

Week 6: JavaScript 2

ผศ. ดร. เกื้อแก้ว 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	

วันนี้เราจะเรียนเรื่อง...

- Math object
- Loops
- Array
- Unobtrusive JavaScript
- Form
- GitHub

Math object

- Methods: abs, ceil, cos, floor, log, max, min, pow, random, round, sin, s
qrt, tan

```
1 console.log(Math.PI);  
2 console.log(Math.round(4.7));  
3 let a = Math.pow(8, 2);  
4 console.log(a);
```

- Properties: E, PI

for loop (same as Java)

```
for (initialization; condition; update) {  
    statements;  
}
```

```
7  var sum = 0;  
8  for (var i = 0; i < 100; i++) {  
9      sum = sum + i;  
10 }  
11 console.log(sum);
```

while loops (same as Java)

```
while (condition) {  
    statements;  
}
```

```
14  var sum = 0;  
15  var i = 0;  
16  while (i < 100){  
17      sum = sum+i;  
18      i += 1;  
19  }  
20  console.log(sum);
```

แบบฝึกหัด

1. คำนวณผลรวมตั้งแต่ 0 ถึง 99 ที่หารด้วย 2 ลงตัวโดยใช้ while loop
2. คำนวณผลรวมตั้งแต่ 0 ถึง 99 ที่หารด้วย 3 ลงตัวโดยใช้ for loop

Arrays

```
var array_name = [item1, item2, ...];
```

- length property (grows as needed when elements are added)
- Array indexes start with 0.
- สามารถประกาศตัวแปร Array ได้ 2 แบบ

Creating an Array

แบบที่ 1

```
23 var name = [];  
24 var names = [5,8,9,7,4,10];  
25 console.log(names[5]);  
26 names[5] = 6;  
27 console.log(names);
```

แบบที่ 2

```
29 var names = new Array(5,8,9,7,4,10)  
30 console.log(names[4]);
```

Array methods

- Array serves as many data structures: list, queue, stack, ...
- Methods: concat, join, pop, push, reverse, shift, slice, sort, splice, toString, unshift
- push and pop เพิ่ม/ลบ สมาชิกใน array ที่ตัวสุดท้าย
- unshift and shift เพิ่ม/ลบ สมาชิกใน array ที่ตัวแรก
- shift and pop จะ return ค่า สมาชิกที่ถูกลบออกมา

ตัวอย่าง

```
33 var fruits = ["Banana", "Orange", "Apple", "Mango"];
34 console.log(fruits);
35
36 fruits.pop();
37 console.log(fruits);
38
39 fruits.push("Mango");
40 console.log(fruits);
41
42 fruits.shift();
43 console.log(fruits);
44
45 fruits.unshift("Banana");
46 console.log(fruits);
47
48 fruits.sort();
49 console.log(fruits);
```

Array as an object

- In JavaScript, **arrays** use numbered indexes.
- In JavaScript, **objects** use named indexes.
- Two ways to initialize an object
- ตัวอย่าง 1

```
51 // array object
52 var person = [];
53 person["firstName"] = "John";
54 person["lastName"] = "Doe";
55 person["age"] = 46;
56 var x = Object.keys(person).length;
57 console.log(x);
```

ตัวอย่าง 2

```
59  var person = {  
60      firstName: "John",  
61      lastName: "Doe",  
62      age: 46  
63  };  
64  console.log(person);
```

Accessing Object Properties

- เราสามารถ access object properties ได้ 2 แบบ
- `objectName.propertyName`
- `objectName["propertyName"]`

```
59  var person = {  
60      firstName: "John",  
61      lastName: "Doe",  
62      age: 46  
63  };  
64  console.log(person);  
65  console.log(person.lastName);  
66  console.log(person["lastName"]);
```

Object methods

- Objects สามารถมี **methods** ได้
- Methods คือการทำงานบางอย่างที่อยู่ภายใน object เดียวกัน
- Methods สามารถสร้างได้โดยใช้ function
- ตัวอย่างเช่น

```
69 var person = {  
70     firstName: "John",  
71     lastName: "Doe",  
72     age: 46,  
73     fullName : function() {  
74         return this.firstName + " " + this.lastName;  
75     }  
76 };  
77 console.log(person.fullName())
```

this keyword คือการอ้างถึง object ที่ function นั้นอาศัยอยู่
ในตัวอย่างนี้ this คือ person object

For/of (ES6)

- for/of statement คือการวนลูปใน object หรือ array โดยไม่ต้องใช้ index หรือ key
- ตัวอย่างเช่น

```
79 // for of
80 let cars = ["BMW", "Volvo", "Mini"];
81 let text = "";
82
83 for (let x of cars) {
84     text += x + " ";
85 }
86 console.log(text);
```

Unobtrusive JavaScript

- คือการแยก JavaScript ออกจาก HTML
- JavaScript event code ที่เราก่อนหน้านี้จะเรียกว่า obtrusive เป็นสิ่งที่ไม่ควรทำ
- เราจะใช้ JS DOM ในการเชื่อม event handler กับ JS function
- allows separation of web site into 3 major categories:
 - content (HTML) - what is it?
 - presentation (CSS) - how does it look?
 - behavior (JavaScript) - how does it respond to user interaction?

Obtrusive event handlers (bad)

- เป็นสิ่งที่ไม่ควรทำ (HTML is cluttered with JS code)
- goal: เอา JavaScript code ออกจาก HTML body

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

```
function myFunction()  
|   alert("Hello!")  
}  
}
```

event handler in JavaScript code

- `objectName.onevent = function;`

```
<button id="clickme"> click me for alert msg!</  
button>  
<hr>
```

```
var clickButton = document.getElementById("clickme");  
clickButton.onclick = myFunction;
```

เกิดอะไรขึ้น???

An unobtrusive event handler

```
<button id="clickme"> click me for alert msg!</button>  
<hr>
```

```
function myFunction(){  
    alert("Hello!");  
}  
  
function pageLoad(){  
    var clickButton = document.getElementById("clickme");  
    clickButton.onclick = myFunction;  
}  
  
window.onload = pageLoad;
```

HTML Forms

- ใช้สำหรับเก็บข้อมูลของ user

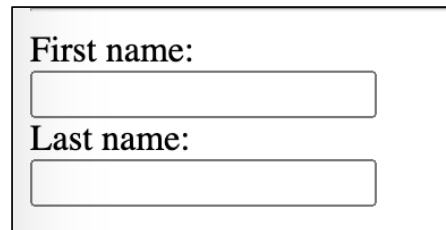
<form>

...

form elements

...

</form>



First name:

Last name:

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Forms: Submit Button

- `<input type="submit">` ใช้สำหรับสร้างปุ่มกดเพื่อทำการจัดการกับข้อมูลใน form

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
  <br>
  <input type="submit" value="Submit">
</form>
```

First name:	<input type="text"/>
Last name:	<input type="text"/>
	<input type="submit" value="Submit"/>

Forms: Action Attribute

- action attribute ใช้เพื่อบอกว่า ข้อมูลใน form จะถูกส่งไปที่ใด

```
<form action="index.html">
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
  <br>
  <input type="submit" value="Submit">
</form>
```


Forms: Name Attribute

- Input field แต่ละตัวจะต้องมี **name attribute** เพื่อการส่งข้อมูล ถ้า input field ไม่มี name attribute ข้อมูลจะไม่ถูกส่ง

```
<form action="index.html">
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
  <br>
  <input type="submit" value="Submit">
</form>
```

Forms: Grouping Form Data

- `<fieldset>` ใช้สำหรับรวมกลุ่ม data ใน form
- `<legend>` ใช้สำหรับสร้าง caption ให้กับ กลุ่มในแต่ละ form

```
<form action="index.html">
  <fieldset>
    <legend>Profile</legend>
    First name:<br>
    <input type="text" name="firstname"><br>
    Last name:<br>
    <input type="text" name="lastname">
    <br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```



Profile

First name:

Last name:

Submit

JavaScript Form Validation

Attribute	Description
disabled	Specifies that the input element should be disabled
max	Specifies the maximum value of an input element
min	Specifies the minimum value of an input element
pattern	Specifies the value pattern of an input element
required	Specifies that the input field requires an element
type	Specifies the type of an input element

ตัวอย่าง

```
<form action="index.html">
  <fieldset>
    <legend>Profile</legend>
    First name:<br>
    <input type="text" name="firstname" required
    ><br>
    Last name:<br>
    <input type="text" name="lastname">
    <br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Profile

First name:

Last name:

Submit

onsubmit Event

- ทำการเรียก JavaScript function เมื่อกดปุ่ม submit
- จากตัวอย่างเดิม
 - ทำการสร้าง id = "myForm" ที่ form ใน HTML

```
window.onload = pageLoad;
function pageLoad(){
    var form = document.getElementById("myForm");
    form.onsubmit = myFormfunction;
}
function myFormfunction(){
    alert("OK");
}
```

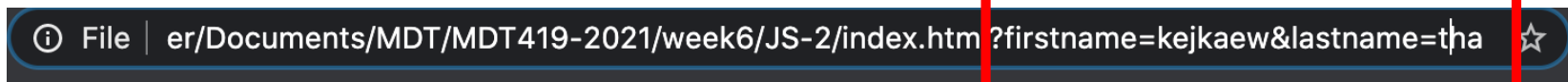
Form elements Collection

- document.forms จะ return element ที่อยู่ใน <form>
- เราจะ เรียกข้อมูลที่อยู่ใน form โดยใช้ name attribute
- จากตัวอย่างเดิม
 - เราทำการเพิ่ม name = “myForm” ใน form element

```
function myFormfunction(){  
    alert("OK");  
    var x = document.forms["myForm"]["firstname"];  
    alert(x.value);  
}
```

URLSearchParams()

- URL parameters หรือเรียกอีกอย่างว่า Query string parameter คือ การส่งข้อมูลสั้นๆ จาก webpage ผ่านทาง URL เพื่อส่งให้อีก webpage หนึ่ง หรือจะส่งเป็นการส่งจาก server ก็ได้
- ตัวอย่างเช่น จากตัวอย่างที่แล้ว จะได้



File | er/Documents/MDT/MDT419-2021/week6/JS-2/index.htm?firstname=kejkaew&lastname=tha

URL parameters

URLSearchParams()

- เราสามารถใช้ URLSearchParams() API ในการเรียกใช้ parameter ใน URL ได้
- URLSearchParams.get(*parameter_name*) ใช้สำหรับเรียกดูค่าของ *parameter_name* ตัวแรกได้
- ตัวอย่างเช่น


```
113 // URL parameter
114 const queryString = window.location.search;
115 console.log(queryString);
116
117 const urlParams = new URLSearchParams(queryString);
118 const firstname = urlParams.get('firstname')
119 console.log(firstname);
120
121 const lastname = urlParams.get('lastname')
122 console.log(lastname);
```

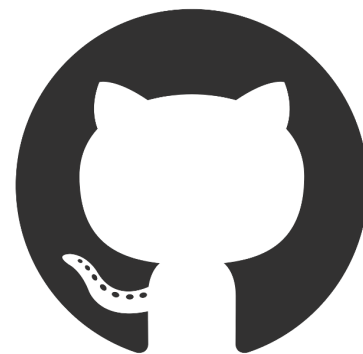
Output ที่จะได้จาก **console.log**

?firstname=kejkaew&lastname=tha

kejkaew

tha

GitHub



What is GitHub

- Server สำหรับให้บริการ Git
- พัฒนามาจาก Git (version control systems)
 - Git มีลักษณะเป็นแบบ command line tool
 - GitHub มีลักษณะเป็นแบบ Web-based graphical interface
- เก็บความเปลี่ยนแปลงของไฟล์ในโปรเจค
- สามารถสร้าง website for your projects ได้

คำที่ควรรู้ใน GitHub

- Repository = Project สำหรับใช้งาน Git
- Clone = นำ Repository จาก Remote มาลงเครื่อง
- Fork = เป็นการ copy repository มาไว้ใน git ของเรา
- Commit = ทำการ Backup เก็บไว้ใน Version control systems
- Pull request = เวลาที่มี Commit จากคนอื่นเข้ามาที่เรา จะเกิด pull request
- Conflict = เกิดจากการแก้ไขโค้ดที่เดียวกันและไม่สามารถรวมกันได้
- Merge Commit = เมื่อมีการแก้ไข และ commit จะต้องทำการ merge ก่อน commit นั้น
- Fetch = อัปเดตและดู History ทั้งหมดที่อยู่บน Remote

ตัวอย่าง

- สมัครสมาชิก GitHub
- สร้าง repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner



Repository name *

/ mdt-test1 ✓

Great repository names are short and memorable. Need inspiration? How about **vigilant-carnival**?

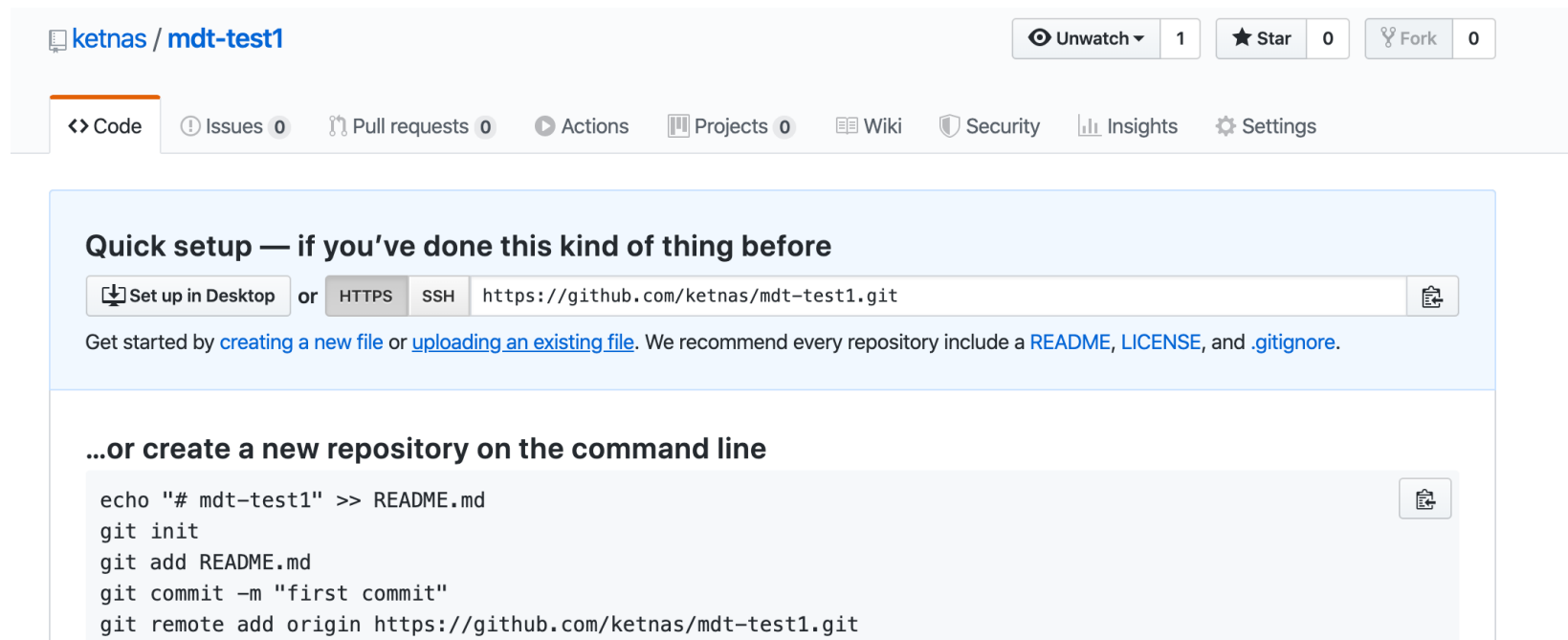
Description (optional)



Public

Anyone can see this repository. You choose who can commit.

■ Click at uploading an existing file





ketnas / mdt-test1

Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

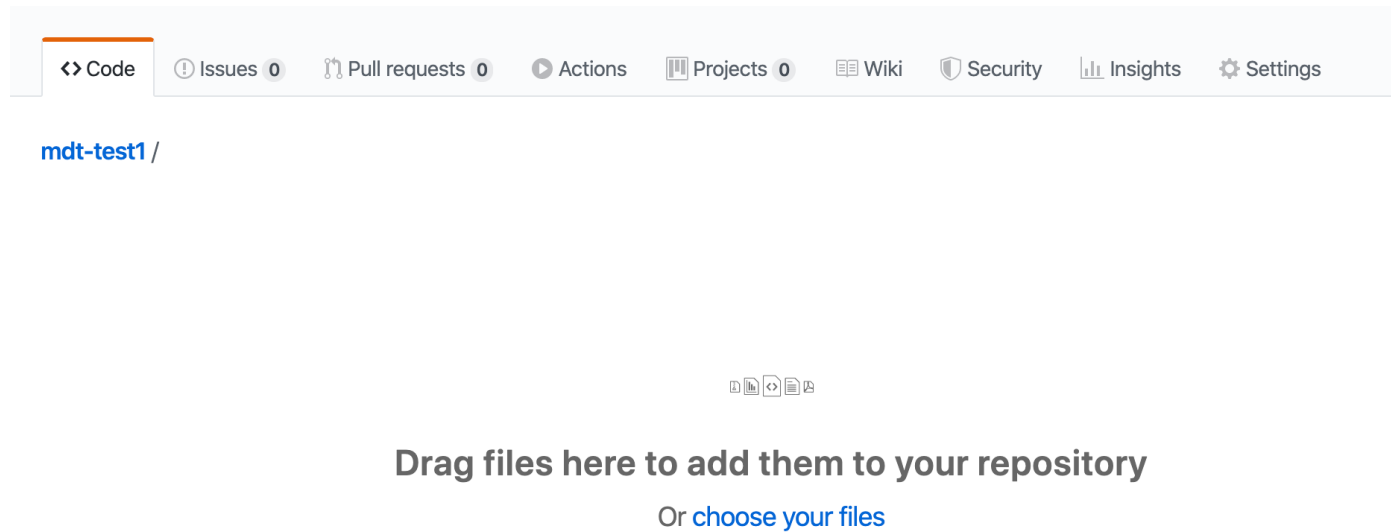
 Set up in Desktop or **HTTPS** **SSH** 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# mdt-test1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/ketnas/mdt-test1.git
```

■ ลาก index.html file มาวาง



■ กด commit



Commit changes


Add files via upload

Add an optional extended description...

Commit changes

Cancel

■ กด setting

 ketnas / mdt-test1

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

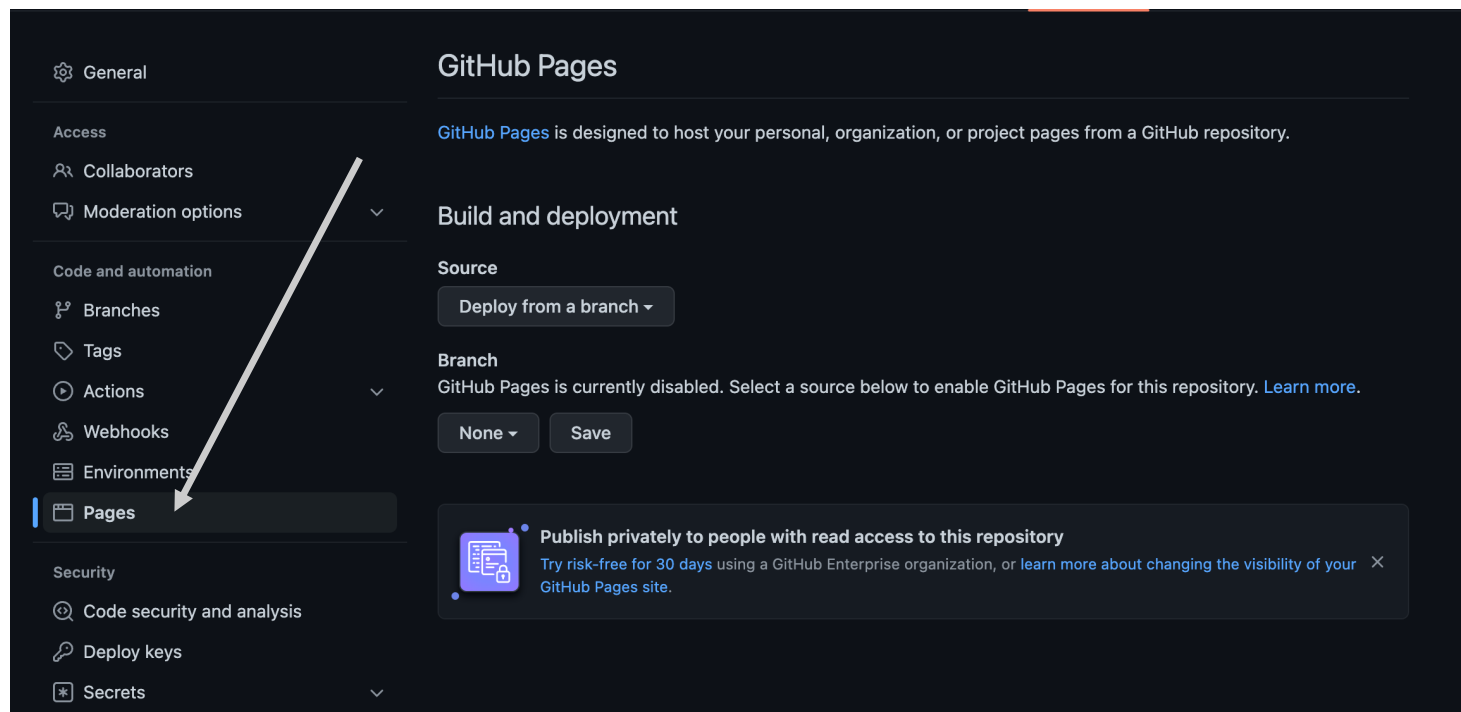
Settings

No description, website, or topics provided.

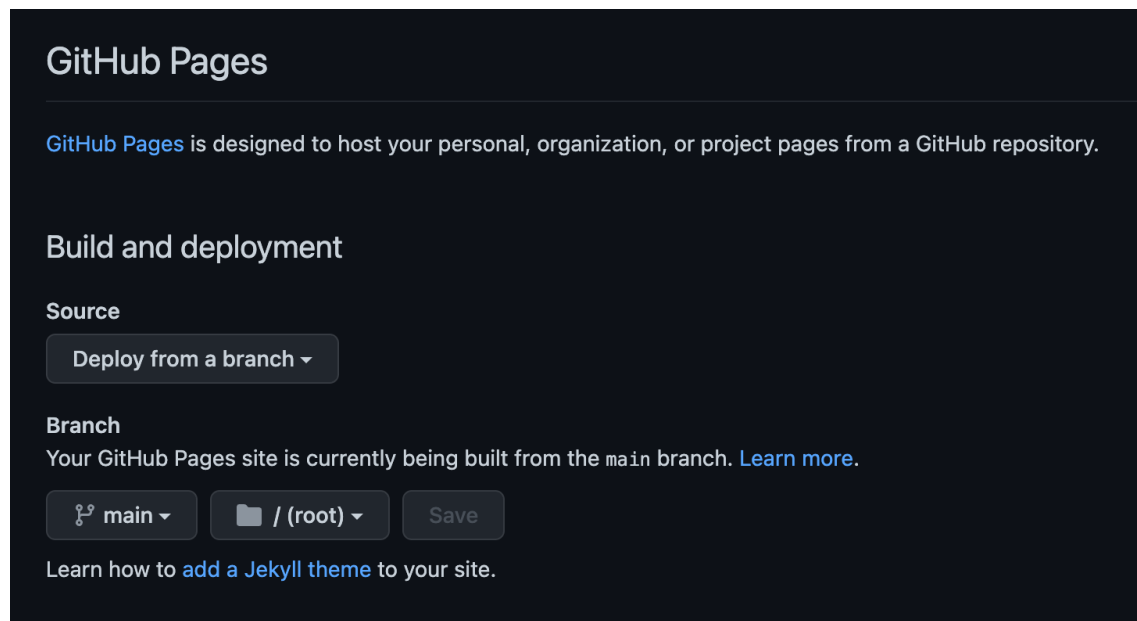
[Manage topics](#)

Edit

■ ไปที่ตรง Pages



- ที่ source เลือก Deploy from a branch และเลือก main Branch และกด save
- จะได้ URL เป็นชื่อ username ของเรา เช่น ketnas.github.io และตามด้วยชื่อ repository เช่น <https://ketnas.github.io/mdt-test1/>



Assignment 6: register and login

- Download assignment6.zip file
- ทำการสร้างระบบ register และ login โดยมีรายละเอียดดังนี้
 - ในหน้า register ต้องมีการ ตรวจสอบว่า ผู้ใช้กรอกข้อมูลครบทุกช่องหรือยัง ถ้ายังจะไม่มีมีการเปลี่ยนหน้า ไป login page
 - ในหน้า register ต้องมีการ ตรวจสอบว่า ผู้ใช้กรอก password กับ retype password ตรงกันหรือไม่ ถ้าไม่ตรงกัน จะไม่มีการเปลี่ยนหน้า ไป login page
 - ในหน้า login ต้องมีการตรวจสอบว่า ผู้ใช้กรอก username กับ password ตรงกับที่ register ไว้หรือไม่ ถ้าไม่ตรงให้ แสดง alert ให้ user กรอกใหม่
- การบ้าน ทำการตกแต่ง register กับ login ของเรา ให้สวยงาม

Register

* require

First name*: Last name*:

Gender*: ☐ Male ☐ Female ☐ Other

Birthday*:

Email address*:

Username*:

Password*:

Retype Password*:

Week 6 : Classroom game

- อย่าลืมทำ classroom game ใน LEB2
- สัปดาห์หน้ามีสอบ Quiz นะคะ

References

- <https://www.w3schools.com/>
- *Web Programming Step by Step* by Marty Stepp, Jessica Miller, and Victoria Kirst, <http://www.webstepbook.com/>