

Computer Programming & Problem Solving Skills

DAY 4 - ROAD TO FRONT-END DEVELOPER BOOTCAMP

```
scientist.rb * default.rb x observation.rb
!   the mutable result of running an experiment.
! class Scientist::Result
!   attr_reader :candidates
!   attr_reader :control
!   attr_reader :experiment
!   attr_reader :ignored
!   attr_reader :mismatched
!   # An Array of Observations in execution order.
!   attr_reader :observations
!
!   # Internal: Create a new result.
!
!   # experiment - the Experiment this result is for
!   # observations: - an Array of Observations, in execution order
!   # control: - the control Observation
!
!   def initialize(experiment, observations = [], control = nil)
!     @experiment = experiment
!     @observations = observations
!     @control = control
!     @candidates = observations - [control]
!     evaluate_candidates
!
!     freeze
!
!   end
!
!   # Sub-class: the Experiment's context
!   def context
!     experiment.context
!   end
!
!   # Public: the name of the experiment
!   def experiment_name
!     experiment.name
!   end
!
!   # Public: as the result of an experiment?
!   def matched?
!     true
!   end
!
!   # Lib: to JSON
!   def to_json
!     observations.map { |o| o.to_json }
!   end
!
!   # Lib: to string
!   def to_s
!     observations.map { |o| o.to_s }.join("\n")
!   end
!
!   # Lib: freeze
!   def freeze
!     observations.freeze
!   end
!
!   # Lib: inspect
!   def inspect
!     "#<#{self.class}:>#{@control}<#{self.class}:>#{@candidates}"
!   end
!
!   # Lib: hash
!   def hash
!     observations.hash
!   end
!
!   # Lib: eql?
!   def eql?(other)
!     other.is_a?(Scientist::Result) && other.observations == observations
!   end
!
!   # Lib: ==
!   def ==(other)
!     eql?(other)
!   end
!
!   # Lib: to_s
!   def to_s
!     observations.map { |o| o.to_s }.join("\n")
!   end
!
!   # Lib: to_json
!   def to_json
!     observations.map { |o| o.to_json }
!   end
!
!   # Lib: inspect
!   def inspect
!     "#<#{self.class}:>#{@control}<#{self.class}:>#{@candidates}"
!   end
!
!   # Lib: hash
!   def hash
!     observations.hash
!   end
!
!   # Lib: eql?
!   def eql?(other)
!     other.is_a?(Scientist::Result) && other.observations == observations
!   end
!
!   # Lib: ==
!   def ==(other)
!     eql?(other)
!   end
!
```

การวนซ้ำ (Loop)





การวนซ้ำโดยอิงกับจำนวนรอบ
เราจะทำงานซ้ำไปเรื่อย ๆ 10 รอบ

การวนซ้ำโดยอิงกับจำนวนรอบ

เราจะทำงานซ้ำไปเรื่อย ๆ 5 รอบ

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
}
```

แบบทดสอบที่ 22

ลองทำโปรแกรมวนซ้ำตัวเลข 0 - 100

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
}
```

แบบทดสอบที่ 23

ลองทำโปรแกรมวนซ้ำตัวเลข 0 - 1000

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
}
```

การวนซ้ำโดยอิงกับเงื่อนไข^๑
ขณะที่ แล้วจะทำอะไรต่อไปเรื่อย ๆ ?

การวนซ้ำโดยอิงกับเงื่อนไข^ๆ บนจะที่ ... แล้วจะทำอะไรต่อไปเรื่อยๆ ?

```
let j = 1;  
while (j <= 5) {  
    console.log(j);  
    j++;  
}
```

```
let j = 1;  
while (j <= 5) {  
    console.log(j);  
    j++;  
}
```

**ຈະວນຫຼາໄປເຮື່ອຍ ຫຼື ທີ່
ເງື່ອນໄຂໃນກາຣວນຫຼ້າຍັງ
ຄອງເປັນ True**

```
let j = 5;
while (j > 0) {
    console.log(j);
    j--;
}
```

```
let num = 2;  
while (num <= 100) {  
    console.log(num);  
    num *= 2;  
}
```

ฟังก์ชันคืออะไร ? (Function)



$f(x) = ?$

console.log()
prompt()
alert()

พงก์ชันคือ ส่วนการทำงานที่จะทำ
เมื่อถูกเรียกใช้เท่านั้นโดย ..
“เรารสามารถส่งข้อมูลให้กับพงก์ชันได้
และ บางครั้ง พงก์ชันอาจส่งข้อมูลกลับ
มาหาเราได้เช่นกัน”

```
// ประกาศฟังก์ชัน
function add(a, b) {
    return a + b;
}

// เรียกใช้ฟังก์ชัน
console.log(add(5, 3));
```

แบบทดสอบที่ 24
ลองสร้างฟังก์ชัน "Hello ชื่อเรา"
รูปแบบของเราเอง

Collection Data

อยากรู้ว่ามีอะไรบ้าง ?

วิธีการเข้าถึงข้อมูลรูปแบบ array

ชื่อตัวแปร[ตำแหน่งข้อมูล]

วิธีการเข้าถึงข้อมูลรูปแบบ array

ชื่อตัวแปร[ตำแหน่งข้อมูล]

แต่ตำแหน่งเริ่มต้นจะเริ่มต้นที่ 0

วิธีการเข้าถึงข้อมูลรูปแบบ array

```
let fruits = ["apple", "banana", "cherry", "date"];
console.log(fruits);
```

วิธีการเข้าถึงข้อมูลรูปแบบ Array

```
let colors = ["red", "green", "blue", "yellow"];
console.log(colors[1]); // Outputs: "green"
```

แบบทดสอบที่ 25

ผลลัพธ์ของการทำงานจะได้ออกมาอย่างไร ?

```
let colors = ["red", "green", "blue", "yellow"];
console.log(colors.length);
```

เรารสามารถใช้ .length เพื่อคูณขนาดข้อมูลได้

```
let colors = ["red", "green", "blue", "yellow"];
console.log(colors.length);
```

เราสามารถใช้ ชื่อตัวแปร.push(ข้อมูล)
ในการเพิ่มข้อมูลลงไปใน Array ได้

```
let animals = ["cat", "dog", "bird"];
animals.push("elephant");
console.log(animals);
```

การแก้ไขข้อมูลใน Array

```
let numbers = [1, 2, 3, 4];
numbers[2] = 10;
console.log(numbers); // Outputs: [1, 2, 10, 4]
```

ลบสมาชิกหลังสุดใน Array

```
let cities = ["New York", "Los Angeles", "Chicago"];
let removedCity = cities.pop();
console.log(cities);      // Outputs: ["New York", "Los Angeles"]
console.log(removedCity); // Outputs: "Chicago"
```

การค้นหาข้อมูลใน Array

```
let books = ["Moby Dick", "1984", "The Great Gatsby"];
let index = books.indexOf("1984");
console.log(index); // Outputs: 1
```

การนำ Array มาใช้กับ Loop

```
let names = ["Alice", "Bob", "Charlie"];
for (let i = 0; i < names.length; i++) {
    |
    console.log(names[i]);
}
```

การบ้าน : สำหรับท่านที่ ไม่เคยเรียน หรือ ลืมเกี่ยวกับการเขียนโปรแกรมไปแล้ว

ชื่อบทเรียน	ความคืบหน้า
1. แนะนำวิทยาการคอมพิวเตอร์เบื้องต้น	กิจกรรมทดสอบ
2. เชิงจัด	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
3. คิดคำนวน	กิจกรรมทดสอบ
4. การเขียนภาษาขั้นตอน	กิจกรรมทดสอบ
5. คิลปิน	1 2 3 4 5 6 7 8 9 10
6. อัลกอริทึม	กิจกรรมทดสอบ
7. คิลปิน 2	1 2 3 4 5 6 7 8 9 10 11
8. พังผืด	กิจกรรมทดสอบ
9. ชوانา	1 2 3 4 5 6 7 8 9 10 11
10. เสื่อนไช	กิจกรรมทดสอบ
11. คิลปิน 3	1 2 3 4 5 6 7 8 9 10 11
12. การแต่งเพลง	กิจกรรมทดสอบ
13. ชوانา 2	1 2 3 4 5 6 7 8 9 10
14. Abstraction	กิจกรรมทดสอบ
15. คิลปิน 4	1 2 3 4 5 6 7 8 9 10
16. โปรแกรมการถ่ายทอดข้อมูล	กิจกรรมทดสอบ
17. ชوانา 3	1 2 3 4 5 6 7 8 9
18. อินทร์เน็ต	กิจกรรมทดสอบ
19. คิลปิน 5	1 2 3 4 5 6
20. Wrap-up	กิจกรรมทดสอบ

Accelerated Intro to CS คอร์ส

This 20-hour คอร์ส covers the core computer science and programming concepts in คอร์ส 2-4. The คอร์ส is designed for use with ages 10-18. Check out คอร์ส 2-4 for a more complete experience!

ลองดูหน้า

ขอความช่วยเหลือ

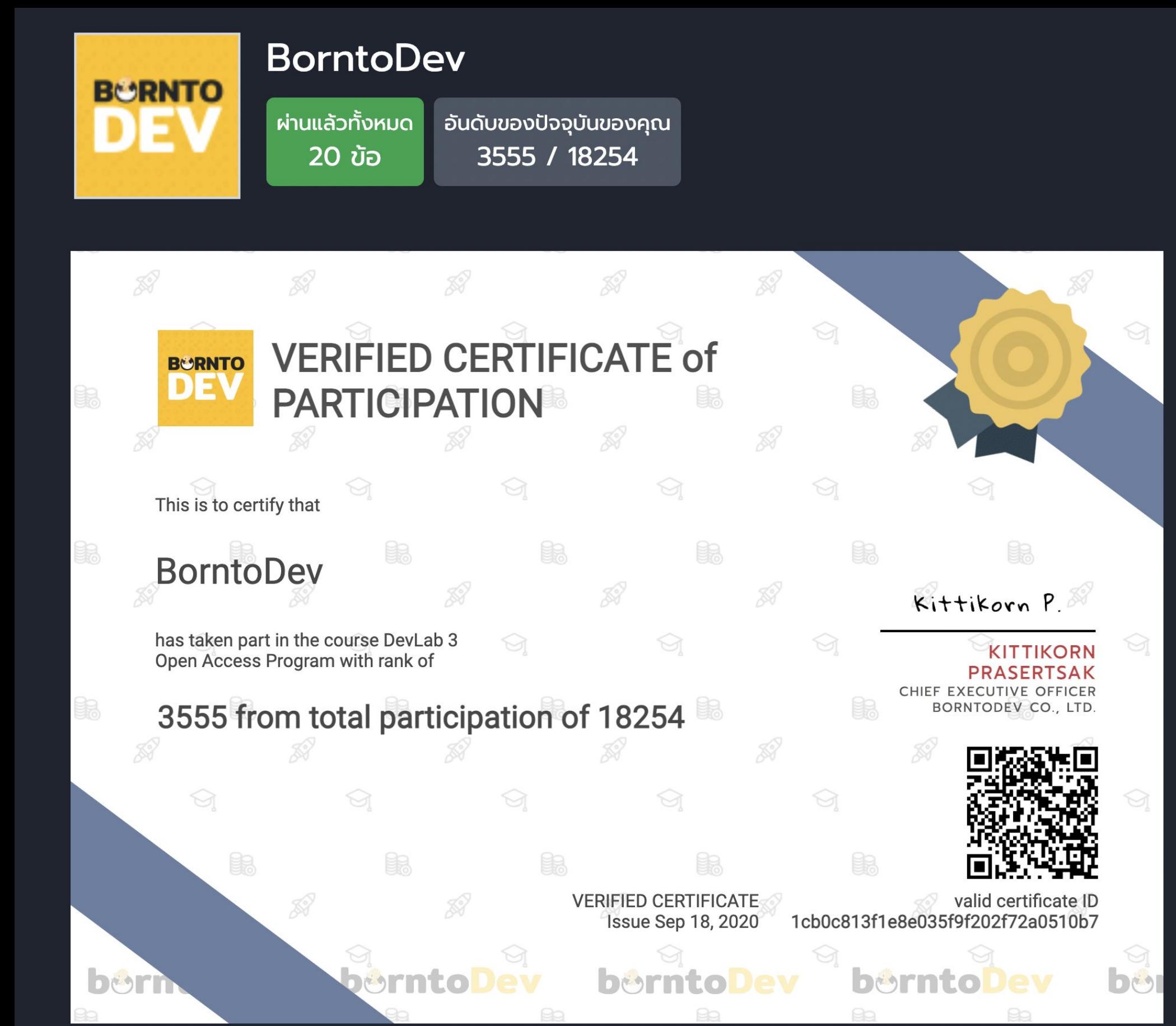


ใช้เวลาประมาณ 6 - 8 ชั่วโมงโดยประมาณ
มีแปลภาษาไทยให้สำหรับท่านที่ไม่ถนัดภาษา
อังกฤษ

โดยให้ทำบทเรียนที่ 2,5,7,9,11,13,15,17 และ 19
ให้ครบทั้งหมด ด้วยตัวเอง

ต้องทำให้เสร็จก่อนลุยการบ้านในวันนี้

หลังจากเรียนจบ จะได้ **Certificate** กลับบ้านกันคนละใบ
ยืนยันสมัครงาน เรียนต่อ ก็ใช้ยื่นประกอบได้ !



กลับมาที่ฟังก์ชันกันต่อ

ฟังก์ชัน สามารถส่งค่ากลับด้วย return ได้ด้วยนะ!

```
// ประกาศฟังก์ชัน
function add(a, b) {
    return a + b;
}

// เรียกใช้ฟังก์ชัน
console.log(add(5, 3));
```

ฟังก์ชัน สามารถส่งค่ากลับด้วย return ได้ด้วยนะ !

```
1  function isEven(num) {  
2      |     return num % 2 === 0;  
3  }  
4  
5  console.log(isEven(4)); // Output: true  
6  console.log(isEven(7)); // Output: false
```

ฟังก์ชัน สามารถส่งค่ากลับด้วย return ได้ด้วยนะ !

```
1 function celsiusToFahrenheit(celsius) {  
2     return (celsius * 9/5) + 32;  
3 }  
4  
5 console.log(celsiusToFahrenheit(0)); // Output: 32
```

ฟังก์ชัน สามารถส่งค่ากลับด้วย return ได้ด้วยนะ !

```
1  function maxOfTwo(a, b) {  
2      |    return a > b ? a : b;  
3  }  
4  
5  console.log(maxOfTwo(3, 9)); // Output: 9  
6  console.log(maxOfTwo(12, 5)); // Output: 12
```

ฟังก์ชัน สามารถส่งค่ากลับด้วย return ได้ด้วยนะ !

```
1  function rectangleArea(length, width) {  
2      |     return length * width;  
3  }  
4  
5  console.log(rectangleArea(5, 10)); // Output: 50
```

Collection Data

อยากรู้ว่า Collection Data คืออะไร ?

วิธีการเข้าถึงข้อมูลรูปแบบ List

ชื่อตัวแปร[ตำแหน่งข้อมูล]

วิธีการเข้าถึงข้อมูลรูปแบบ List

ชื่อตัวแปร[ตำแหน่งข้อมูล]

แต่ตำแหน่งเริ่มต้นจะเริ่มต้นที่ 0

ทำไม^ๆ
ตำแหน่งเริ่มต้นที่ 0 ?

รู้้มีอะไร ? ว่านอกจาก Array แล้ว
ยังมีตัวอื่น ๆ อีกนะ ! มารู้จัก Map กัน !

ข้อมูลจะถูกเก็บในลักษณะ Key , Value

The screenshot shows the MDN Web Docs page for the `Map` object. The page is in dark mode. The navigation bar at the top includes links for `mdn web docs`, `References`, `Guides`, `Plus`, `Blog`, `Play`, `AI Help`, `Theme`, a search bar, `Log in`, and `Sign up for free`. The breadcrumb navigation shows the path: `References > JavaScript > Reference > Standard built-in objects > Map`. The page title is `Map`. On the left, there's a sidebar for `Standard built-in objects` with a section for `Map` which is currently selected. The main content area starts with a brief description: "The `Map` object holds key-value pairs and remembers the original insertion order of the keys. Any value (both objects and primitive values) may be used as either a key or a value." Below this is a `Try it` section containing a JavaScript demo. The demo code is as follows:

```
JavaScript Demo: Map
1 const map1 = new Map();
2
3 map1.set('a', 1);
4 map1.set('b', 2);
5 map1.set('c', 3);
6
7 console.log(map1.get('a'));
8 // Expected output: 1
9
10 map1.set('a', 97);
11
12 console.log(map1.get('a'));
13 // Expected output: 97
14
15 console.log(map1.size);
16 // Expected output: 3
17
18 map1.delete('b');
19
20 console.log(map1.size);
21 // Expected output: 2
22
```

The right sidebar contains a list of sections: `In this article`, `Try it`, `Description`, `Constructor`, `Static properties`, `Static methods`, `Instance properties`, `Instance methods`, `Examples`, `Specifications`, `Browser compatibility`, and `See also`.

การเข้าถึงสามารถทำได้โดยการใส่ key ลงไปใน .get()

```
1 const map1 = new Map();
2
3 map1.set('a', 1);
4 map1.set('b', 2);
5 map1.set('c', 3);
6
7 console.log(map1.get('a'));
8 // Expected output: 1
9
10 map1.set('a', 97);
11
12 console.log(map1.get('a'));
13 // Expected output: 97
14
15 console.log(map1.size);
16 // Expected output: 3
17
18 map1.delete('b');
19
20 console.log(map1.size);
21 // Expected output: 2
22
```

เรารู้ว่าสามารถทำการวนซ้ำภายใน Map ได้

```
1 let map2 = new Map( [
2   ['apple', 1],
3   ['banana', 2]
4 ]);
5
6 for (let [key, value] of map2) {
7   console.log(`Key: ${key}, Value: ${value}`);
8 }
9 // Output:
10 // Key: apple, Value: 1
11 // Key: banana, Value: 2
```

หรือต้องดูจนหาข้อมูลใน Map ว่ามีข้อมูลนี้ ๆ ไหม ก็ทำได้ง่ายๆ

```
let map3 = new Map();
map3.set('name', 'Alice');

console.log(map3.has('name')); // Output: true
console.log(map3.has('age')); // Output: false
```

การลบข้อมูลใน Map

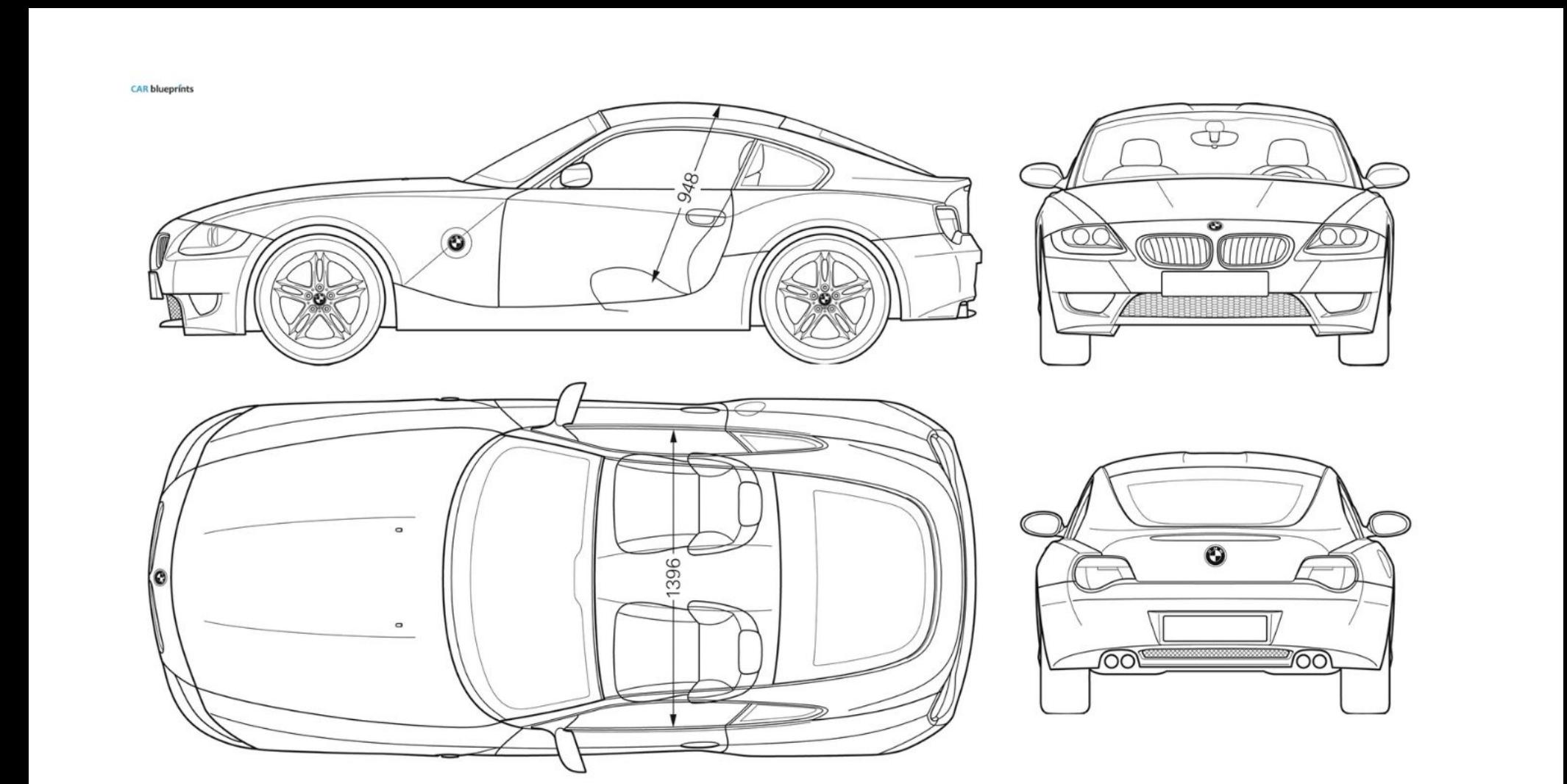
```
1 let map4 = new Map();
2 map4.set('city', 'New York');
3
4 console.log(map4.get('city')); // Output: New York
5
6 map4.delete('city');
7 console.log(map4.get('city')); // Output: undefined
```

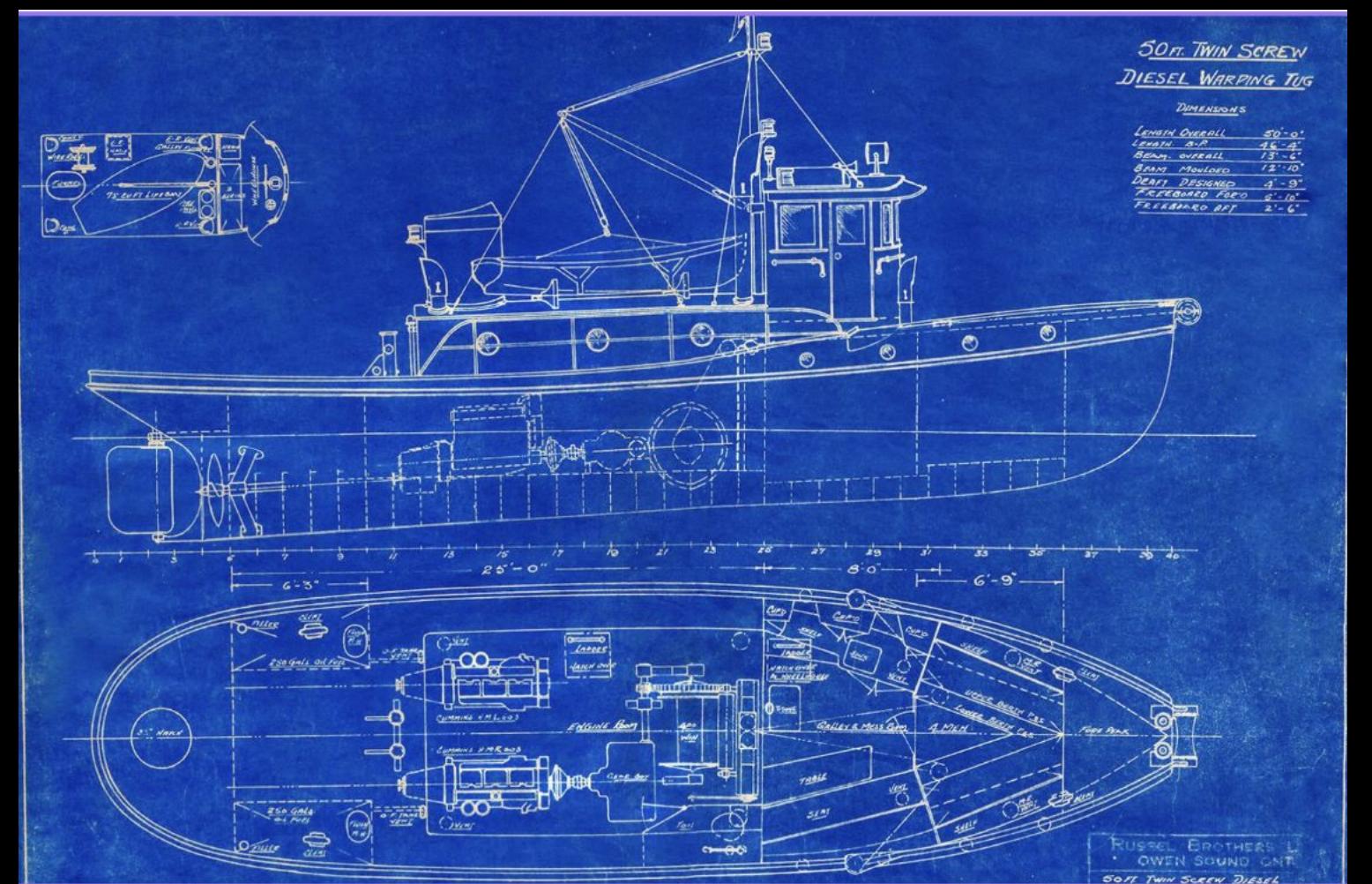
ก่อนจะไปกล่าวว่า
เราพอเห็นความแตกต่าง
ของทั้งสองตัวนี้หรือยังนะ ?

ตัวต่อมา ขอแนะนำให้รู้จักกับ Set

```
1 // Creating a Set
2 let fruits = new Set();
3
4 // Adding values
5 fruits.add('apple');
6 fruits.add('banana');
7 fruits.add('apple'); // Won't be added again since it's a duplicate
8
9 console.log(fruits.has('apple')); // Output: true
10 console.log(fruits.size); // Output: 2
11
12 // Deleting a value
13 fruits.delete('apple');
14 console.log(fruits.has('apple')); // Output: false
15
16 // Iterating through a Set
17 for (let fruit of fruits) {
18   console.log(fruit);
19 }
```

มารู้จักกับ การเขียนโปรแกรมเชิงวัตถุ
Object Oriented Programming กันดีกว่า !





เปรียบเสมือน
พิมพ์เขียวของสิงของ

การเขียนโปรแกรม
เชิงวัตถุจะเป็นการมองภาพ
ส่วนต่าง ๆ ของโปรแกรมเป็นชิ้นส่วนต่าง ๆ

ภายในวัตถุต่าง ๆ มักจะมี 2 สิ่งคือ
Attributes (คุณลักษณะ)
Behaviors (พฤติกรรม)



Attributes (គុណលักษ្រณៈ)
**ຈាន់ត្រូវបានការពារ
ដើម្បីរៀបចំការងារ
នៃគ្រប់គ្រងការងារ
និងការងារ**



- + ชื่อ
- + อายุ
- + น้ำหนัก
- + ขนาด
- + สี
- + ลาย



- + ชื่อ : ประเกกข้อมูล
- + อายุ: ประเกกข้อมูล
- + น้ำหนัก: ประเกกข้อมูล
- + ขนาด: ประเกกข้อมูล
- + สี: ประเกกข้อมูล
- + ลาย: ประเกกข้อมูล



Behaviors (ພັດຕິກຣມ)
**គ້ອງນານ ຫຣ້ວ ກາຣກະທໍາ
ກ່ຽວຕາຄຸນ້ນ ໃຊ້ ສາມາຮຄທໍາໄດ້**



- + กิน
- + เคลื่อนที่
- + นั่ง
- + นอน



- + กົບ()
- + ເຄລື່ອນທີ່()
- + ນັ້ງ()
- + ນອນ()



ຕົວແປຣ

(ຫຼັອ Properties, Getter Setter)

- + ທີ່ອ : ປະເກດຂ້ອມູລ
- + ອາຍຸ: ປະເກດຂ້ອມູລ
- + ນໍ້າຫັນ: ປະເກດຂ້ອມູລ
- + ຂາດ: ປະເກດຂ້ອມູລ
- + ສີ: ປະເກດຂ້ອມູລ
- + ລາຍ: ປະເກດຂ້ອມູລ

ຝຶງກໍບັນ

- + ກິບ()
- + ເຄລື່ອນທີ()
- + ນັ້ງ()
- + ນອນ()



แบบทดสอบที่ 41

Attributes และ
Behaviors มีอะไรบ้าง ?



แบบทดสอบที่ 42

Attributes และ
Behaviors มีอะไรบ้าง ?

```
1 class Tiger {  
2     constructor(name, age, weight) {  
3         this.name = name; //Property  
4         this.age = age; //Property  
5         this.weight = weight; //Property  
6     }  
7  
8     roar() { //Method  
9         console.log(` ${this.name} roars! `);  
10    }  
11  
12    eat(pounds) { //Method  
13        this.weight += pounds;  
14        console.log(` ${this.name} eats ${pounds} pounds of food. `);  
15    }  
16}  
17  
18 const rajah = new Tiger('Rajah', 5, 450); //Instance  
19 rajah.roar(); //Rajah roars!  
20 rajah.eat(5); //Rajah eats 5 pounds of food.
```



```
18 const rajah = new Tiger('Rajah', 5, 450); //Instance
19 rajah.roar(); //Rajah roars!
20 rajah.eat(5); //Rajah eats 5 pounds of food.
21
22 const redTiger = new Tiger('Red Tiger', 3, 300); //Instance
23 redTiger.roar(); //Red Tiger roars!
24 redTiger.eat(10); //Red Tiger eats 10 pounds of food.
25
26 const blueTiger = new Tiger('Blue Tiger', 4, 350); //Instance
27 blueTiger.roar(); //Blue Tiger roars!
28 blueTiger.eat(15); //Blue Tiger eats 15 pounds of food.
29
```

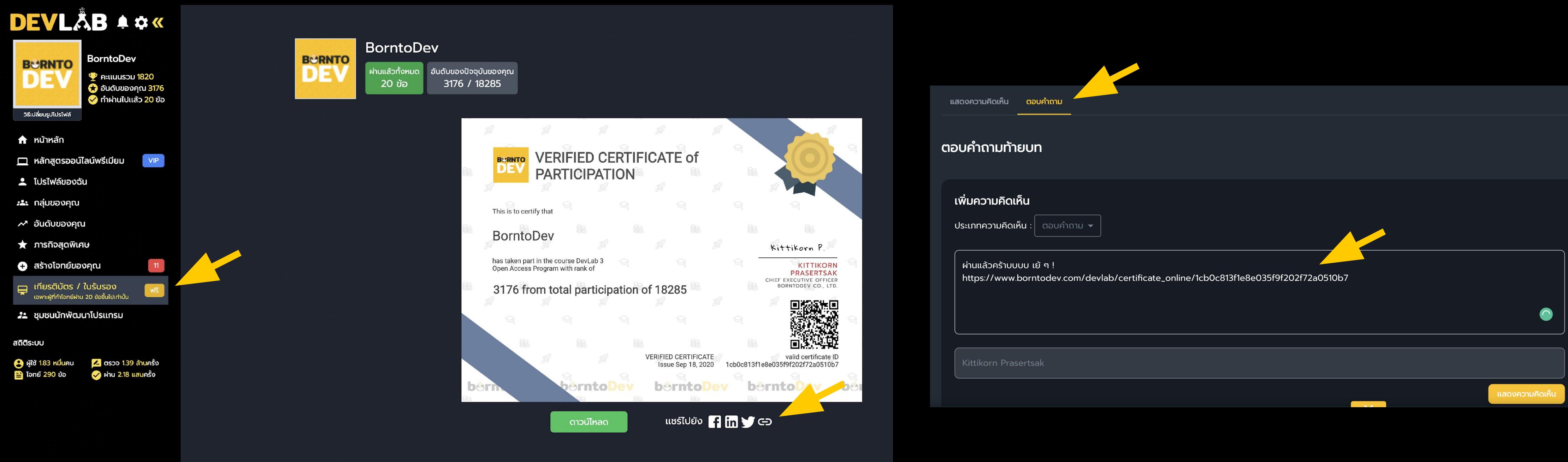


```
1 class Car {  
2     constructor(brand, model) {  
3         this.brand = brand; // Property  
4         this.model = model; // Property  
5     }  
6  
7     display() { // Method  
8         console.log(` ${this.brand} ${this.model}`);  
9     }  
10 }  
11  
12 const myCar = new Car('Toyota', 'Corolla');  
13 myCar.display();
```

ของแถม การหาความรู้ และ ใช้งานฟังก์ชันอื่น ๆ เพิ่มเติม

The screenshot shows a web browser displaying the W3Schools website. The navigation bar at the top includes links for Tutorials, Exercises, Get Certified, Services, Bootcamps, Spaces, and a user profile icon. The main menu below the bar has tabs for HTML, CSS, JAVASCRIPT (which is highlighted in green), SQL, PYTHON, JAVA, PHP, HOW TO, W3.CSS, C, C++, C#, BOOTSTRAP, and REACT. On the left side, there is a sidebar with a list of JavaScript topics: JS Introduction, JS Where To, JS Output, JS Statements, JS Syntax, JS Comments, JS Variables, JS Let, JS Const, JS Operators, JS Arithmetic, JS Assignment, JS Data Types, JS Functions, JS Objects, JS Events, JS Strings, JS String Methods (which is also highlighted in green), JS String Search, JS String Templates, JS Numbers, JS BigInt, and JS Number Methods. The main content area on the right is titled "JavaScript String slice()" and contains a brief explanation: "slice() extracts a part of a string and returns the extracted part in a new string. The method takes 2 parameters: start position, and end position (end not included)." Below this, there is an "Example" section with sample code: "let text = "Apple, Banana, Kiwi"; let part = text.slice(7, 13);". A "Try it Yourself" button is located below the example. At the bottom of the page, there is a yellow "Note" section with the text "JavaScript counts positions from zero."

การบ้านสัปดาห์นี้ #1



ลุยต่อใน Dev Lab เหมือนเดิม ! โดยสามารถดูได้จากใน School เมื่อได้ Certificate แล้ว มาแชร์ลิงก์กันในคำตอบได้เลย

การบ้านสัปดาห์นี้ #2

The image shows two screenshots side-by-side. On the left is the w3schools.com JavaScript tutorial homepage, featuring a sidebar with various JS topics and a main content area with introductory text about JavaScript. On the right is a dark-themed feedback form titled 'แสดงความคิดเห็นของคุณ' (Please leave your comment). It includes fields for 'เพิ่มความคิดเห็น' (Add comment), a dropdown menu for 'ประเภทความคิดเห็น' (Type of comment) set to 'พูดถูก' (True), and a text area for 'ความคิดเห็น' (Comment). A yellow arrow points to the top right corner of the feedback form, and another yellow arrow points to the 'เพิ่มความคิดเห็น' button.

คึกขาหา Tip / เทคนิคใน w3school ในการใช้งานฟังก์ชัน หรือ ใช้งานประเภท
ข้อมูลต่าง ๆ โดยอธิบายว่า มันคือฟังก์ชันอะไร, เจ่งยังไง, ทำงานยังไง
และ ได้ผลลัพธ์เป็นยังไง
(โดยห้ามซ้ำกับเพื่อน หรือ ซ้ำกับในบทเรียนนะครับ)