



Software Park Thailand  
</Code Camp>

# Javascript Features

---

Pavit Pimchanagul





## OUTLINE

### 1. Introduction to JS

- What's javascript & Feature
- Javascript History
- Web History
- Technical Documentation
- How to run javascript with HTML

*cont'd...*



Software Park Thailand  
</Code Camp>



## OUTLINE

### 2. Some Javascript Core Feature (Basic)

- DataType
- Variable
- Built in function / procedure
- Function

### 3. Some Javascript Core Feature (Intermediate)

- Object
- Property and Method
- Prototyped based
- Built in Object & Complex Data type

### 4. Building box of javascript program





Software Park Thailand  
</Code Camp>

# Introduction to JS

*JavaScript is a fun language  
with an incredible amount of possibilities.*





## OUTLINE

- What's javascript & Feature
- Javascript & Web History
- How to run javascript with HTML



# What's Javascript

JS

**Dynamic Type**

High Level

Garbage  
Collected

**Single Thread**

Script Language

**Interpreter**

**Runtime**

JIT Complied

Event-Driven

Multi Paradigm

**Imparative**

Procedural

**Functional**

OOP



**Prototype Based**

# Javascript Features (1/3)



Variables



Functions



Data Types



Control flow



Dates



Arrays



Loops



Objects



Operators



Scope

## Javascript Features (2/3)



Callbacks



Promises



Async Await



Event loops



Inheritance



Regular  
Expressions



Prototyping

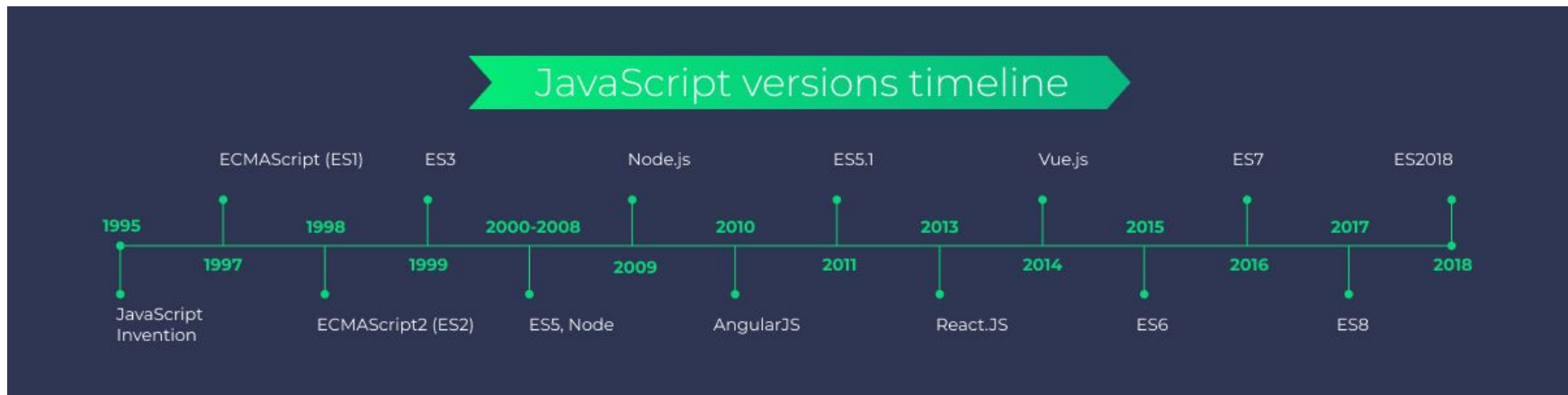


Generators



\_\_\_\_\_

# History



# History (Standard)

Edition	Date published	Name	Changes from prior edition	Editor
1	June 1997		First edition	Guy L. Steele Jr.
2	June 1998		Editorial changes to keep the specification fully aligned with ISO/IEC 16262 international standard	Mike Cowlishaw
3	December 1999		Added <a href="#">regular expressions</a> , better string handling, new control statements, try/catch exception handling, tighter definition of errors, formatting for numeric output and other enhancements	Mike Cowlishaw
4	<i>Abandoned</i>		Fourth Edition was abandoned, due to political differences concerning language complexity. Many features proposed for the Fourth Edition have been completely dropped; some were incorporated into the sixth edition.	
5	December 2009		Adds "strict mode," a subset intended to provide more thorough error checking and avoid error-prone constructs. Clarifies many ambiguities in the 3rd edition specification, and accommodates behaviour of real-world implementations that differed consistently from that specification. Adds some new features, such as getters and setters, library support for <a href="#">JSON</a> , and more complete <a href="#">reflection</a> on object properties. <sup>[11]</sup>	Pratap Lakshman, Allen Wirfs-Brock
5.1	June 2011		This edition 5.1 of the ECMAScript standard is fully aligned with third edition of the international standard ISO/IEC 16262:2011.	Pratap Lakshman, Allen Wirfs-Brock
6	June 2015 <sup>[12]</sup>	ECMAScript 2015 (ES2015)	See <a href="#">6th Edition - ECMAScript 2015</a>	Allen Wirfs-Brock
7	June 2016 <sup>[13]</sup>	ECMAScript 2016 (ES2016)	See <a href="#">7th Edition - ECMAScript 2016</a>	Brian Terlson
8	June 2017 <sup>[14]</sup>	ECMAScript 2017 (ES2017)	See <a href="#">8th Edition - ECMAScript 2017</a>	Brian Terlson
9	June 2018 <sup>[15]</sup>	ECMAScript 2018 (ES2018)	See <a href="#">9th Edition - ECMAScript 2018</a>	Brian Terlson
10	June 2019 <sup>[10]</sup>	ECMAScript 2019 (ES2019)	See <a href="#">10th Edition - ECMAScript 2019</a>	Brian Terlson, Bradley Farias, Jordan Harband

# History

- **JavaScript** ถูกพัฒนาโดย Netscape Communication เปิดตัวครั้งแรกในปี 1995
- เพื่อให้ JavaScript มีมาตรฐานมากขึ้น ทาง Netscape จึงส่ง JavaScript ให้องค์กรจัดการกำหนดมาตรฐานด้านการจัดการข้อมูล เทคโนโลยีการสื่อสาร และการใช้งานอิเล็กทรอนิกส์ หรือที่เรียกว่า **ECMA** จัดการกำหนดมาตรฐานภาษา script ขึ้น
- ECMA กำหนดมาตรฐานรุ่นแรกมาเรียกว่า ECMASCRIPT หรือ **ES**

# History (ES2015)

- **ES6** คือ มาตรฐานภาษา scripting ECMAScript รุ่นที่ 6 ซึ่งถูกปล่อยออกมาในปี **2015**
- เป็นรุ่นที่นิยมใช้มากที่สุดในปัจจุบัน เนื่องจากเป็นรุ่นที่มีการเพิ่ม features ต่างๆเข้ามามากมาย และเกิดการพัฒนาที่ทำให้เห็นการเปลี่ยนแปลงอย่างมากเมื่อเทียบกับรุ่นอื่นก่อนหน้านี้
- อีกหลายรุ่นที่ตามมาในภายหลังก็ไม่ได้เปลี่ยนแปลงไปจาก ES6 มากนัก Dev ส่วนใหญ่ขนานนาม JavaScript รุ่นนี้ว่า **Modern JavaScript**

# Web History ( 1.0 : Static Site)



# Web History ( 2.0 : Web Application)

JavaScript ใช้เพิ่มเติมลูกเล่นในการแสดงผลบนเว็บเพจ เช่น คลิกเมาส์ การตรวจสอบการกรอกข้อมูลในฟอร์ม เป็นต้น



# Web History ( 2.0 : Web Application)



Loading...

Log in



Email Need an account? [Sign up](#)

Password Show

[Log in](#)

[Forgot password?](#)



Log in

Email Need an account? [Sign up](#)

Password Show

Log in

[Forgot password?](#)

Help

HTML

CSS

JavaScript



## Fun Fact

- ในอดีต JavaScript มีอีกชื่อเรียกว่า LiveScript
- แต่ในขณะนั้น ภาษา Java ได้รับความนิยมอย่างมาก จึงเปลี่ยนชื่อเป็น JavaScript
- ในปัจจุบันมีคนเข้าใจว่า JavaScript เป็นภาษาเดียวกันกับ Java
- แต่ในความเป็นจริง JavaScript ไม่ได้มีความสัมพันธ์อันใดกับภาษา Java เลย

# Technical Documentation (Dev manual)

- [JavaScript MDN](#)
- [W3Schools](#)
- [JavaScript Info](#)
- <https://eloquentjavascript.net/>

# Run Javascript in browser

- วิธีเรียกใช้งาน JavaScript ทำได้โดยแทรก `<script></script>` เข้าไปในไฟล์ html ทำได้ 2 วิธี คือ
  - Internal JavaScript
  - External JavaScript

# Internal Script

```
<body>  
  <script>  
    console.log('Hello Codecamp Student');  
  </script>  
</body>
```

# External Script

```
<body>  
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.js"></script>  
  <script src="main.js"></script>  
</body>
```

# Bugs

**\*\* ใน script tag ถ้า src ถูกกำหนดแล้ว คำสั่งที่เขียนใน tag นั้นจะไม่ทำงาน \*\***

```
<body>
  <script src="/js/main.js">
    console.log('Internal script') // the content is ignored, because src is set
  </script>
</body>
```



**LAB**



# LAB-1 : Fundamental

- จงเขียนโค้ดเพื่อแสดง alert box บน web browser โดยมีข้อความว่า “My name is ” ตามด้วยชื่อตัวเอง
- ให้เขียนโค้ดทั้งวิธี Internal JavaScript และ External JavaScript





## LAB-2 : Fundamental

- จงเขียนโค้ดเพื่อ import external javascript library เข้ามาใช้ใน file html
- ให้ import library font-awesome และ jQuery



Software Park Thailand  
</Code Camp>

# JS Core Features (Basic)

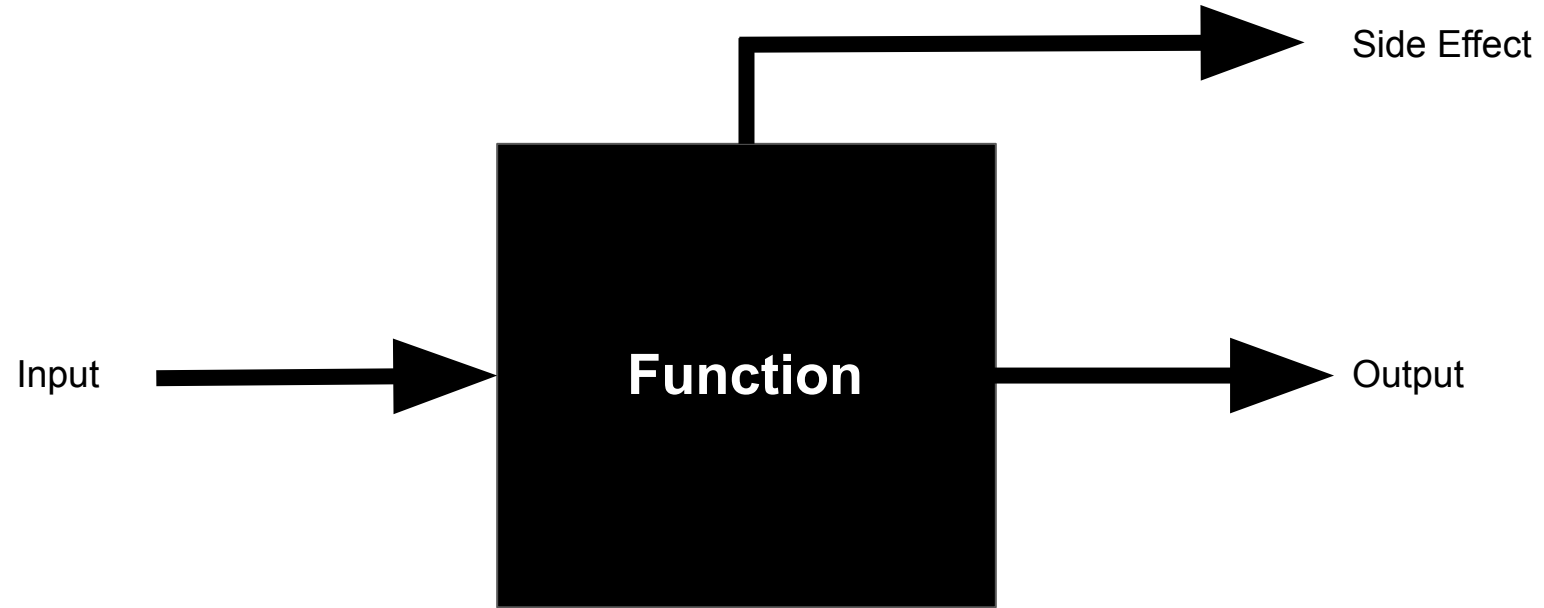
*JavaScript is a fun language  
with an incredible amount of possibilities.*



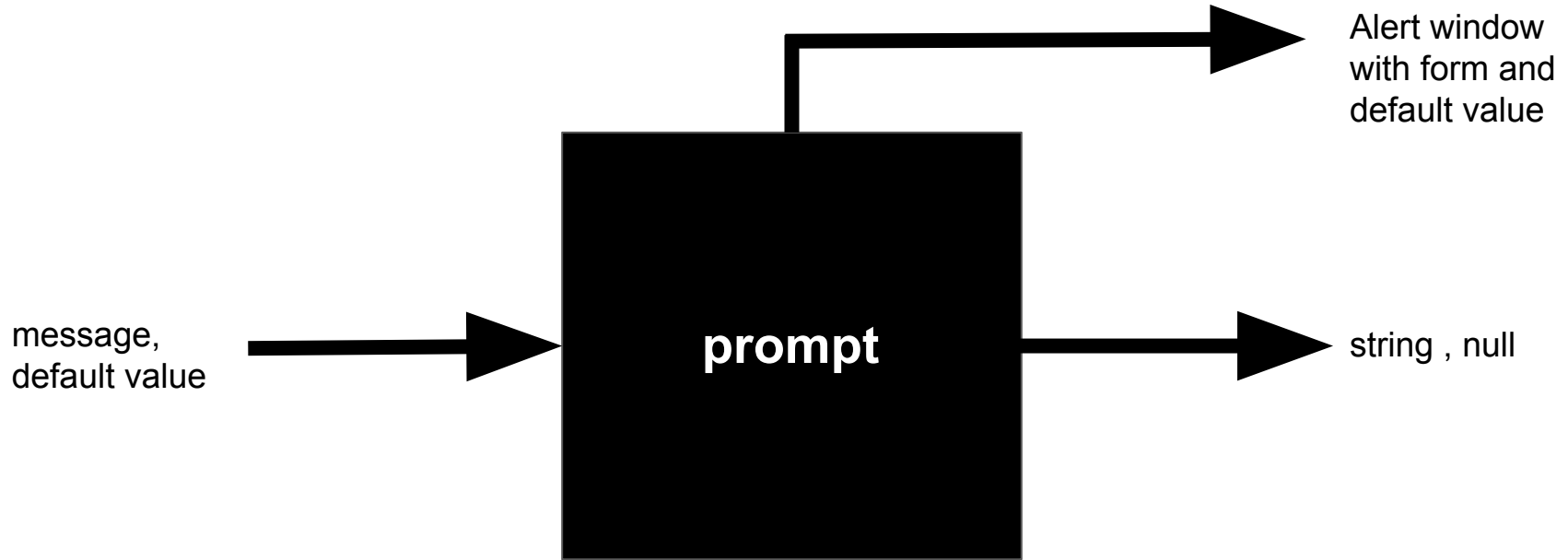
## built-in function

- ทดสอบการใช้งานฟังก์ชัน [prompt](#)
- ทดสอบการใช้งานฟังก์ชัน [console.log](#)

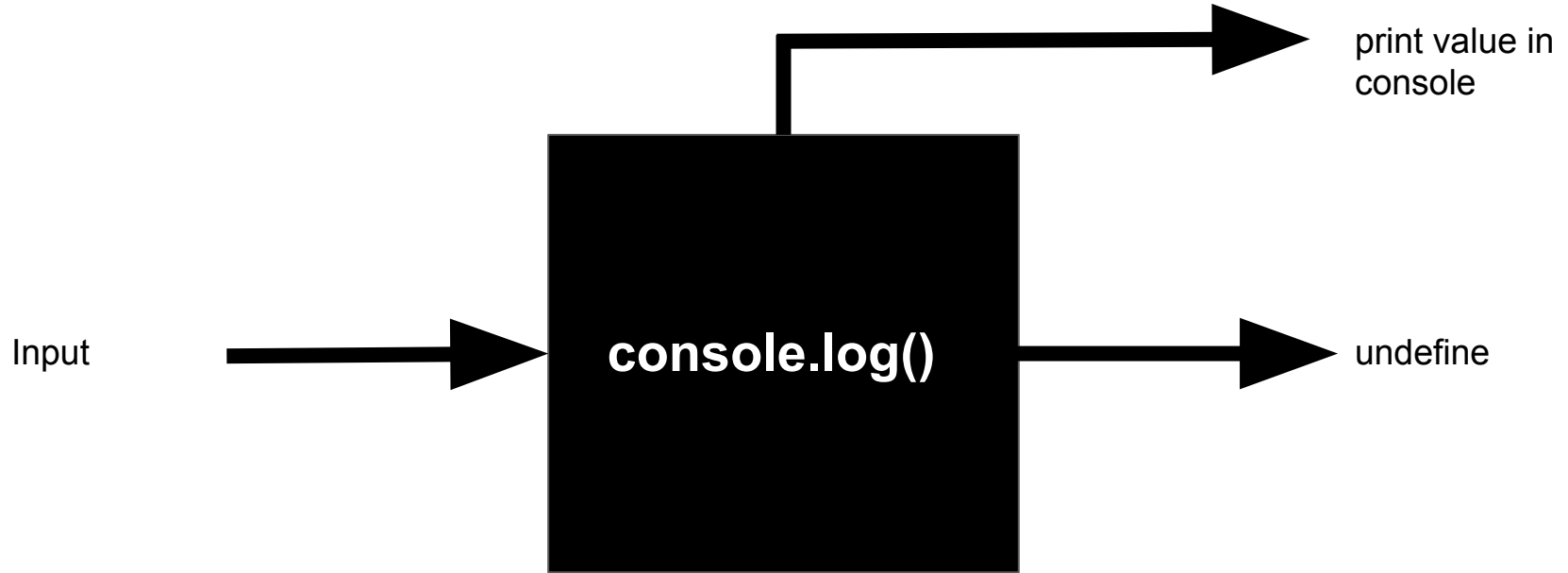
# Summary



# Summary



# Summary





Software Park Thailand  
</Code Camp>

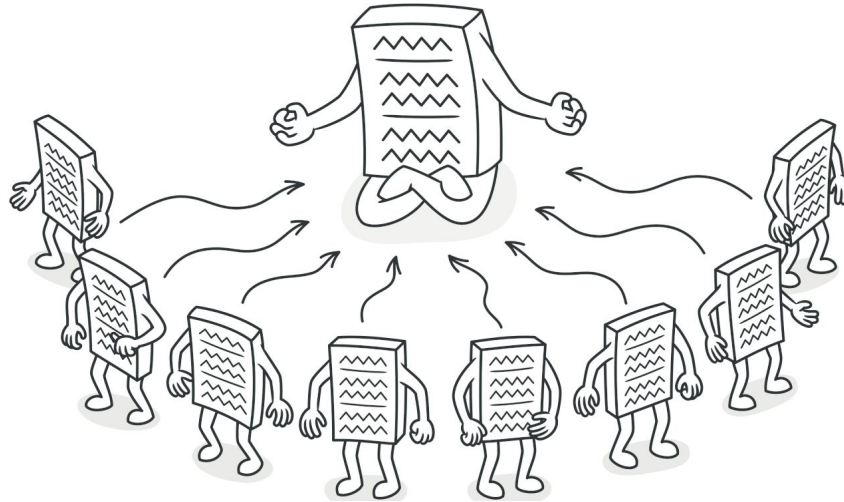
# JS Core Features (Intermediate)

*JavaScript is a fun language  
with an incredible amount of possibilities.*



# Prototype Based

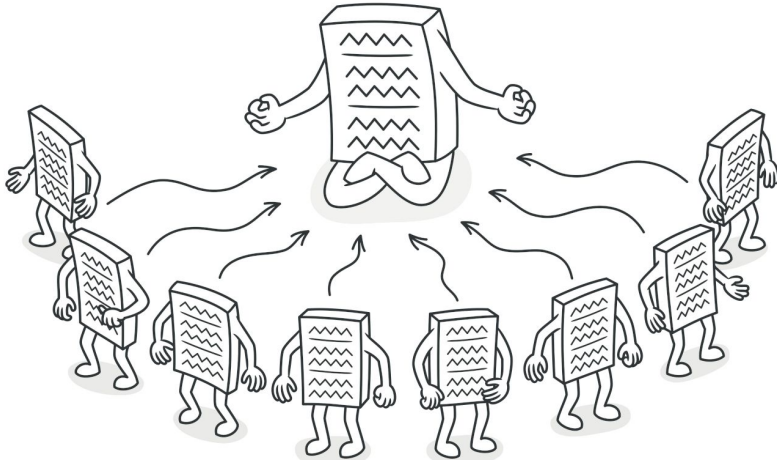
JavaScript is a **prototype-based** language, meaning object **properties and methods** can be **shared through** generalized **objects** that have the ability to be cloned and extended.



ref : <https://refactoring.guru/design-patterns/singleton>



# Prototype Based

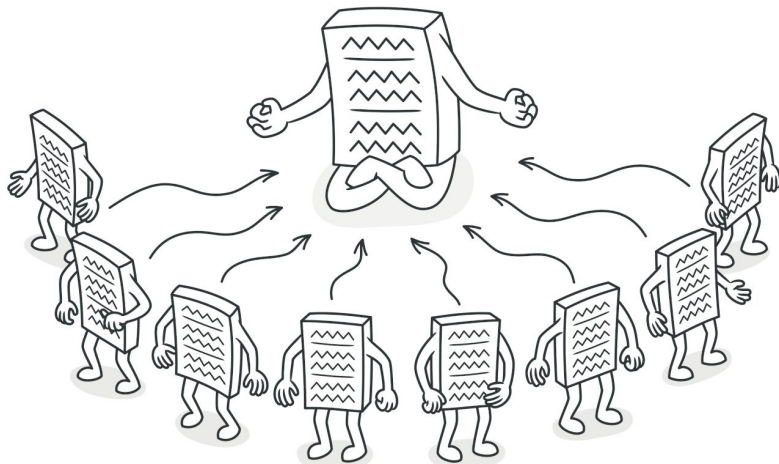


ref : <https://refactoring.guru/design-patterns/singleton>

```
// let num = 5  
let num = Number(5);
```

```
isNaN(num); //false  
parseInt(num); // 5  
num.toString(); // "5.5"  
num.toFixed(4); // 5.5000
```

# Prototype Based



ref : <https://refactoring.guru/design-patterns/singleton>

```
// let str = 'Hello'  
let str = String('Hello');
```

```
str.at(2);  
str.charAt(5);  
str.charCodeAt(2);  
str.indexOf('l');  
str.indexOf('0');  
str.includes('el');  
str.toLowerCase();  
str.toUpperCase();  
str.repeat(5);
```



Software Park Thailand  
</Code Camp>

# JS Buidling blog

*Let's rock with lego*

# Program Structure



Software Park Thailand  
</Code Camp>

**Environment**

**Functions**

*Control Flow*

**Loop**

**Variable**

**Statements**

**Branch**

**DataType**

**Expression**

**Comment**

# Program Structure

**Environment**

**Functions**

*Control Flow*

**Loop**

**Variable**

**Statements**

**Branch**

**DataType**

**Expression**

**Comment**



Software Park Thailand  
</Code Camp>

# Statement

# Statement



Software Park Thailand  
</Code Camp>

**Environment**

**Functions**

*Control Flow*

**Loop**

**Variable**

**Statements**

**Branch**

**DataType**

**Expression**

**Comment**

# Statement

## Statement

- Statement ไม่จำเป็นต้องจบคำสั่งด้วย semicolon
- ในกรณีที่ไมใส่ semicolon JavaScript จะทำการใส่ semicolon ให้เองเรียกว่า automatic semicolon insertion โดยปกติ จะใส่ semicolon เมื่อขึ้นบรรทัดใหม่ แต่มีข้อยกเว้นบางกรณี

```
alert('Hello')
```

```
alert(1+  
      2+  
      3)
```



# Statement

## Statement

- \*\* หากยังไม่แน่ใจเรื่อง automatic semicolon insertion ให้ทำการ ใส่ semicolon เมื่อจบ Statement เสมอ



# Comment

# Comment



Software Park Thailand  
«/Code Camp»

**Environment**

**Functions**

*Control Flow*

**Loop**

**Variable**

**Statements**

**Branch**

**DataType**

**Expression**

**Comment**

# Comment

## Comment

- Comment คือสิ่งที่เขียนไว้ในโปรแกรมเพื่ออธิบายการทำงานของโค้ด
- เขียนไว้ส่วนไหนก็ได้ของโปรแกรม เพราะ Comment นั้นจะไม่มีผลต่อการทำงานของโปรแกรม

```
// initial first name is John Doe  
var name = 'John Doe';
```

# Comment

## Comment

- การเขียน Comment มีทั้งหมด 2 แบบ
  - Comment แบบบรรทัดเดียว
  - Comment แบบหลายบรรทัด

```
// one line comment
```

```
/*  
multiple  
line  
comment  
*/
```



Software Park Thailand  
</Code Camp>

