

Javascript Basic

BY PAIBOON PANUSBORDEE
FRONT-END PROGRAMMER BOOTCAMP



Website Request



Frontend

HTML

CSS

React



Backend

Node.js

REST API

DevOps



Database

MySQL

Javascript

Javascript คืออะไร?



Javascript คืออะไร?



JS

Javascript Version

- ▶ ECMAScript 5 (ES5)
- ▶ ECMAScript 6 (ES6) / ECMAScript 2015 (ES2015)
- ▶ ECMAScript 7 (ES7) / ECMAScript 2016 (ES2016)
- ▶ **ECMAScript 8 (ES8) / ECMAScript 2017 (ES2017) ← NOW!!**



คุณสมบัติของ Javascript

▶ Weakly typed

▶ `123 == "123"`

▶ `123 === 123`

▶ `123 !== "123"`

▶ `'123' * 2`

▶ Dynamic programming language

▶ `let a = "string";`

▶ `a = 123;`

▶ `a = {key1:1, key2:2};`

▶ `a.key1 = 5`

Javascript Variable



Variable Declarations

▶ ~~var a = 1;~~

▶ let a = 1;

▶ const a = 1;



Primitive Data Types

- ▶ Boolean
- ▶ Number
- ▶ String
- ▶ Null
- ▶ Undefined
- ▶ Symbol

```
let a = [1,2,3];  
  
typeof a; // “Object”
```

Object type

▶ Object

▶ Array

▶ มีคุณลักษณะเป็น Pointer

```
let a = [1,2,3];  
typeof a; // "Object"
```



Global Objects / Standard built-in objects

- ▶ Infinity
- ▶ NaN (Not a Number)
 - ▶ `parseInt("any string")`
 - ▶ `parseFloat("any string")`
- ▶ Date
- ▶ JSON
- ▶ Promise
- ▶ Error



Object == Array

- ▶ `const a = {key1 : "value1",key2 : "value2"}`
 - ▶ `a.key1`
 - ▶ `a['key1']`
- ▶ `const b = {"key1" : "value1", "key2" : "value2"}`
- ▶ `key1` คือ `key` หรือ `Properties` ของ `object a`
- ▶ ในภาษา `Static` จะเรียกว่า `Hash` หรือ `Dictionary`
- ▶ `const` ของตัวแปร `Object` ยังสามารถแก้ไขค่า `properties` ได้เช่น
`b.key1 = "new value";`
จะไม่ `Error`

Array and Object Declaration

▶ Array

▶ `const arr = [];`

▶ `arr[0] = "myValue";`

▶ `arr.push("myNextValue");`

▶ Object

▶ `const obj = {};`

▶ `obj.myProperties = "myValue";`

▶ `obj["myProperties2"] = "myValue2";`

▶ `obj.myFunction = function() {
 alert("Hello World");
}`

Function



Function Declaration

```
function hello(parameter1, parameter2, parameter3) {  
    alert('hello' + parameter1 + parameter2 + parameter3);  
};
```

```
let hello = function(parameter1, parameter2, parameter3) {  
    alert('hello' + parameter1 + parameter2 + parameter3);  
};
```

```
hello ('1', '2', '3');
```



Function as parameters

```
let hello = function(parameter1, parameter2, parameter3) {  
    alert('hello' + parameter1 + parameter2 + parameter3);  
};  
function func(callback) {  
    callback('1', '2', '3');  
}  
func(hello);
```



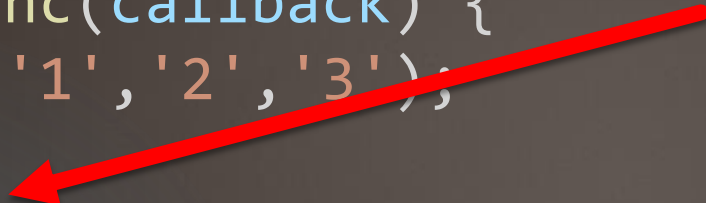
Callback Function

```
function func(callback) {  
    callback('1', '2', '3');  
}  
func(function(parameter1, parameter2, parameter3) {  
    alert('hello' + parameter1 + parameter2 + parameter3);  
}));
```



Anonymous Function

```
function func(callback) {  
    callback('1', '2', '3');  
}  
func(function(parameter1, parameter2, parameter3) {  
    alert('hello' + parameter1 + parameter2 + parameter3);  
}));
```



Function in Object

```
const myObject = {  
  myProperty: 123,  
  myFunction: function (message) {  
    alert(message);  
  }  
};  
alert(myObject.myProperty);  
myObject.myFunction("Hello World!");
```

Callback Function

```
const hiddenBox = $("#banner-message");  
$("#ToggleButton").on( "click", function(event) {  
    hiddenBox.show();  
});
```



Callback Function

```
$("#button").click(function(){  
    $.ajax({url: "demo_test.txt", success: function(result){  
        $("#div1").html(result);  
    }});  
});
```



Function as value or memory address?

```
function hello3() {  
    return 1;  
}
```

```
console.log(hello3); // Function  
console.log(hello3()); // 1
```



Function as value or memory address?

```
function hello3() {  
    return 1;  
}
```

```
function tryHello3(callbackFunction) {  
    return callbackFunction();  
}
```

```
console.log(hello3()); // 1  
console.log(hello3); // Function  
console.log( tryHello3(hello3) ); // 1  
console.log( tryHello3(hello3()) ); // ?
```

Function as value or memory address?

```
1
❌ ▶ Uncaught TypeError: callbackFunction is not a function
    at tryHello3 (<anonymous>:6:12)
    at <anonymous>:12:14
>
```


Useful Function (Cheat List)

- ▶ `Array.length`
- ▶ `parseInt()`
- ▶ `parseFloat()`
- ▶ `String.substr()`
- ▶ `String.search()`
- ▶ `String.replace()`
- ▶ `console.log()`
- ▶ `Array.slice()`
- ▶ `Array.push()`
- ▶ `Array.unshift()`
- ▶ `Array.sort()`
- ▶ `Array.reverse()`
- ▶ `Array.indexOf()`
- ▶ `alert()`

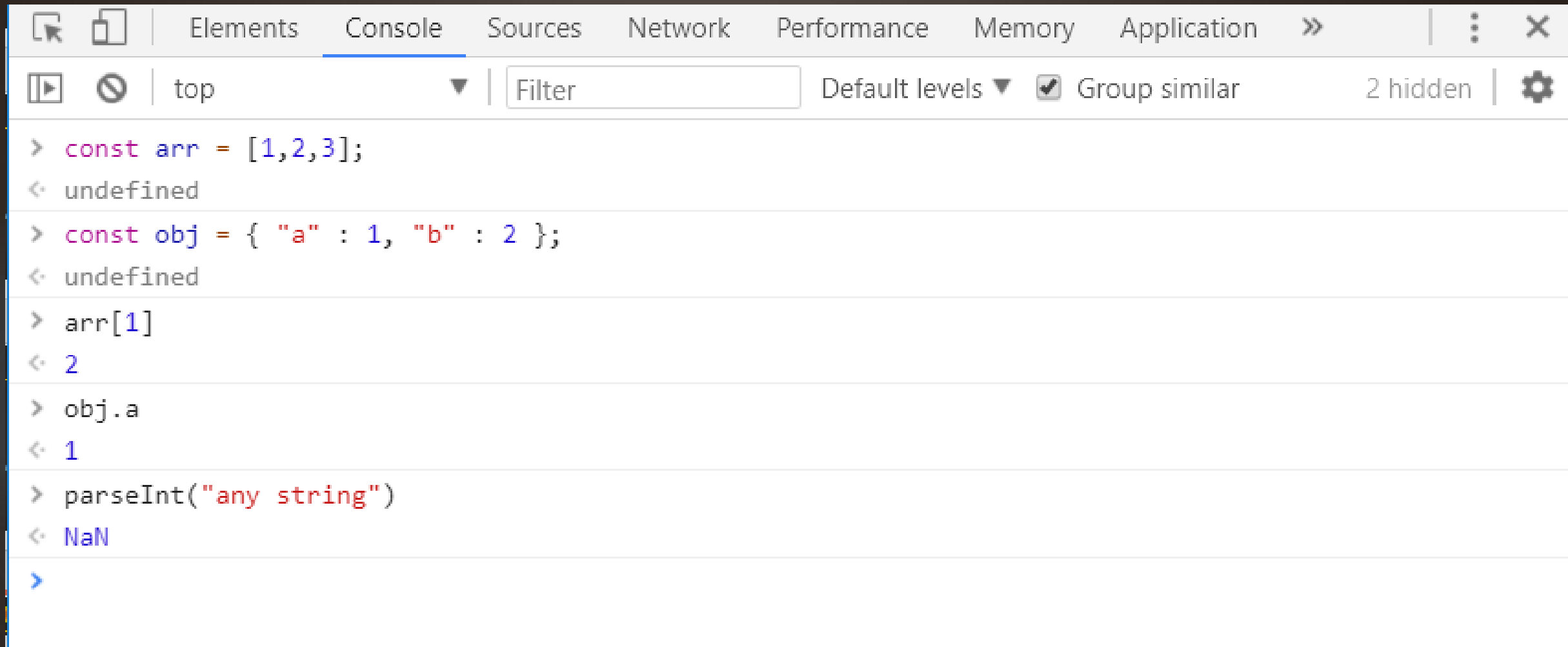
Useful Class

▶ [Date](#)

▶ [Math](#)



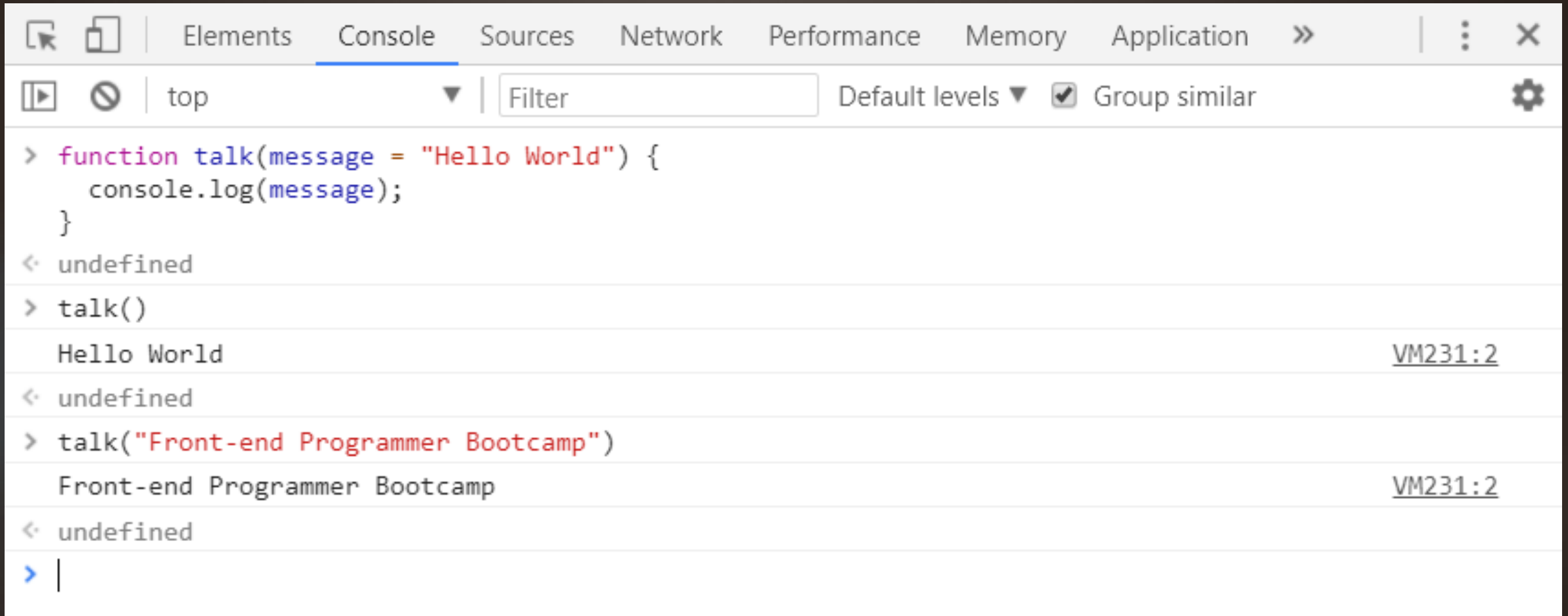
Google Chrome Developer Tools (F12)



The screenshot shows the Google Chrome Developer Tools interface with the Console tab selected. The top navigation bar includes tabs for Elements, Console, Sources, Network, Performance, Memory, and Application. The Console panel shows a series of commands and their outputs:

```
> const arr = [1,2,3];  
< undefined  
  
> const obj = { "a" : 1, "b" : 2 };  
< undefined  
  
> arr[1]  
< 2  
  
> obj.a  
< 1  
  
> parseInt("any string")  
< NaN  
  
>
```

Default parameters



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following sequence of commands and outputs:

```
> function talk(message = "Hello World") {  
  console.log(message);  
}  
< undefined  
> talk()  
Hello World  
< undefined  
> talk("Front-end Programmer Bootcamp")  
Front-end Programmer Bootcamp  
< undefined  
> |
```

The output 'Hello World' and 'Front-end Programmer Bootcamp' are both followed by a link to 'VM231:2'.

การแทรก Javascript ใน HTML

```
<html>
<head>
<title>My Title</title>
<script>
    alert("Alert Hello World");
</script>
</head>
<body>
<p>Hello World</p>
<p>Hello World2</p>
<p>Hello World3</p>
</body>
</html>
```



การแทรก Javascript ใน HTML

```
<html>
<head>
<title>My Title</title>
</head>
<body>
<p>Hello World</p>
<p>Hello World2</p>
<p>Hello World3</p>
<script>
    alert("Alert Hello World");
</script>
</body>
</html>
```



การแทรก Javascript ใน HTML

```
<html>
<head>
<title>My Title</title>
</head>
<body>
<p>Hello World</p>
<p>Hello World2</p>
<p>Hello World3</p>
</body>
</html>
<script>
    alert("Alert Hello World");
</script>
```



การเรียกไฟล์ Javascript จากข้างนอก

```
<html>
<head>
<title>My Title</title>
<script src="myFirstJavascript.js"></script>
</head>
<body>
<p>Hello World</p>
<p>Hello World2</p>
<p>Hello World3</p>
</body>
</html>
```

Index.html

```
alert("Alert Hello World");
```

myFirstJavascript.js



Lab 1: Google Chrome Developer Tools

- ▶ ทดลองประกาศ Function ใส่ตัวแปรและโยนค่าไปมา
- ▶ ทดลอง Function ต่างๆ ใน javascript
- ▶ ทดลอง Date, Math ใน javascript
- ▶ ทดลอง scope ของตัวแปรที่ประกาศด้วย let เปรียบเทียบกับ var
- ▶ ทดลอง Object Pointer

```
let b = {'animal':'cat'};  
let c = b;  
c.animal = 'dog';  
console.log(b);  
console.log(c);
```

```
for (var i=0; i<10; i++) {  
    // do something  
}  
console.log(i);  
for (let j=0; j<10; j++) {  
    // do something  
}  
console.log(j);
```



Lab 1: Try Brackets

- ▶ Download Brackets จาก <http://brackets.io/>
- ▶ Install Brackets
- ▶ Create index.html และสร้าง tag <p> จำนวนหนึ่ง
- ▶ คลิก File -> Live Preview แล้วทดลองแก้ไข HTML



JavaScript Object Notation (JSON)



JSON คืออะไร?

- ▶ รูปแบบมาตรฐานการส่งข้อมูลในทุกภาษา
- ▶ สามารถอ่านได้ง่าย
- ▶ เบา (Lightweight)



JSON example

```
const students = [  
  {"firstname": "Somchai", "lastname": "SudLor", "age": "50"},  
  {"firstname": "Somsri", "lastname": "SudSuay", "age": "40"},  
  {"firstname": "Luke", "lastname": "Skywalker", "age": "60"}  
]
```

JSON vs XML

```
const students = [  
  {"firstname":"Somchai","lastname":"SudLor","age":"50"},  
  {"firstname":"Somsri","lastname":"SudSuay","age":"40"},  
  {"firstname":"Luke","lastname":"Skywalker","age":"60"}  
]
```

JSON

```
<?xml version="1.0" ?>  
<students>  
  <student>  
    <firstname>Somchai</firstname>  
    <lastname>SudLor</lastname>  
    <age>50</age>  
  </student>  
  <student>  
    <firstname>Somsri</firstname>  
    <lastname>SudSuay</lastname>  
    <age>60</age>  
  </student>  
  <student>  
    ...  
  </student>  
</students>
```

XML

JSON Data

▶ "firstname" : "Somchai"

▶ Key:Value



JSON Object

- ▶ `{"firstname": "Somchai", "lastname": "SudLor", "age": "50"}`
- ▶ `{firstname: "Somchai", lastname: "SudLor", age: "50"}`

JSON Compare

firstname	lastname	age
Somchai	SudLor	50
Somsri	SudSuay	40
Luke	Skywalker	60

```
[  
  {"firstname":"Somchai","lastname":"SudLor","age":"50"},  
  {"firstname":"Somsri","lastname":"SudSuay","age":"40"},  
  {"firstname":"Luke","lastname":"Skywalker","age":"60"}  
]
```

JSON function

- ▶ `JSON.stringify()` คือการ encode json ให้อยู่ในรูปของ string
- ▶ `JSON.parse()` คือการ decode json จาก string ให้กลับเป็น json object

JSON.parse

▶ ไม่สามารถ parse ได้หาก key ไม่มี Double Quote

▶ `let myVar = JSON.parse('{"firstname":"Somchai"}');`



▶ `let myVar = JSON.parse('{\'firstname\':"Somchai"}');`

▶ `let myVar = JSON.parse('{firstname:"Somchai"}');`



Lab 2: JSON

- ▶ ทดลอง Copy ไปใส่ Google Chrome Developer Tools ว่า

```
const students = [  
  {"firstname": "Somchai", "lastname": "SudLor", "age": "50"},  
  {"firstname": "Somsri", "lastname": "SudSuay", "age": "40"},  
  {"firstname": "Luke", "lastname": "Skywalker", "age": "60"}  
]
```

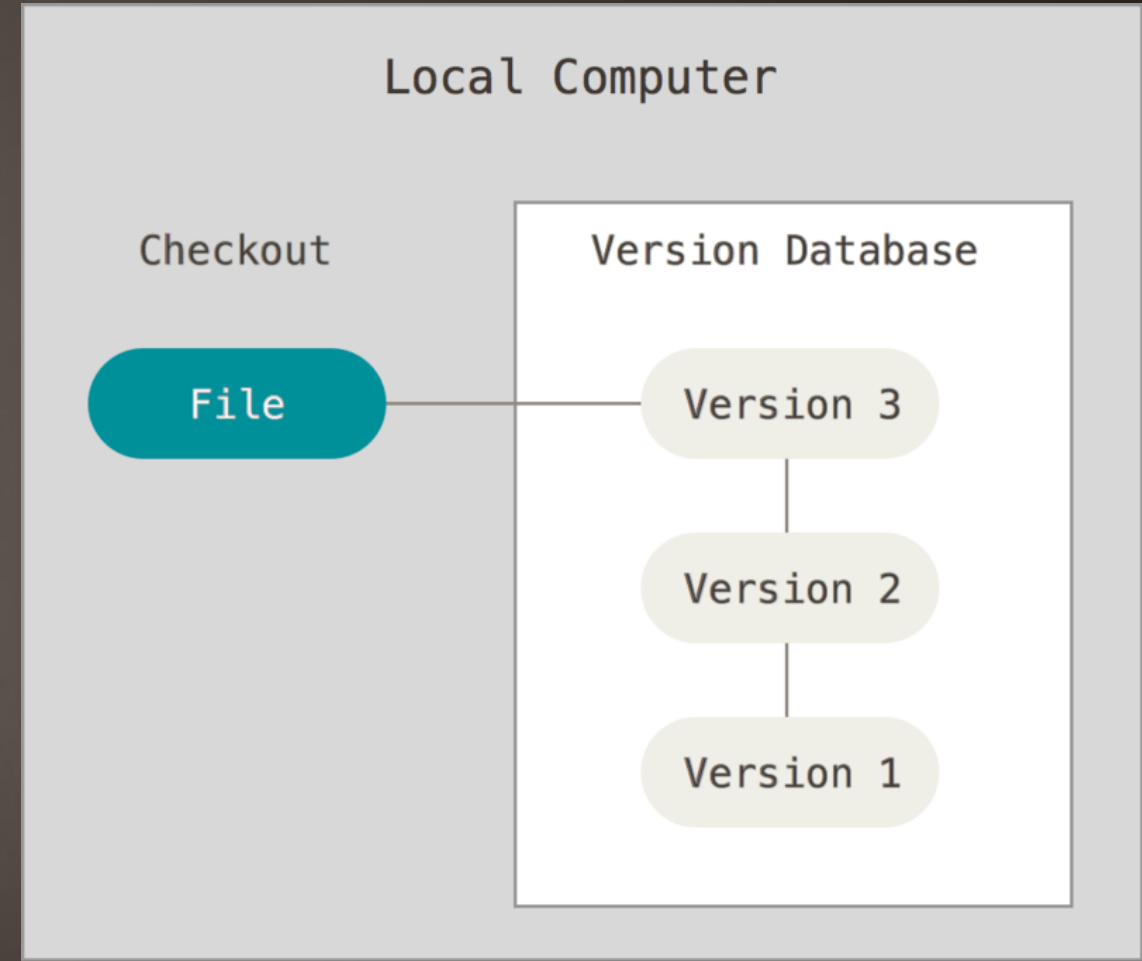
- ▶ ทดลองพิมพ์ students[0].firstname
- ▶ ทดลองพิมพ์ students[2].age = 120;
- ▶ let strStudent = JSON.stringify(student);
- ▶ JSON.parse(strStudent);

Version Control (GIT)



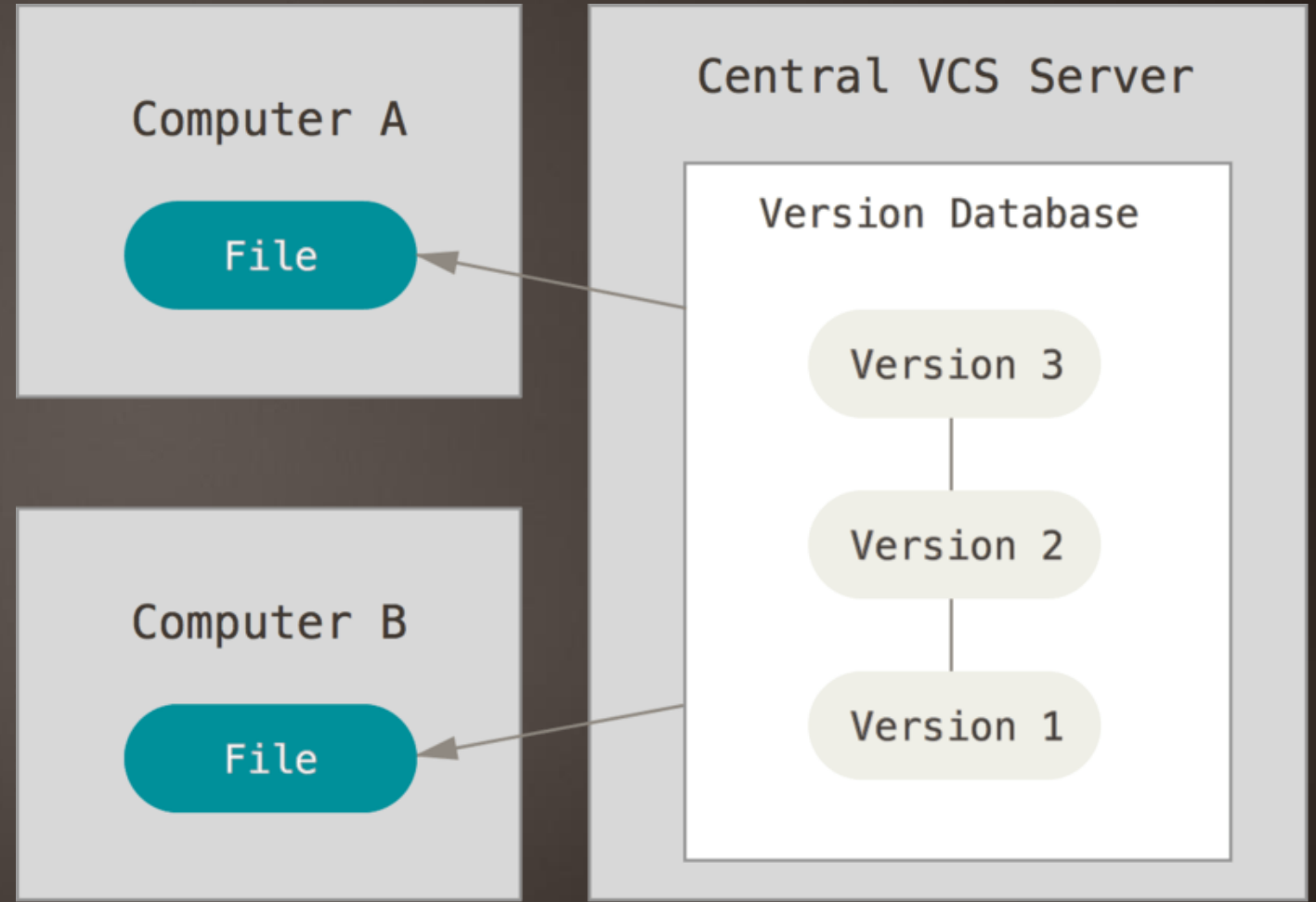
Local Version Control

- ▶ IDE History
- ▶ Local History

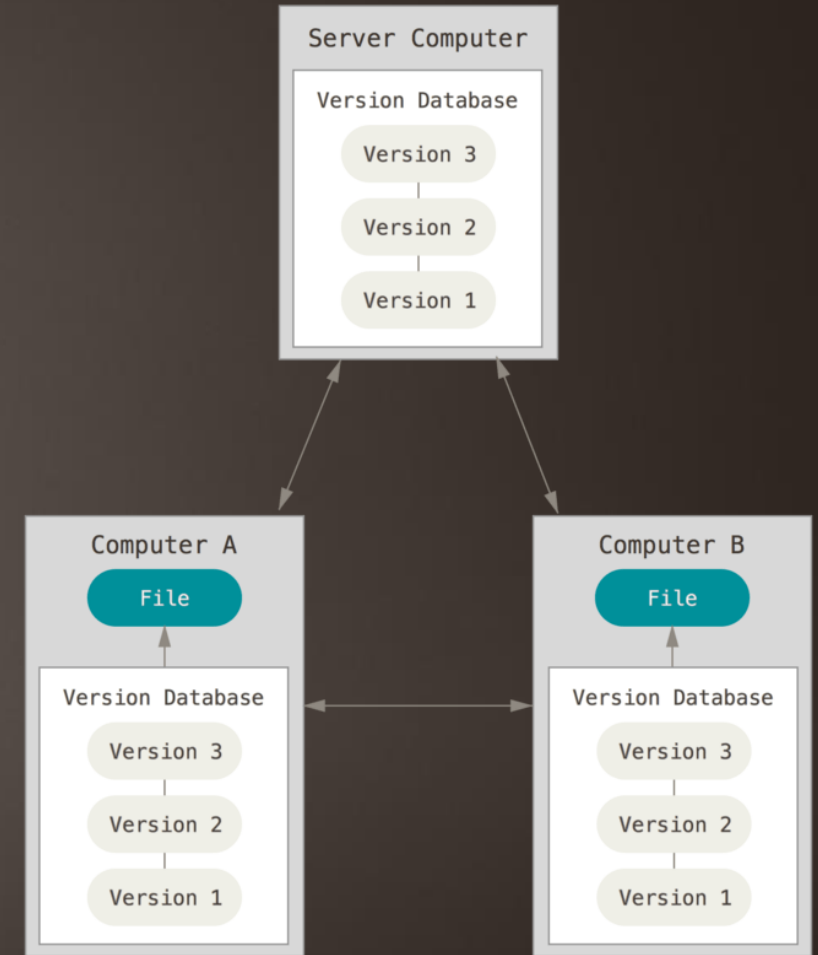
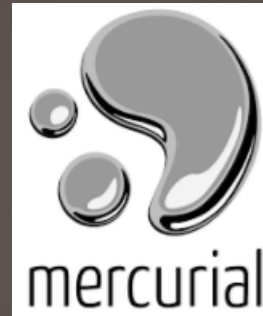


Centralized Version Control

CVS



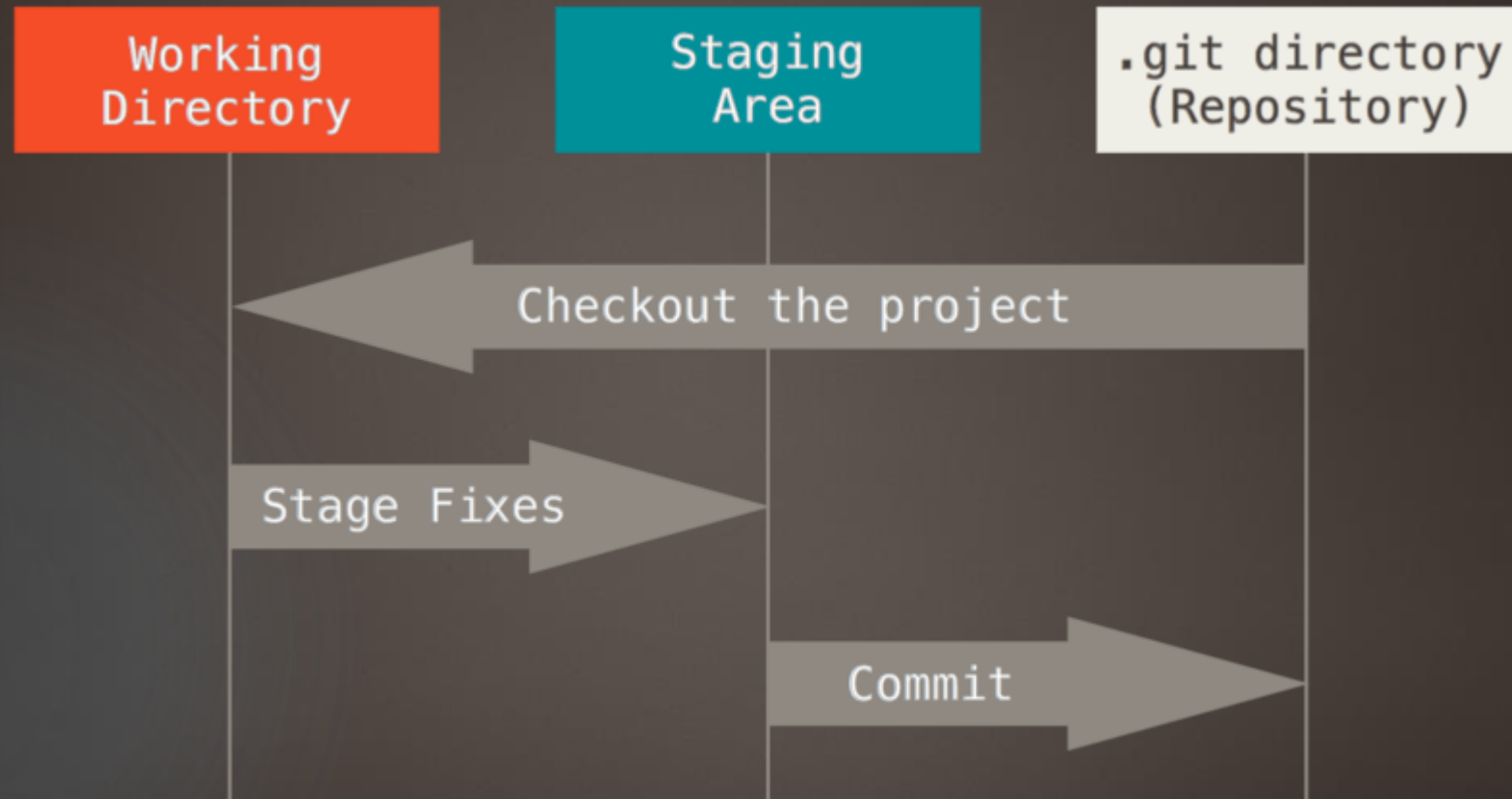
Distributed Version Control

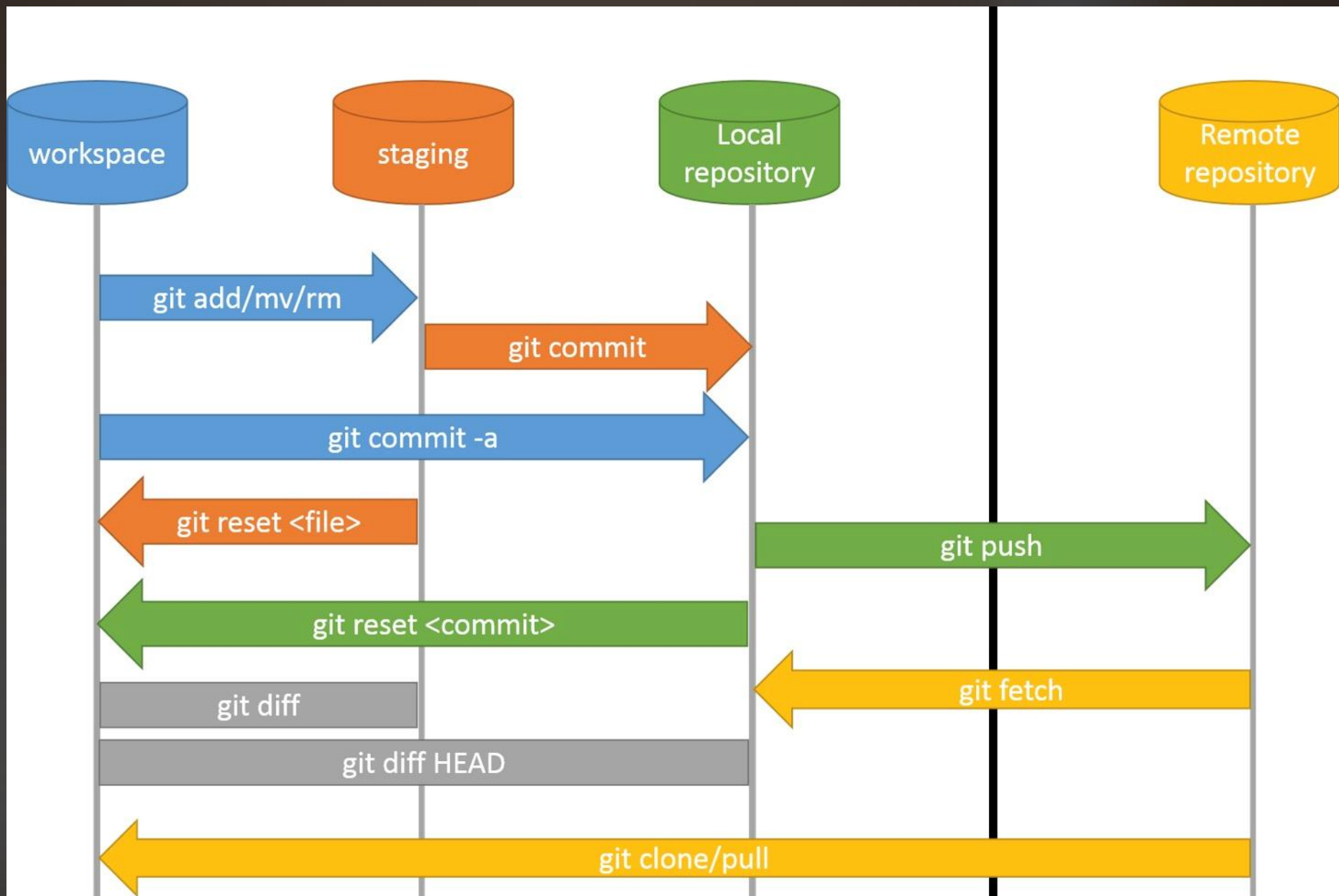


GIT มีข้อดีอย่างไร?

- ▶ สามารถทำงาน Offline ได้
- ▶ หาก Server กลางเสียหาย จะยังคงมี Backup ในเครื่องของทุกคน
- ▶ สามารถรวม Code กับเพื่อนร่วมงานได้แม้ Server กลางจะล่ม
- ▶ ทำงานได้เร็วกว่าแบบ Centralized เพราะสามารถดึงข้อมูลจากภายในเครื่องเราเองได้เลย
- ▶ แยก Branch เพื่อทำการทดลองเสี่ยงๆ ได้ง่ายมาก
- ▶ Merge Code ได้มีข้อผิดพลาดต่ำกว่าระบบ Centralized และมี Conflict น้อยกว่า

Git Basic





GIT Command

Command	ทำอะไร
git init	สร้าง repository เปล่าๆ
git clone <url>	ดาวน์โหลดไฟล์มาลงที่เครื่อง
git add <filename>	เพิ่มไฟล์เข้าระบบ
git commit -m "any comment you want here"	แก้ไขความเปลี่ยนแปลง
git reset	ลบไฟล์ที่ add เข้า staging ไปผิด
git status	ดูความเปลี่ยนแปลงที่เกิดขึ้นทั้งหมด
git branch	สร้าง Branch
git checkout	เปลี่ยน Branch, แก้ไขไฟล์กลับไปสถานะก่อนแก้ไข
git log	ดูประวัติการเปลี่ยนแปลงของไฟล์นั้นๆ
git push	ส่งการเปลี่ยนแปลงไปยังเครื่องอื่น

Lab 3: GIT

- ▶ ทดลองสร้าง Repository ใน gitlab.com
- ▶ ทดลอง Clone มาลงเครื่องตัวเอง
- ▶ ทดลอง Commit
- ▶ ทดลอง Push



Homework



การบ้าน #1 จงสร้างไฟล์ JSON จากตารางนี้

id	firstname	lastname	company	salary
1001	Luke	Skywalker	Walt Disney	40000
1002	Tony	Stark	Marvel	1000000
1003	Somchai	Jaidee	Love2work	20000
1004	Monkey D	Luffee	One Piece	9000000

เซฟไฟล์ชื่อ homework1.json



การบ้าน #2 จงเขียนโปรแกรม Javascript ดังนี้

- ▶ จงนำผลลัพธ์ JSON จากข้อ 1 มาใส่ตัวแปรชื่อว่า peopleSalary (ไม่ต้อง Read File ใดๆ แค่ Copy Paste มาทั้งก้อน) แล้วนำมาเขียนโปรแกรมวนลูปเพื่อสร้างเป็น JSON ตัวใหม่ที่หน้าตาเหมือนเดิมตัด property company ออกจากข้อมูลทุก row ในตาราง
- ▶ จงใช้ for...in ในการวนลูปเพื่อ get ค่า key มาจาก JSON ในข้อ 1 ห้ามระบุชื่อ field โดยตรงในโปรแกรม เช่น “firstname”, “lastname” โดยเด็ดขาด (หากไม่เคยใช้ จง Google ศึกษาวิธีใช้เอง)
- ▶ ที่บรรทัดสุดท้ายส่ง console.log(); เพื่อพิมพ์ค่า JSON ออกมา
- ▶ สร้างไฟล์ชื่อ homework2.js

การบ้าน #3 จงเขียนโปรแกรม Javascript ดังนี้

- ▶ จงนำการบ้าน #2 มาต่อยอดอีกครั้ง โดยแก้ไข salary จากเดิมเป็น Number เปลี่ยนเป็น Array เพื่อใช้ปรับฐานเงินเดือนเพิ่มขึ้นในอีก 3 ปี โดย ให้เงินเดือนขึ้นปีละ 10% เช่น [20000,22000, 24200] โดยทำให้ครบทุกคน
- ▶ ที่บรรทัดสุดท้ายส่ง console.log(); เพื่อพิมพ์ค่า JSON ออกมา
- ▶ สร้างไฟล์ชื่อ homework3.js

การบ้าน #4 จงสำรวจข้อมูล JSON ดังนี้

▶ จงเข้า Link

<https://drive.google.com/open?id=13ejCsIsXKkLx9XUTMRmxqOhMG0tNlcHS>

เพื่อ Copy ข้อมูล JSON ไปใส่ตัวแปรที่สร้างขึ้นและลองเขียน โปรแกรมเพื่อทำ 3 อย่างคือ

- ▶ นับว่ามีข้อมูลนับหน้าแต่ละสีคนละกี่สี โดยเก็บผลลัพธ์ลงตัวแปรชื่อ color เป็นชนิด object ดังนี้เช่น
{brown:4,green:3,blue:5}
 - ▶ นับว่ามีข้อมูลเพศชายกี่คน เพศหญิงกี่คน เก็บผลลัพธ์ลงตัวแปรชื่อ gender เป็นชนิด object เช่น {male: 1, female: 2}
 - ▶ แต่ละคนนับว่ามีเพื่อนทั้งหมดกี่คนและเพิ่ม properties “friendCount” เข้าไปในข้อมูลของคนแต่ละคนที่นับได้ด้วย
- ▶ ที่บรรทัดสุดท้ายส่ง console.log(); เพื่อพิมพ์ค่าข้างบน 3 อย่างออกมาทั้งหมด
- ▶ สร้างไฟล์ชื่อ homework4.js