ยนแปลงอยู่เสมอ ECMAScript จึงมีการปรับปรุงอยู่หลายครั้ง เช่น ES1, ES2, ES3 และ ES5 ตามลำดับ ต่อมาในปี ค.ศ. 2015 ก็มีการปรับปรุง ECMAScript ครั้งใหญ่ โดยออกเวอร์ชัน ES6 พร้อมทั้งเปลี่ยนชื่อเวอร์ชันจาก ES6 ให้กลายเป็น ES2015

หลังจากนั้น ECMAScript ได้ถูกอัปเดตในทุก ๆ ปี โดยใช้เลขปี ค.ศ. แทนชื่อเวอร์ชันในแบบเดิม เช่น ES2016, ES2017, ES2018, ES2019, ES2020, ES2021, ES2022, ES2023, ES2024 และ ES2025 ตามลำดับ

## ทดสอบ JavaScript ในแบบออนไลน์

การศึกษา JavaScript สามารถทำได้หลายช่องทาง ซึ่งวิธีหนึ่งที่นิยมเพิ่มขึ้นมากในปัจจุบัน คือ การศึกษา JavaScript ผ่านระบบออนไลน์ ผู้อ่านสามารถรอกชุดคำสั่งภาษา JavaScript ลงในเบราว์เซอร์ ผลลัพธ์การรันชุดคำสั่งก็จะปรากฏบนเบราว์เซอร์ได้ทันที

เว็บไซต์ที่สามารถทดสอบ JavaScript ในแบบออนไลน์ มีอยู่ด้วยกันหลายเว็บไซต์ เช่น <https://replit.com>, <https://codepen.io> และ <https://playcode.io> เป็นต้น



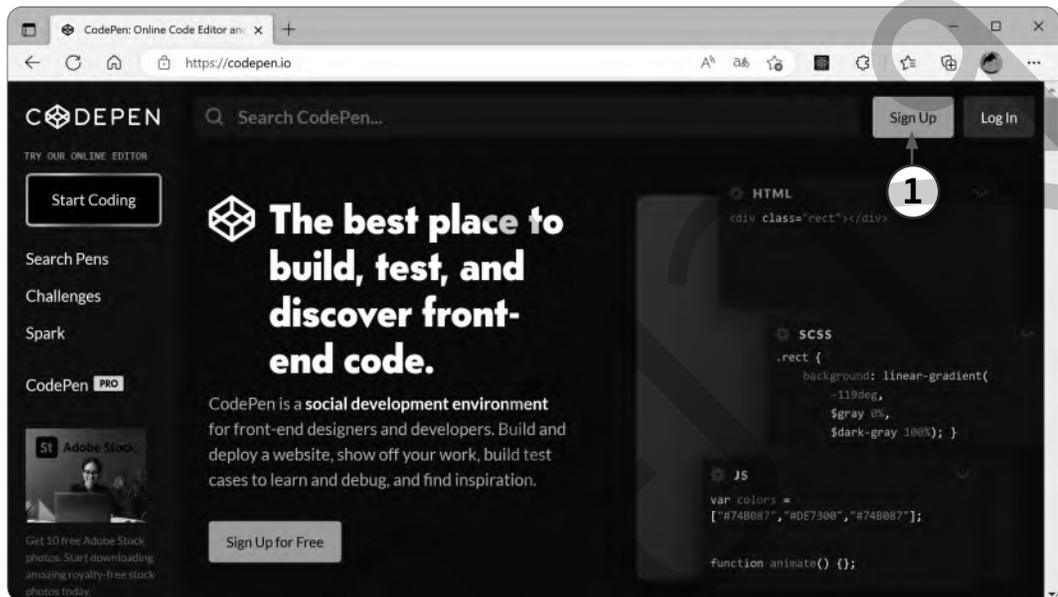
▲ รูปแสดงตัวอย่างการทดสอบโค้ด JavaScript บนเบราว์เซอร์

ในบทแรกนี้จะเริ่มต้นศึกษา JavaScript ผ่านทาง CodePen (ที่แอดเดรส <https://codepen.io>) โดยเหตุผลที่เลือกทดสอบ JavaScript ผ่านทาง CodePen เนื่องจากเป็นระบบออนไลน์ที่ไม่ต้องติดตั้งอะไรให้ยุ่งยาก, สามารถบันทึกโค้ดไว้ดูในภายหลัง, สามารถแชร์โค้ดคำสั่งไปให้คนอื่น ๆ ได้

## ลงทะเบียนกับ <https://codepen.io>

ก่อนทดสอบโค้ด JavaScript กับ CodePen จะต้องสมัครสมาชิกก่อน เพื่อให้มีโปรไฟล์สำหรับเก็บโค้ดที่ได้เขียนไว้ และสามารถแชร์โค้ดไปให้สมาชิกคนอื่น ๆ สามารถนำโค้ดไปใช้งานได้

1. เปิดเบราว์เซอร์ และไปยังเว็บไซต์ <https://codepen.io> แล้วคลิกปุ่ม Sign Up เพื่อสมัครสมาชิก



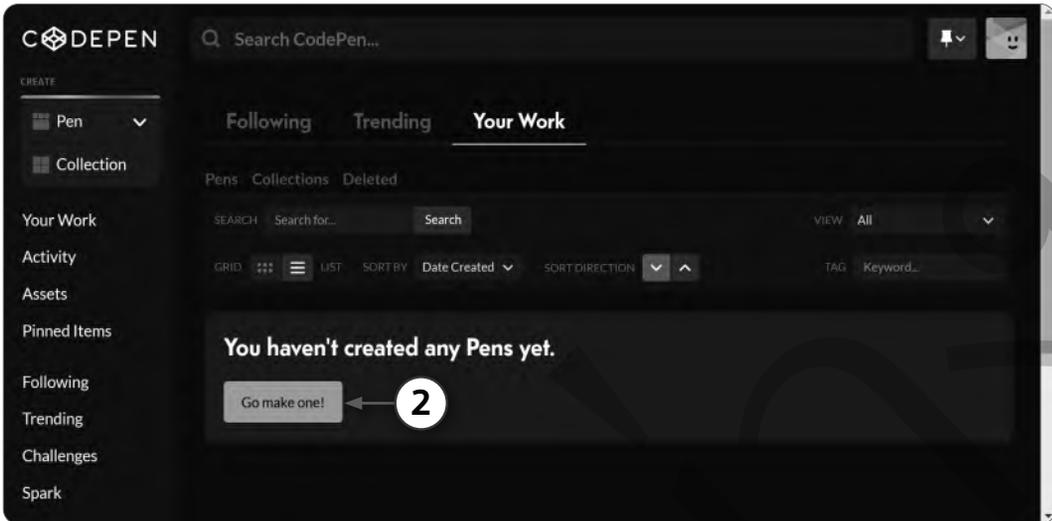
2. ทำตามขั้นตอนต่าง ๆ ตามที่ทางเว็บไซต์แนะนำ เช่น ถ้าสมัครผ่านอีเมลจะต้องเข้าไปคลิกลิงก์ในอีเมลเพื่อยืนยันการสมัครด้วย จากนั้นให้เข้าสู่ระบบให้เรียบร้อย

## โค้ดแรกกับ JavaScript

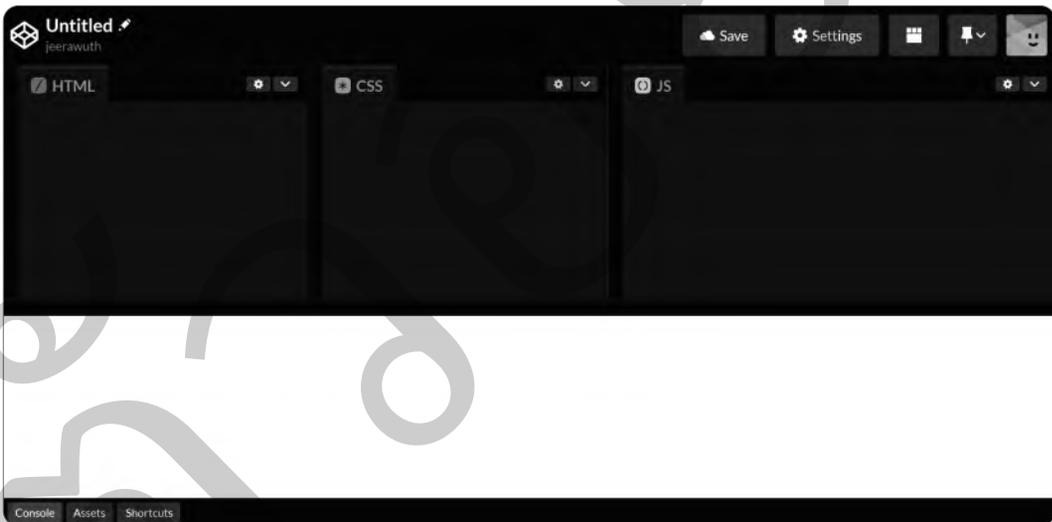
คำสั่งแรกของ JavaScript ที่ต้องทราบคือ คำสั่ง `console.log()` ใช้เพื่อพิมพ์ข้อความออกมาที่หน้าจอ เช่น เมื่อต้องการพิมพ์ข้อความ "Hello JavaScript" ก็ให้ใช้คำสั่ง `console.log("Hello JavaScript")` เป็นต้น

1. เปิดเบราว์เซอร์และไปยัง <https://codepen.io> และล็อกอินเข้าสู่ระบบให้เรียบร้อย จากคลิกเมนู Your Work เพื่อแสดงพื้นที่เก็บงานที่เคยได้ทำเอาไว้ (ในการใช้งานครั้งแรกจะเป็นพื้นที่ว่าง ๆ ยังไม่มีข้อมูลอะไรอยู่)

2. คลิกปุ่ม Go make one! เพื่อเริ่มลงมือเขียนโค้ด



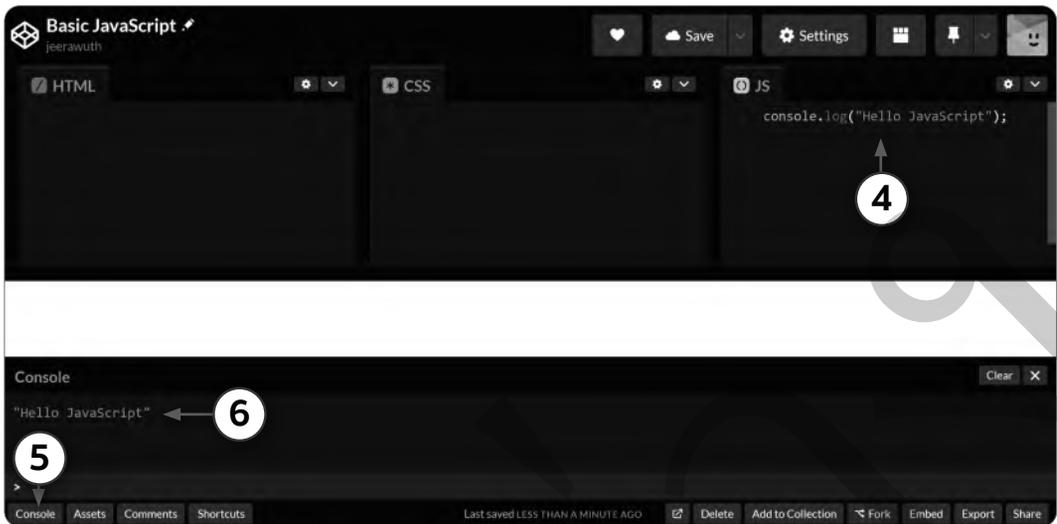
3. ระบบจะกำหนดหน้าต่างขึ้นมาจำนวน 4 หน้าต่าง ด้านบนเรียงจากซ้ายไปขวา คือ หน้าต่าง HTML, CSS และ JS ส่วนด้านล่าง คือ หน้าต่างที่ใช้แสดงผลลัพธ์



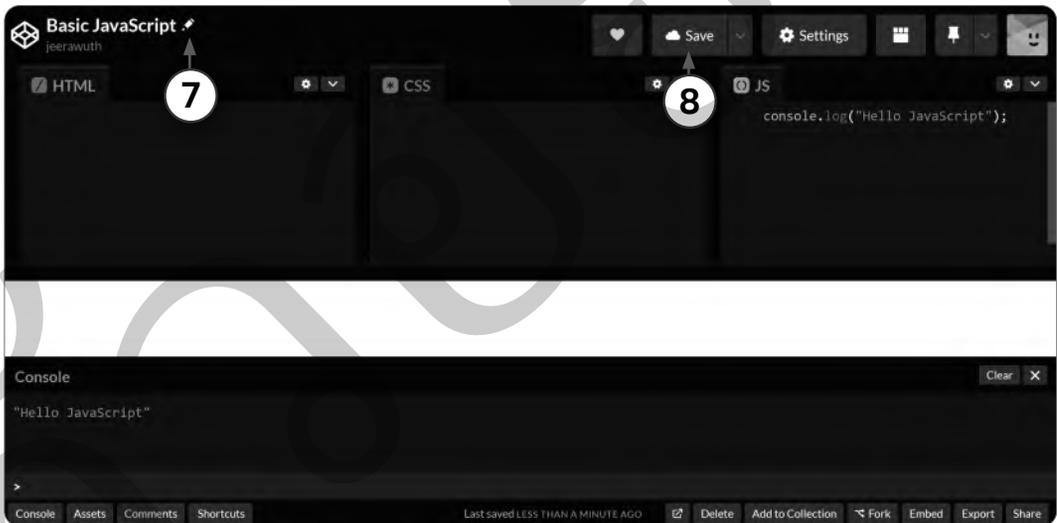
4. ที่หน้าต่าง JS ให้กรอกโค้ดภาษา JavaScript ลงไปดังนี้

```
console.log("Hello JavaScript");
```

5. คลิกปุ่ม **Console** เพื่อแสดงหน้าต่างคอนโซล
6. ผลลัพธ์คำว่า Hello JavaScript จะถูกแสดงที่หน้าต่างคอนโซล ซึ่งเกิดจากคำสั่งที่ได้พิมพ์ไว้ในขั้นตอนที่ 4



7. สามารถตั้งชื่อได้ โดยคลิกปุ่มรูปดินสอ  และแก้ไขชื่อตามต้องการ เช่น Basic JavaScript
8. คลิกปุ่ม Save เพื่อบันทึกโค้ด



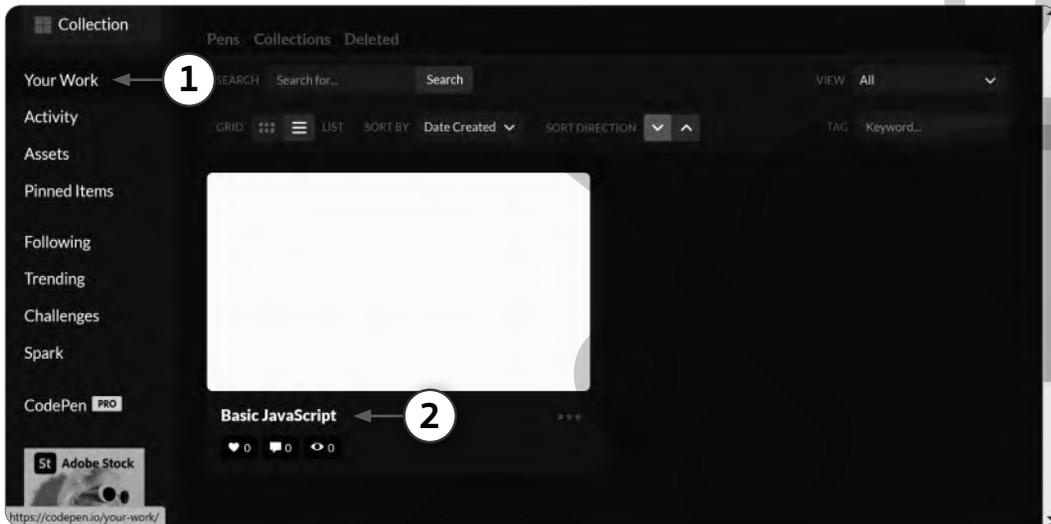
## Note

โดยปกติเมื่อมีการบันทึกโค้ดครั้งแรกไปแล้ว ถ้ามีการแก้ไขโค้ด CodePen ก็จะบันทึกโค้ดให้อัตโนมัติ ซึ่งผู้ใช้สามารถยกเลิกการบันทึกอัตโนมัตินี้ได้โดยคลิกลูกศรข้างปุ่ม Save และแก้ไขตัวเลือกตามต้องการ

## แก้ไขโค้ดเดิมที่เคยกรอกไว้

หลังจากได้ทดลองเขียนโค้ด JavaScript กับ CodePen และบันทึกโค้ดเก็บไว้ ในครั้งต่อไปสามารถสร้างโค้ดชิ้นใหม่ หรือเลือกโค้ดที่เคยสร้างไว้ก็ได้ ดังนี้

1. คลิกที่ Your Work
2. จะปรากฏรายการที่เคยทำเอาไว้ ให้คลิกเลือกโค้ดที่ต้องการใช้งาน (ในตัวอย่างให้คลิกที่ Basic JavaScript)



## ตัวแปร (Variable)

ตัวแปร (Variable) คือ ชื่อที่ใช้อ้างอิงไปยังพื้นที่ในหน่วยความจำ เพื่อใช้งานข้อมูลที่อยู่ในหน่วยความจำนั้น ๆ เช่น กำหนดชื่อ PI เพื่อใช้อ้างอิงค่าตัวเลข 3.14159 หรือกำหนด my\_box เพื่ออ้างอิงไปยังตัวเลข 10 เป็นต้น การกำหนดค่าให้กับตัวแปร จะเริ่มจากใส่คีย์เวิร์ด var จากนั้นให้กำหนดชื่อตัวแปรไว้ด้านซ้าย ตามด้วยเครื่องหมายเท่ากับ (=) และกำหนดค่าที่ต้องการกำหนดให้กับตัวแปร

```
var variable_name = data;
```

- var คือ คีย์เวิร์ดสำหรับใช้ประกาศตัวแปร
- variable\_name คือ ชื่อของตัวแปร
- data คือ ข้อมูลหรือออบเจกต์ที่ต้องการกำหนดให้กับตัวแปร
- ปิดท้ายคำสั่งด้วยเครื่องหมาย ; (Semi Colon)

## ตัวอย่าง 1.1

### การประกาศตัวแปร

ต่อไปนี้เป็นตัวอย่างการประกาศตัวแปร (ตั้งชื่อตัวแปร) และกำหนดค่าให้กับตัวแปร

```
1 var first_name = "Sombat";
2 var age = 14;
3 age = 40;
```

- บรรทัดที่ 1 ประกาศตัวแปรโดยใช้ชื่อว่า first\_name และกำหนดค่าด้วยข้อความ "Sombat"
- บรรทัดที่ 2 ประกาศตัวแปรโดยใช้ชื่อว่า age และกำหนดค่าตัวแปร age ด้วยเลข 14
- บรรทัดที่ 3 กำหนดค่าตัวแปร age ด้วยตัวเลข 40

## อินพุตและเอาต์พุต

แอปพลิเคชันหรือโปรแกรมจะมีพื้นฐานที่สำคัญอยู่ 2 ส่วน คือ อินพุต (Input) และเอาต์พุต (Output) โดยอินพุตคือการใช้โปรแกรมส่งข้อมูลบางอย่างเข้าไปยังโปรแกรม (เช่น กรอกข้อความ คลิกปุ่ม เลื่อนเมาส์ ฯลฯ) ส่วนเอาต์พุต คือ ข้อมูลบางอย่างที่ส่งออกมาจากโปรแกรม (เช่น การแสดงข้อความ การแสดงปุ่ม รูปภาพ ฯลฯ)



▲ รูปแสดงอินพุตและเอาต์พุต

การพัฒนาแอปพลิเคชันจะต้องกำหนดอินพุต เอาต์พุตรูปแบบต่าง ๆ เพื่อใช้ติดต่อสื่อสารระหว่างโปรแกรมกับผู้ใช้ ซึ่งอินพุตขั้นพื้นฐาน ได้แก่ การพิมพ์ข้อความโดยใช้คีย์บอร์ดส่งเข้าไปยังโปรแกรม ส่วนเอาต์พุตขั้นพื้นฐานคือการแสดงข้อความออกมาที่หน้าจอ

## แสดงหน้าต่างโต้ตอบด้วยคำสั่ง alert

ใน JavaScript หากต้องการแสดงข้อความออกมาที่หน้าต่างโต้ตอบ (เป็นการกำหนดเอาต์พุต) จะใช้คำสั่ง alert ซึ่งการใช้ alert ใน JavaScript มีรูปแบบพื้นฐานเป็นดังนี้

```
alert(message?);
```

- **alert** คือ ฟังก์ชันที่ใช้แสดงหน้าต่างโต้ตอบของเบราว์เซอร์
- **message** คือ ข้อความ (หรือค่าจากตัวแปร) ที่ต้องการนำมาแสดงบนหน้าจอ (message? มีความหมายว่า ค่า message ที่กำหนดลงไปนั้น สามารถกำหนดค่าหรือไม่กำหนดก็ได้) โดยข้อความนี้จะต้องอยู่ภายในเครื่องหมาย Double quote หรือ Single quote เช่น "กรุณากรอกอายุ" หรือ 'กรุณากรอกอายุ' ตามลำดับ

### ตัวอย่าง 1.2

#### การใช้งานฟังก์ชัน alert() และ console.log()

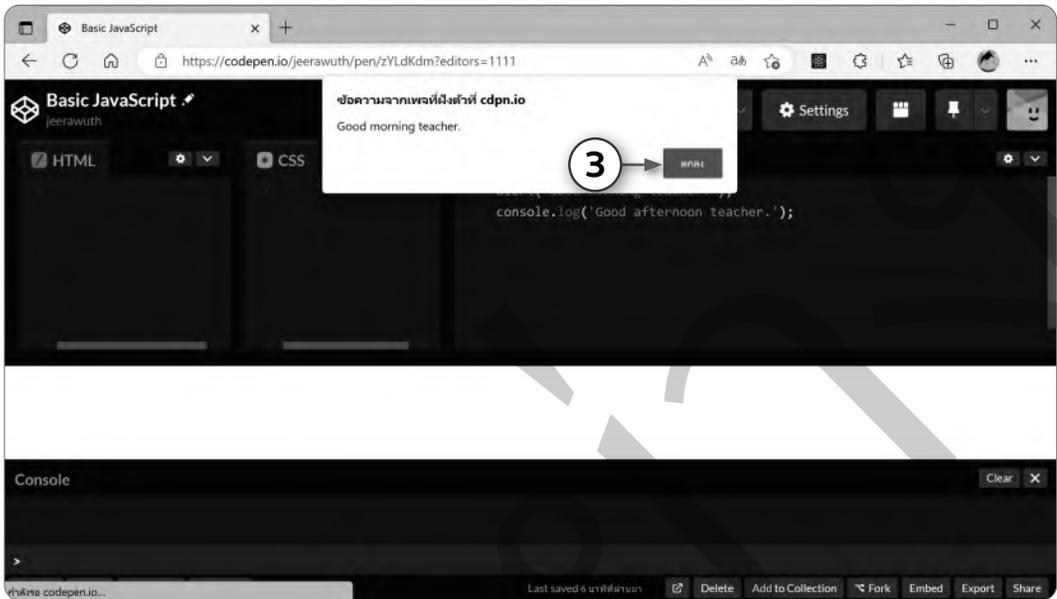
ตัวอย่างการแสดงผลข้อความในหน้าต่างโต้ตอบ เปรียบเทียบกับการแสดงข้อความที่หน้าต่างคอนโซลของเบราว์เซอร์

1. กรอกโค้ดคำสั่งเป็นดังต่อไปนี้

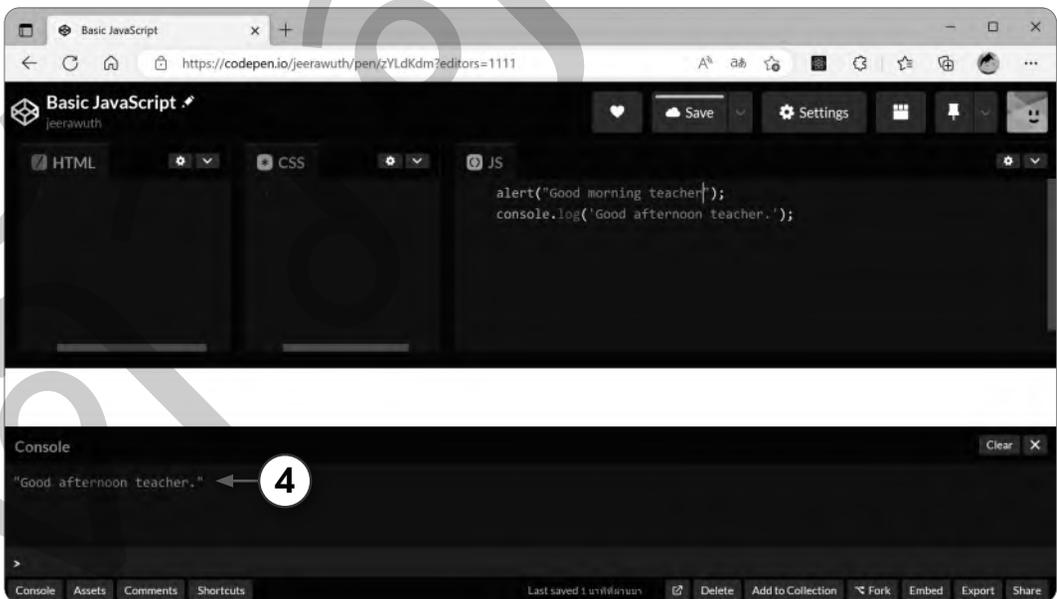
```
1 alert("Good morning teacher.");
2 console.log('Good afternoon teacher.');
```

- บรรทัดที่ 1 ใช้คำสั่ง alert แสดงหน้าต่างโต้ตอบ โดยกำหนดข้อความ "Good morning teacher." สังเกตว่าเราใช้เครื่องหมาย " " ครอบข้อความ
  - บรรทัดที่ 2 ใช้คำสั่ง console.log() เพื่อพิมพ์ข้อความออกมา โดยกำหนดข้อความเป็น 'Good afternoon Teacher' สังเกตว่าสามารถใช้เครื่องหมาย ' ' ครอบข้อความแทนการใช้ " " ก็ได้
2. โดยปกติ CodePen จะบันทึกโค้ดให้โดยอัตโนมัติทุกครั้งที่มีการแก้ไข แต่ถ้าต้องการบันทึกด้วยตนเอง ให้คลิกปุ่ม Save

3. จะพบหน้าต่างโต้ตอบแสดงข้อความ "Good morning teacher." ให้คลิกปุ่ม ตกลง เพื่อปิดหน้าต่างโต้ตอบ



4. ที่หน้าต่างคอนโซลจะพบข้อความ 'Good afternoon teacher.' (ถ้าไม่พบหน้าต่างคอนโซลให้คลิกปุ่ม Console เพื่อสั่งให้แสดงหน้าต่างคอนโซล)



## รับข้อความที่เป็นอินพุตด้วยคำสั่ง prompt

หากต้องการส่งอินพุตที่เป็นข้อความเข้าไปยังเบราว์เซอร์ สามารถใช้ฟังก์ชัน `prompt()` ซึ่งการใช้ `prompt` ใน JavaScript มีรูปแบบพื้นฐาน เป็นดังต่อไปนี้

```
var variable = prompt(message?);
```

- **variable** คือ ตัวแปรที่ใช้เก็บข้อความซึ่งผู้ใช้ได้กรอกลงไป
- **prompt** คือ ฟังก์ชันที่ใช้รับข้อความที่ผู้ใช้กรอกจากคีย์บอร์ด
- **message** คือ ข้อความที่ต้องการนำมาแสดงบนหน้าจอ เพื่อบอกจุดประสงค์ว่าจะนำข้อความที่ผู้ใช้กรอกไปทำอะไร โดยข้อความจะต้องอยู่ภายในเครื่องหมาย Double quote หรือ Single quote เช่น "กรุณากรอกอายุ" หรือ 'กรุณากรอกอายุ' ตามลำดับ

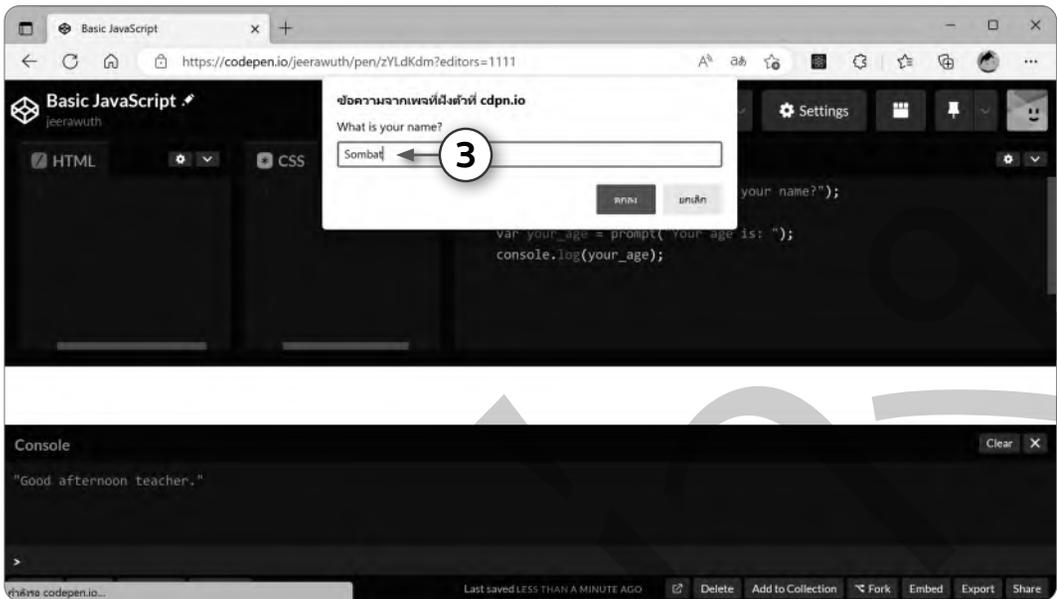
## ตัวอย่างการใช้งานฟังก์ชัน prompt

ตัวอย่างการใช้งานฟังก์ชัน `prompt()` เช่น สั่งให้แสดงข้อความถามชื่อผู้ใช้งาน "What is your name?" เมื่อผู้ใช้กรอกชื่อแล้วกดปุ่ม <Enter> ชื่อของผู้ใช้ก็จะถูกกำหนดให้กับตัวแปร `your_name`

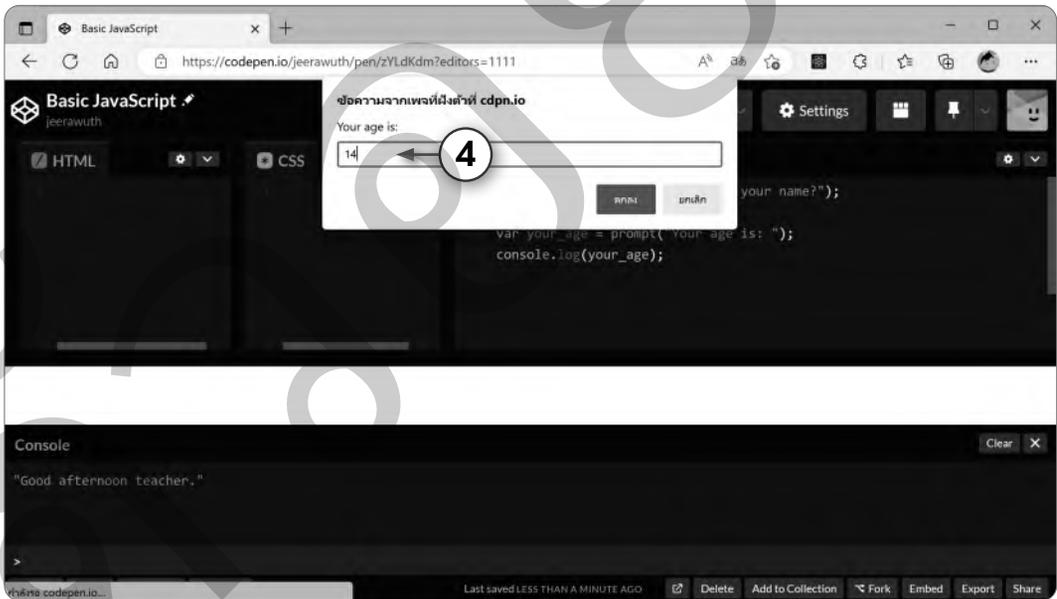
1. ไปยังหน้าต่าง JS ของ CodePen
2. กรอกโค้ดคำสั่งเป็นดังต่อไปนี้

```
1 var your_name = prompt("What is your name?");
2 console.log(your_name);
3 var your_age = prompt("Your age is: ");
4 console.log(your_age);
```

- บรรทัดที่ 1 ใช้คำสั่ง `prompt()` รับข้อความจากผู้ใช้ เมื่อผู้ใช้กรอกชื่อแล้วกดปุ่ม <Enter> ชื่อที่เป็นข้อความจะถูกนำมาเก็บยังตัวแปร `your_name`
  - บรรทัดที่ 2 ใช้คำสั่ง `console.log()` แสดงข้อความที่อยู่ในตัวแปร `your_name` ออกมา
  - บรรทัดที่ 3 ใช้คำสั่ง `prompt()` รับข้อความจากผู้ใช้ เมื่อผู้ใช้กรอกอายุแล้วกดปุ่ม <Enter> อายุที่ผู้ใช้กรอกจะถูกนำมาเก็บยังตัวแปร `your_age` สังเกตว่า แม้ว่าอายุจะเป็นตัวเลข แต่การกรอกตัวเลขด้วยคีย์บอร์ด ตัวเลขนี้จะถือว่าเป็นข้อความชนิดหนึ่ง (ไม่ใช่ตัวเลขที่จะนำมาคำนวณค่าได้)
  - บรรทัดที่ 4 ใช้คำสั่ง `console.log()` แสดงข้อความที่อยู่ในตัวแปร `your_age` ออกมา
3. หลังจากบันทึกโค้ด จะปรากฏหน้าต่าง "What is your name?" ให้กรอกชื่อ (เช่น Sombat) แล้วคลิกปุ่ม ตกลง



4. จะปรากฏหน้าต่าง "Your age is: " ให้กรอกอายุ (เช่น 14) แล้วกดปุ่ม <Enter>



5. ชื่อที่ใช้กรอก จะถูกนำมาแสดงที่หน้าจอ (ในตัวอย่างแสดงข้อความ Sombat)

# ใช้ JavaScript เพียงภาษาเดียว

สร้างได้ทั้ง Frontend และ Backend

ด้วยเนื้อหาที่แน่นแต่กระชับ

ปรับแต่งให้เข้าใจได้ง่าย

อ่านและทำตามได้แบบ

## Step-By-Step

เก่งได้ แม้ไม่มีพื้นฐานมาก่อน



- สรุปพื้นฐาน JavaScript ที่จำเป็นต้องทราบ
- อธิบาย JavaScript ด้วยโค้ดตัวอย่าง พร้อมคำอธิบายในทุกขั้นตอน
- ทดสอบโค้ด JavaScript ผ่านทางระบบออนไลน์ และการใช้ Visual Studio Code
- พื้นฐานเกี่ยวกับ Scope และ Closure
- วิธีใช้งานอาร์เรย์และอาร์เรย์เมธอดโดยละเอียด
- การสร้างออบเจกต์ด้วยวิธีต่าง ๆ การใช้เมธอด ฟิลด์ และพร็อพเพอร์ตี้
- ความแตกต่างระหว่าง Factory Function, Constructor Function และการใช้ Class

- วิธีกำหนดเมธอดและพร็อพเพอร์ตี้ไปยัง Prototype
- ตัวอย่างการใช้ JavaScript จัดการกับ DOM
- สรุปพื้นฐาน Node.js พร้อมตัวอย่างการสร้างแอปพลิเคชันในฝั่งเซิร์ฟเวอร์
- ตัวอย่างการอ่านเขียนไฟล์ใน Node.js
- การสร้าง REST API โดยใช้ Express
- พื้นฐานและตัวอย่างการใช้งาน Authentication และ Authorization
- การเข้ารหัสข้อมูล (Encrypt) และถอดรหัสข้อมูล (Decrypt)
- วิธีใช้งาน Template Engine ใน Node.js
- การใช้งานฐานข้อมูล MongoDB พร้อมตัวอย่างแอปพลิเคชัน

```
EXPLORE  # app.js
- MONGODBPROJECT
  > inquiry
  > ms
  > purypcode
  > sasiprap
  > sift
  > smart-buffer
  > socks
  > sparse-bitfield
  > strnum
  > trad
  > trilb
  > @types
  > uuid
  > uuid-conversions
  > whatwg-url
  [!] package-lock.json
  # app.js
  [!] package.json
  [!] package-lock.json
  > OUTLINE

# app.js
1 mongoose.set('strictQuery', false);
2 mongoose.connect(connectString)
3   .then(() => {
4     console.log('Connection Success!');
5   })
6   .catch(err => {
7     console.log(err);
8   })
9
10 const productSchema = {
11   productName: { type: String, required: true },
12   price: { type: Number, required: true },
13   pictures: { type: String },
14   onstock: { type: Boolean, default: false },
15   stock: { type: Number, default: 0 }
16 };
17
18 const Product = mongoose.model('Product', productSchema);
19
20 Product.create({
21   productName: 'First Product1000',
22   price: 100000
23 })
24   .then(() => {
25     console.log('Add Product OK!');
26   })
27   .catch(err => {
28     console.log(err);
29   });
30
31
32
```



ซื้อสะดวก ส่งถึงบ้านที่ Shopee และ Lazada หรือผ่านทาง  
ร้านหนังสือออนไลน์ [www.thinkbeyondbook.com](http://www.thinkbeyondbook.com)



thinkbeyond books

ISBN (eBook) 885-909-931-022-2



8 85 9099 31 022 2

ราคา 550 บาท