

JavaScript 1

ผศ. ดร. เกี๊จแก้ว ธเนศวร

kejkaew.tha@mail.kmutt.ac.th

ตารางเรียน

สัปดาห์	วันที่	หัวข้อ/รายละเอียด	การบ้าน (ผู้ตรวจ)	ส่ง Assignment
1	8/8/2566	Introduction to Web programming, Basic HTML 1	Assignment 1 (KT,KN)	1
2	15/8/2566	Basic HTML 2 + CSS 1	Assignment 2 (KT,KN)	1,2
3	22/8/2566	CSS 2	Assignment 3 (KT)	1,2,3
4	29/8/2566	Responsive Web design	Assignment 4 (KT,KN)	2,3,4
5	5/9/2566	JavaScript 1 + DOM	Assignment 5 (KT,KN)	3,4,5
6	12/9/2566	JavaScript 2 + GitHub	Assignment 6 (KT)	4,5,6
7	19/9/2566	JavaScript 3	Assignment 7 (KT,KN)	5,6,7
8	26/9/2566	AJAX + JSON	Assignment 8 (KT,KN)	6,7,8
	3/10/2566	Node.js 1	Assignment 9 (KT)	7,8,9
9	10/10/2566	No class	ส่ง portfolio web page (KT)	
10	17/10/2566	Node.js 2	Assignment 10 (KT,KN)	8,9,10
11	24/10/2566	Node.js 3	Assignment 11 (KT,KN)	9,10,11
12	31/10/2566	Node.js 4 + MySQL	Assignment 12 (KT)	10,11,12
13	7/11/2566			10,11,12
14	14/11/2566			11,12
15	21/11/2566			
16	28/11/2566	ส่งโปรเจค (9:00-16:00)	KT	
		สอบปลายภาค ในตาราง	KT	

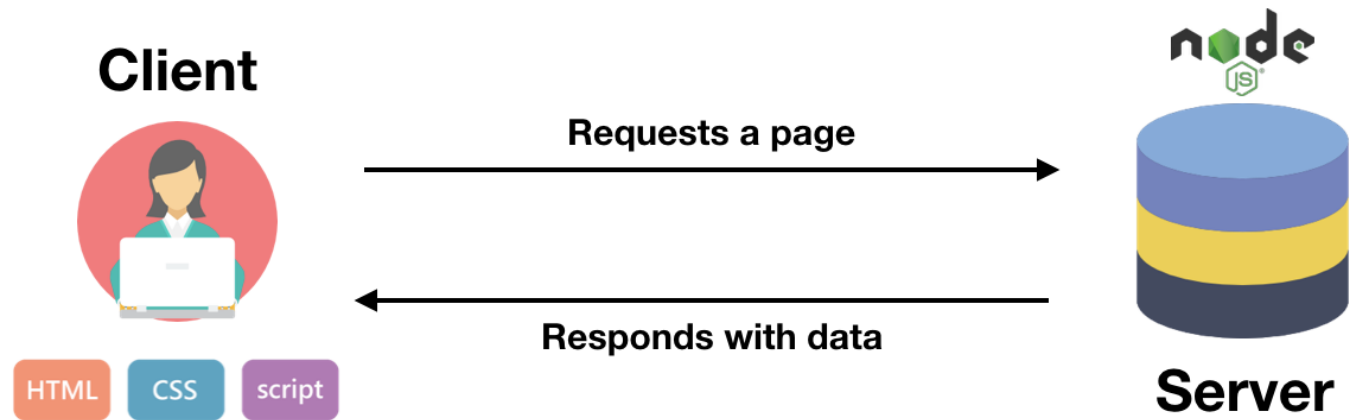
เราเรียนอะไรมาบ้างแล้ว

- How to write **content** for a webpage using **HTML**
- How to add **styles** to a webpage using **CSS** and linking a CSS file to an HTML file

Intro to JS

-
- เมื่อเรารู้ วิธีที่จะสร้าง content และ styles สำหรับ web page แล้ว ต่อมาเราจะทำการสร้าง behavior (interaction) ให้กับ web page ของเรา

Client-Side Scripting



- Client-side script: JS Code runs in browser after page is sent back from server. Often, this code manipulates the page or responds to user actions.

What is JavaScript?

- A programming language ("scripting language")
 - เริ่มต้นพัฒนาเมื่อปี 1995 เรียกว่า LiveScript
 - ใช้สำหรับทำ interactive web page
 - ใช้สำหรับเป็นเครื่องมือในการแก้ไขเนื้อหา HTML (Dynamic Web Page)
 - ใช้สำหรับควบคุม events ต่าง (เช่น ปุ่มกด การ load page)
- JS สามารถแสดงข้อมูลที่เกี่ยวข้องกับเครื่องคอมพิวเตอร์ของ user ได้ เช่น แสดงข้อมูลเกี่ยวกับ web browser ของ user
- JS ใช้สำหรับการคำนวณข้อมูลพื้นฐานบน web browser ของ user ได้ (เช่น check ว่า user กรอกชื่อ ใน textbox หรือยัง)
- ไม่เกี่ยวข้องกับ Java language

JavaScript vs. Java



+



= JavaScript

- JavaScript เป็น Interpreted
- More relaxed syntax and rules
 - Fewer and "looser" data types (ไม่ต้องประกาศ data type)
 - Errors often silent (few exceptions) (ไม่ค่อยแสดง Errors)
- JavaScript's key construct is the function rather than the object/class.
(สร้างโดยใช้ function มากกว่า object/class)
- Contained within a web page and integrates with its HTML/CSS content

HTML linking to a JavaScript file: <script>

- สิ่งที่ต้องมี

```
<script src= "main.js" ></script>
```

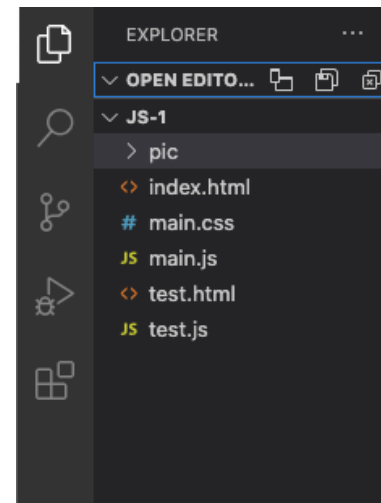
- Link สำหรับ script file ควรอยู่ใน html header
- JS codes จะต้องอยู่ใน file ที่นามสกุล .js
- JS codes สามารถอยู่ใน script tag ใน html header หรือ html body ก็ได้ แต่เป็นสิ่งที่ไม่ควรทำ
 - เราควรแยก content, presentation และ behavior ออกจากกัน

วิธีการรัน JavaScript

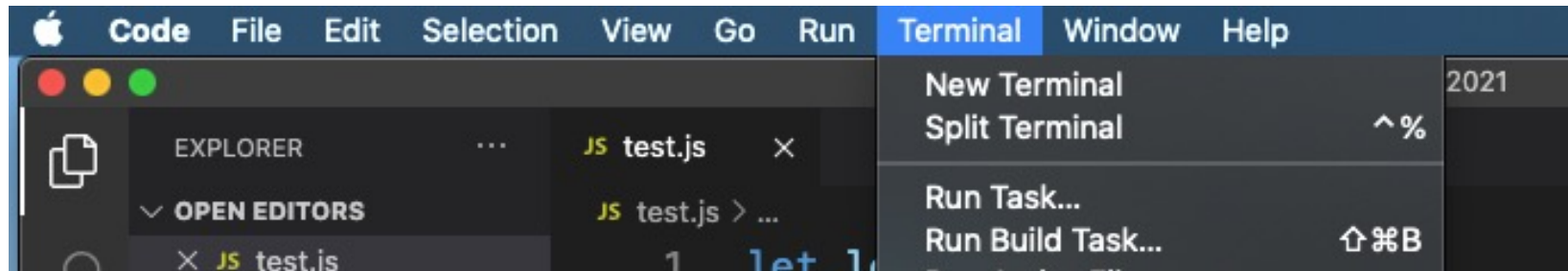
1. Run JS Code in Visual Studio Code
2. Run JS Code via web browser

1. Run JS Code in Visual Studio Code

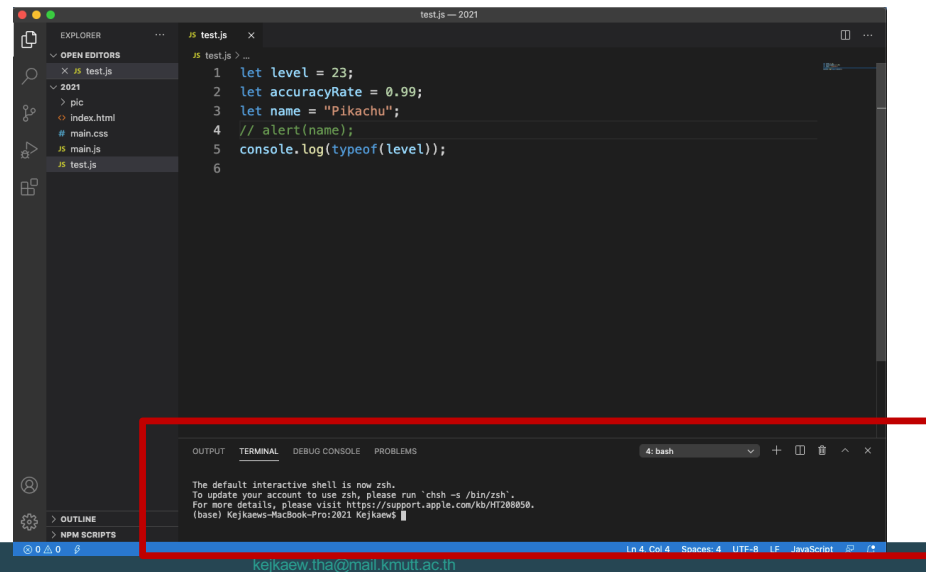
- install NodeJS บน Windows/Macbook จาก link นี้
<https://nodejs.org/en/download/>
- สร้าง folder และสร้าง JS file ใน folder นั้น โดยใช้ VS code หรือใช้ folder: JS-1 และ test.js ที่ อ. ให้



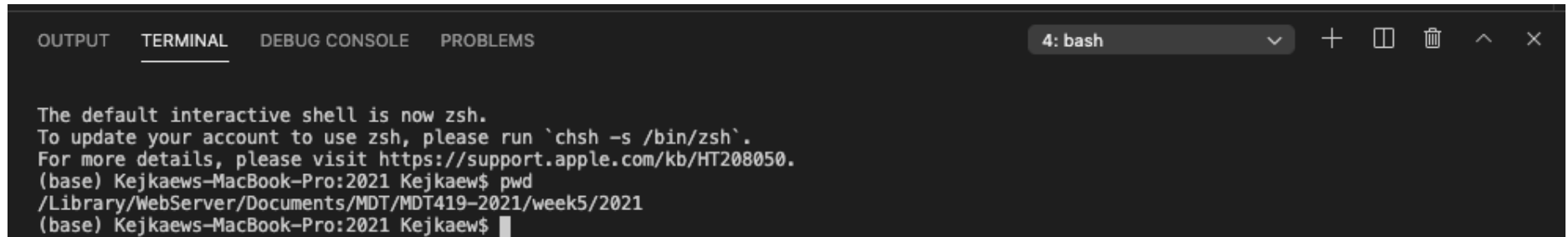
3. กดคำสั่ง Terminal ที่ menu bar เพื่อสร้าง terminal ใหม่



4. จะได้แบบนี้



5. เช็คว่า terminal ทำงานอยู่ใน directory ที่เราสร้างแล้วหรือยัง โดยพิมพ์ pwd (แล้วกด enter)



```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS  4: bash
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Kejkaews-MacBook-Pro:2021 Kejkaew$ pwd
/Library/WebServer/Documents/MDT/MDT419-2021/week5/2021
(base) Kejkaews-MacBook-Pro:2021 Kejkaew$
```

6. ถ้าไม่อยู่ให้ทำการเปลี่ยน working directory ไปที่ที่ควรอยู่ โดยใช้ cd ตามด้วย path และชื่อ ของ folder นั้น

7. ทำการรัน test.js โดยใช้คำสั่ง node test.js

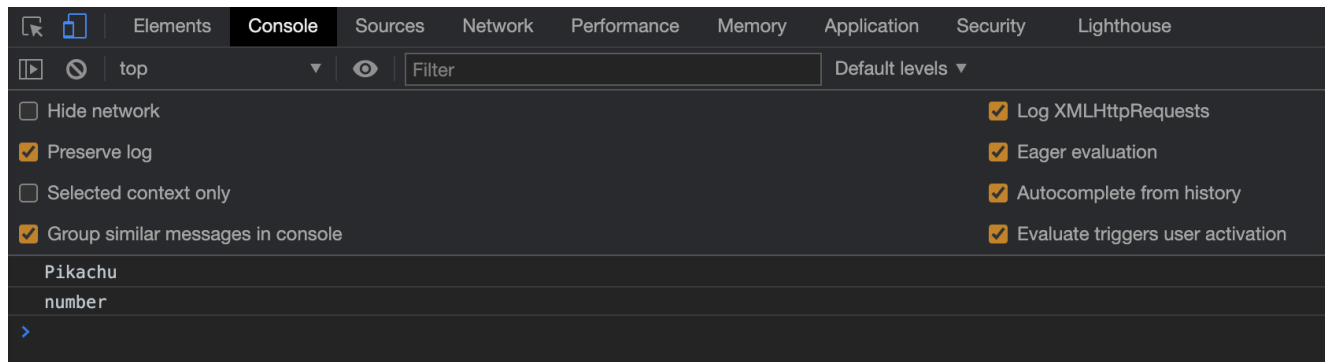
```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS

For more details, please visit https://support.apple.com/kb/HT208050.
(base) Kejkaews-MacBook-Pro:2021 Kejkaew$ pwd
/Library/WebServer/Documents/MDT/MDT419-2021/week5/2021
(base) Kejkaews-MacBook-Pro:2021 Kejkaew$ node test.js
Pikachu
number
(base) Kejkaews-MacBook-Pro:2021 Kejkaew$
```

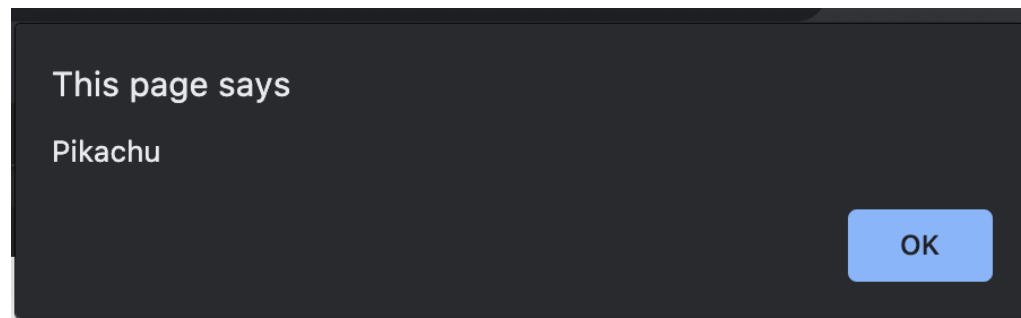
2. Run JS Code via web browser

- จาก test.html file ใน head tag จะมี

```
<script src="test.js"></script>
```
- เมื่อรัน test.html
- console.log แสดงผลผ่านทาง console ของ web browser (กด F12 ที่ Chrome)



- ลอง uncomment คำสั่ง `alert(name);`
- จะได้



Variables and types

- การประกาศตัวแปร เราจะใช้คำว่า `var` ในการเริ่มต้นสร้างตัวแปร (ตัวแปรเป็น case-sensitive นะ)
- JS ไม่มีการระบุประเภทของตัวแปรตอนประกาศ แต่มีการแบ่งประเภทของตัวแปร ("loosely-typed")
- ประเภทของตัวแปร:
Number, Boolean, String, Array, Object, Function, Null, Undefined
- ถ้าต้องการทราบว่าตัวแปรนี้เป็นประเภทอะไร จะใช้คำสั่ง `typeof`

ตัวอย่าง

```
var level = 23;  
var accuracyRate = 0.99;  
var name = "Pikachu";  
alert(name);  
console.log(typeof(level)); // number
```

Note: **var** has two types of scope, **Global Scope** and **Function Scope**.

Using **let** and **const** (ES6)

- **const** keyword to define a variable that cannot be reassigned (ค่าคงที่)
- **let** keyword to define a variable with restricted scope (ตัวแปรที่มีขอบเขตแบบ Block scope) ใช้ได้เหมือน var เกือบทุกอย่าง
- ใช้ได้กับ browser รุ่นใหม่ๆ (> 2015)

ตัวอย่าง

```
let level = 23;  
let accuracyRate = 0.99;  
let name = "Pikachu";  
alert(name);  
console.log(typeof(level));
```

```
const PI = 3.14159;  
PI = 3.14;  
// This will give an error  
PI = PI + 10;  
// This will also give an  
error
```

ความแตกต่างระหว่าง var และ let

- let และ const เป็น Block and function scope
- var เป็น Global and function scope

ตัวอย่างของ var

```
8  var carName = "Volvo";  
9  
10 myFunction();  
11 function myFunction() {  
12     var carName_2 = "BMW";  
13     console.log(carName);  
14     console.log(carName_2);  
15 }  
16 console.log(carName);  
17 console.log(carName_2);
```

Global scope

function scope
แสดงผลว่า Volvo
แสดงผลว่า BMW

แสดงผลว่า Volvo
ERROR!!!

ความแตกต่างระหว่าง var และ let

ตัวอย่าง Block scope (อยู่ใน {...})
(var ไม่มี block scope)

```
20 // block scope
21 {
22     var x = "Volvo";
23     let y = "BMW";
24 }
25 console.log(x);
26 console.log(y);
```

แสดงผลว่า Volvo
ERROR!!!

ประกาศตัวแปรซ้ำ

- ถ้าประกาศตัวแปรซ้ำใน block ค่าของตัวแปรที่ใช้ let จะไม่เปลี่ยนแปลง เช่น

```
28 //Redeclaring variables
29 var x = 5;
30 let y = 10;
31 {
32     var x = 2;
33     let y = 20;
34 }
35 console.log(x);
36 console.log(y);
```

แสดงผลว่า 2

แสดงผลว่า 10

Number type

- Integers และ real numbers คือ ตัวแปรประเภทเดียวกัน นั่นคือ number
- เครื่องหมายการกระทำ : + - * / % ++ -- = += -= *= /= %=
- ลำดับการกระทำเหมือน programming ภาษอื่น
- JS สามารถเปลี่ยนประเภทของตัวแปร ได้เอง เช่น "2" * 3 = 6
- ตัวอย่าง

```
var enrollment = 99;
```

```
var mediangrade = 2.8;
```

```
var credits = 5 + 4 + (2 * 3);
```

String type

- Methods: charAt, charCodeAt, fromCharCode, indexOf, lastIndexOf, replace, split, substring, toLowerCase, toUpperCase
- ตัวอย่าง Methods: `charAt` returns a one-letter String (there is no `char` type)
- `length` ใช้สำหรับหาความยาวของ string
- Concatenation ใช้เครื่องหมาย `+` สำหรับเชื่อม string 2 ตัวเข้าด้วยกัน เช่น `1 + 1` เท่ากับ `2`, แต่ `"1" + 1` จะเท่ากับ `"11"`

ตัวอย่าง string type

```
var mName = "Media Technology" ;  
var sName = mName.substring(0, mName.indexOf(" "));  
var len = mName.length; // 16  
console.log(sName);  
console.log(len);
```

More about Strings

- To convert between numbers and Strings:

```
var magicNum = parseInt("42");
```

```
var mystery = parseFloat("Am I a number?");
```

```
console.log(typeof(magicNum));
```

-
- To access characters of a String, use [index] or charAt:

```
var puppyCount = "10"+ " puppies!";
```

```
var firstLetter = puppyCount[0];
```

```
console.log(typeof(firstLetter));
```

Comments (same as Java)

- `//` single-line comment
- `/*` multi-line comment `*/`
- HTML: `<!-- comment -->`
- CSS/JS: `/* comment */`
- Java/JS: `// comment`

Logical Operators

- Relational: > < >= <=
- Logical: && || !
- Equality: == != === !==
- ส่วนใหญ่แล้ว Logical operators ทำการเปลี่ยนประเภทของตัวแปรให้เหมือนกันก่อนทำการเปรียบเทียบ จึงทำให้ ข้อความต่อไปนี้เป็นจริง :

5 < "7"

42 == 42.0

"5.0" == 5

Logical Operators

- ยกเว้น `===` and `!==` จะทำการเปรียบเทียบทั้งประเภทของตัวแปรและค่าของตัวแปร จึงทำให้ `"5.0" === 5` is false ดังนั้น ถ้าประเภทของตัวแปรต่างกันจะเป็น false ทันที

Boolean Type

```
var iLikeJS = true;
```

```
var ieIsGood = "IE6" > 0; // false
```

- Any value can be used as a Boolean
 - "false" values ได้แก่ 0, 0.0, NaN, "", null, and undefined
 - "truth" values ทุกตัวที่นอกเหนือจากด้านบน

Special Values: null and undefined

```
var foo = null;
```

```
var bar = 9;
```

```
var blue;
```

```
/* At this point in the code, foo is null, bar is  
9, blue is undefined */
```

- undefined: ยังไม่มีการ assign ค่าตัวแปร
- null: มีการ assign ค่าให้กับตัวแปร แต่ใส่เป็น null

if/else Statements (same as Java)

```
if (condition) {  
    statements;  
} else if (condition) {  
    statements;  
} else {  
    statements;  
}
```

ตัวอย่าง

```
var a = 5;  
if (a == 5){  
    console.log("a = 5");  
}else{  
    console.log("a= ??");  
}
```

Defining functions

- การสร้าง function

```
function functionName() {  
    statement;  
    statement;  
    ...  
    statement;  
}
```

ตัวอย่างเช่น

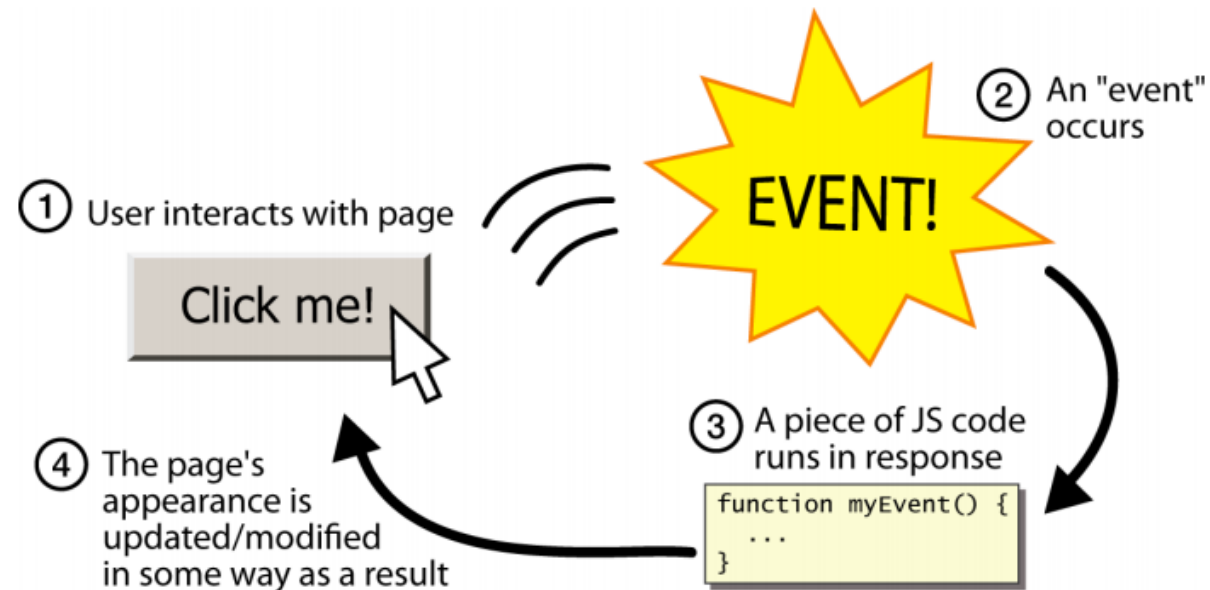
```
function myFunction() {  
    alert("Hello!");  
    alert("Your browser says hi!");  
}
```

- *//วิธีการเรียกใช้*

```
myFunction();
```

Event-Driven Programming

- Event-Driven Programming: writing programs driven by user events



Event Handlers (index.html)

- <element attributes **onclick**="function();">...

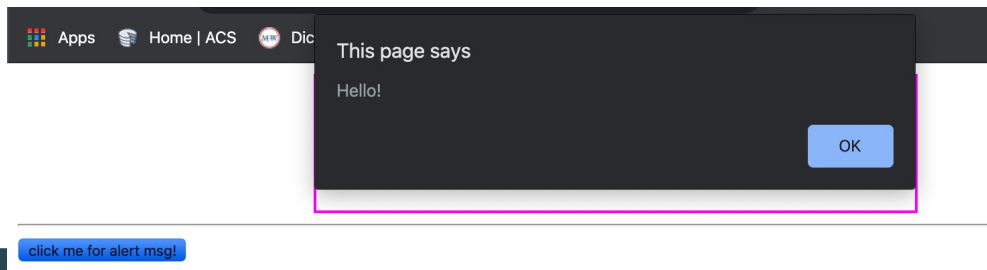
```
function myFunction(){  
    alert("Hello!");  
}
```

- <div **onclick**="myFunction();">Click me! </div>
- JavaScript functions สามารถนำมาใช้สำหรับ event handlers
- เมื่อเราทำบางอย่างกับ element นั้น function จะเริ่มทำงานทันที
- **onclick** เป็น HTML attributes ตัวหนึ่ง ที่นิยมใช้ใน event handlers

Buttons: <button>

- ใช้สำหรับสร้าง ปุ่ม และใส่ข้อความหรือรูปภาพในปุ่ม
- และทำการเชื่อมปุ่ม กับ เหตุการณ์ (event) ใน JS function โดยใช้ onclick attribute

```
<button onclick="myFunction()"> click me for alert msg!</button>  
<hr>
```



```
function myFunction(){  
    alert("Hello!");  
}
```

<input>

- Input tag ใช้สำหรับสร้าง User interface
- type can
be button, checkbox, file, hidden, password, radio, reset, submit, text,
...

```
<input type="text" name="input_val" value="MDT419" />  
<input type="submit" value="Submit Homework!" /><br>  
<hr>
```

<input> Text Fields

- input attributes: disabled, maxLength, readonly, size, value
- size attribute controls onscreen width of text field
- maxlength limits how many characters the user is able to type into the field

```
Student ID <input type="text" size="10" maxlength="8" /> <br>  
Password <input type="password" size="16" /> <br>  
<input type="submit" value="Log In!" /><br>  
<hr>
```

Student ID

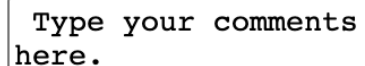
Password

Log In!

Text boxes: <textarea>

- Initial text is placed inside textarea tag (optional)
- optional **readonly** attribute means text cannot be modified

```
<textarea > Type your comments here. </textarea>  
<hr>
```

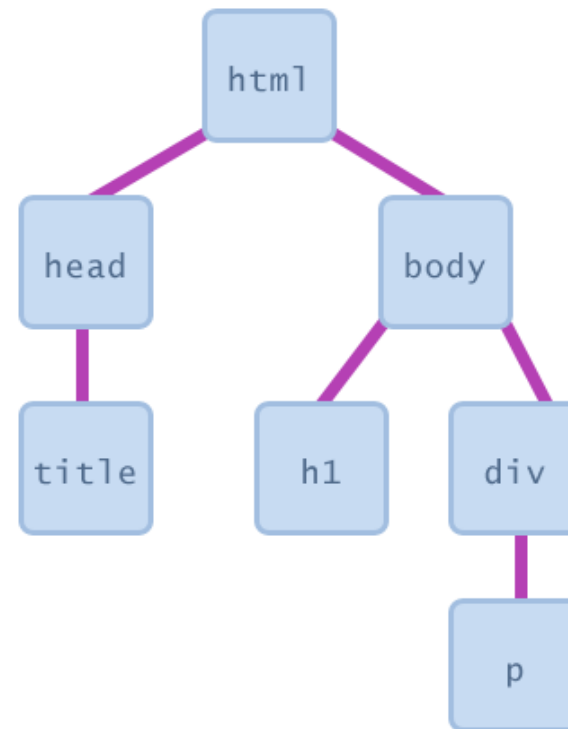


Type your comments
here.

The DOM

- Tree shaped structure ที่สร้างจาก HTML elements ของแต่ละ web page สามารถเข้าถึงได้โดยใช้ JS

```
<html>
  <head>
    <title> ... </title>
  </head>
  <body>
    <h1> ... </h1>
    <div>
      <p> ... </p>
    </div>
  </body>
</html>
```



Document Object Model (DOM)

- a set of JavaScript objects that represent each element on the page
- JS code สามารถตรวจสอบสถานะของ แต่ละ element ได้ เช่น ใน checkbox, JS สามารถ check ได้ว่า box ถูก checked หรือยัง
- สามารถเปลี่ยนเนื้อหาของ HTML ได้
- สามารถเปลี่ยน style ได้

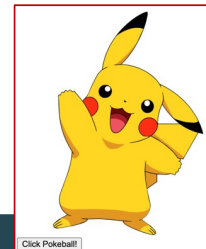
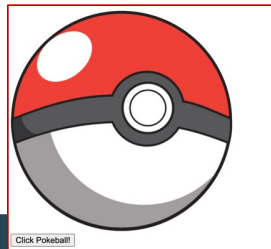
Accessing an Element: document.getElementById

- `var name = document.getElementById("id");`
- คำสั่ง `document.getElementById` จะ return DOM object ของ element ที่มี id นั้น
- ตัวอย่างเช่น

document.getElementById: Example

```
<div>
  
</div>
<button onclick="changeImage();">Click Pokeball!</button>
<hr>
```

```
function changeImage() {
  var pokeballImg = document.getElementById("img01");
  pokeballImg.src = "pic/pikachu.png";
}
```



innerHTML property

- ทุก DOM elements จะมี Property ที่ชื่อว่า innerHTML ใช้สำหรับเปลี่ยน แก้ไข หรือเรียกดู content ของ element นั้น
(ใน HTML body tag)

```
<a id="tw-link" href="http://www.twitter.com">Facebook</a>
```

(เกิดอะไรขึ้น???)

```
var link = document.getElementById("tw-link");  
link.innerHTML = "twitter";
```

[Facebook](#)

```
✖ ▶ Uncaught TypeError: Cannot set property 'innerHTML' of null    main.js:13  
    at main.js:13
```

The window.onload event

```
function functionName() {  
    // put code to initialize the page here  
}
```

```
window.onload = functionName;
```

- เพื่อบอกว่า browser window ว่า ให้รัน **functionName** หลังจากที่ web page โหลดเสร็จแล้ว

ตัวอย่าง

(ใน HTML body tag)

```
<a id="tw-link" href="http://www.twitter.com">Facebook</a>
```

(ใน JavaScript file)

```
window.onload = twitterLink;  
function twitterLink(){  
    var link = document.getElementById("tw-link");  
    link.innerHTML = "twitter";  
}
```

[twitter](#)

Week 5: Classroom game

- ทำ classroom game ใน LEB2 ก่อนสัปดาห์หน้า

Assignment 5 : Blogger

- Download assignment5.zip file
- ทำการสร้าง Blog จาก template ที่อาจารย์ให้ โดย
 - เมื่อเรากดปุ่ม Post ครั้งที่ 1 เราจะ post ข้อความ ลงใน id topic1
 - เมื่อ เรากดปุ่ม Post ครั้งที่ 2 เราจะ reply post ลงใน id comment1
 - เมื่อ เรากดปุ่ม Post ครั้งที่ 3 เราจะ reply post ลงใน id comment2
 - เมื่อเรากดปุ่ม Clear โปรแกรมจะลบข้อความที่ topic และ comment ทั้งหมด
- เราสามารถใช้ `document.getElementById("text1").value` เพื่อเอาข้อความที่เราพิมพ์ออกมา post ใน id ต่างๆ
- การบ้าน ทำการตกแต่ง Blog page ของเรา ให้สวยงาม