

JavaScript 3

ดร. เกี้ยว ชนะวาร

kejkaew.tha@mail.kmutt.ac.th

ส่ง portfolio web page

- กำหนดส่งวันที่ 12 มี.ค. 2563 ก่อน 23:50 น.
- ส่งโดยอัปโหลด zip file ลง LEB2 และพิมพ์ URL link ของ website ของนศ (เช่น GitHub) ต่อนส่งใน LEB2 ด้วย

ขั้นตอนการ zip file



1. นำ website files ทั้งหมดใส่ใน folder เดียวกัน
2. Compress file โดยใช้ ZIP compression
3. อัพ zip file ลง LEB2 ที่กำหนดให้ พร้อมทั้งใส่ URL link ของ website ของเรา
4. ส่งโดยอีเมลตามหน้าถัดไป
5. อย่าลืม references (แหล่งอ้างอิงทั้งหลาย) ถ้าใครไปเอา source codes, template หรือเนื้อหาบางอย่าง มาจากที่อื่น ควรใส่แหล่งอ้างอิง

- Portfolio web page (อยู่ใน index.html) ของนักศึกษาแต่ละคน มีรายละเอียดข้อมูลดังนี้
 - ชื่อ - นามสกุล
 - งานอดิเรก
 - ทักษะ ความสามารถ
 - ความสนใจ
 - รูป
 - มี Link ไปเว็บอื่น เช่น Facebook, Instagram ของเรา
 - หรือรายละเอียดอื่น ๆ (ข้อมูลส่วนตัว เช่น เลขบัตรประชาชน ไม่ obscene!)

- ใน project.html เนื้อหาของเพจเป็นเรื่องเกี่ยวกับ project ต่าง ๆ ที่เราเคยทำหรือคิดว่าจะทำ มีมากกว่า 1 อย่างก็ได้
 - เช่น senior project ที่เรากำลังทำ หรือ project จากที่เคยทำในห้องเรียน
 - ทำ link เชื่อมระหว่าง portfolio (index.html) กับ project (project.html) ด้วย
- HTML ทั้ง 2 files ใช้ CSS file อันเดียวกัน และ ทั้ง 2 HTML files ต้องมี CSS แบบ external CSS นะ

- ใน portfolio HTML file มี link ไปหาหน้า page ที่เราทำ assignment ในห้องเรียน ได้แก่
 - Gluten Free General Tso's Chicken
 - Blog
 - Register and login
 - Square game
 - ห้องหมดนี้ ต้องมี CSS และ JavaScript แบบ external นะครับ

- ชื่อ files และ folders ต่าง ๆ ตรวจสอบด้วยว่าเป็นตัวอักษรเล็กหรือใหญ่ เพราะเราต้องทำให้เหมือนกัน กับตอนที่เราเรียกชื่อ files หรือ folders เพื่อแสดงผลใน HTML file
- การเรียก path ไปหารูป หรือ files ต่าง ๆ ให้ใช้แบบ relative file path โดยอ้างอิงจากตำแหน่ง HTML file นั้น

เกณฑ์การให้คะแนน (15 คะแนน)



| รายละเอียด | คะแนน |
|--|-----------------|
| มีรายละเอียดงานต่าง ๆ ทั้งหมดครบถ้วน | 5 คะแนน |
| แยก CSS และ JavaScript files ได้ถูกต้อง | 5 คะแนน |
| Responsive design | 3 คะแนน |
| ความสวยงาม | 2 คะแนน |
| ไม่ใส่แหล่งอ้างอิง (ถ้าจำเป็นต้องใส่ เช่น chicken หรือใช้ template จากที่อื่น) | -1 คะแนน/ต่อจุด |
| ตั้งชื่อ File หรือชื่อ Folder ผิด link ผิด | -1 คะแนน/ต่อจุด |
| คำหยาบ ทุกประเภท + รูปที่ไม่เหมาะสม | ได้ 0 ทันที |

เราเรียนอะไรแล้วบ้าง



- HTML
- CSS
- JavaScript
 - Variable, control flow, array, event handler
 - Unobtrusive JavaScript
 - Form
- GitHub page

Setting a timer

| method | description |
|---|---|
| setTimeout(function,delayMS); | arranges to call given function after given delay in ms |
| setInterval(function,delayMS); | arranges to call function repeatedly everydelayMSms |
| clearTimeout(timerID);clearInterval(timerID); | stops the given timer |

- ทั้ง setTimeout และ setInterval ID จะ return ID สำหรับ timer
- เราสามารถใช้ ID ตัวนั้นในการ clearTimeout/Interval เพื่อหยุดเวลาได้

ตัวอย่าง

Hello! Click me!



```
<span id="output"></span>
<button id="clickme">Click me!</button>
```

```
window.onload = pageLoad;
function pageLoad() {
    document.getElementById("clickme").onclick= delayedMessage;
}

function delayedMessage() {
    document.getElementById("output").innerHTML= "Wait for it...";
    setTimeout(sayHello, 5000);
}

function sayHello() { // called when the timer goes off
    document.getElementById("output").innerHTML= "Hello!";
}
```



ตัวอย่าง

```
window.onload = pageLoad;  
var timer = null; // stores ID of interval timer  
  
function pageLoad() {  
    document.getElementById("clickme").onclick= delayMsg2;  
}  
  
function delayMsg2() {  
    if (timer === null) {  
        timer = setInterval(rudy, 1000);  
    } else {  
        clearInterval(timer);  
        timer = null;  
    }  
}  
function rudy() { // called each time the timer goes off  
    document.getElementById("output").innerHTML += " MDT!";  
}
```



Passing parameters to timers

- เราสามารถถ�ง parameter ผ่านทาง timer function ได้ โดยการเพิ่ม argument ใน timer function
- doesn't work in IE;

```
window.onload = pageLoad;

function pageLoad() {
    document.getElementById("clickme").onclick= delayedMultiply;
}

function delayedMultiply() {
    // 6 and 7 are passed to multiply when timer goes off
    setTimeout(multiply, 2000, 6, 7);
}
function multiply(a, b) {
    alert(a * b);
}
```

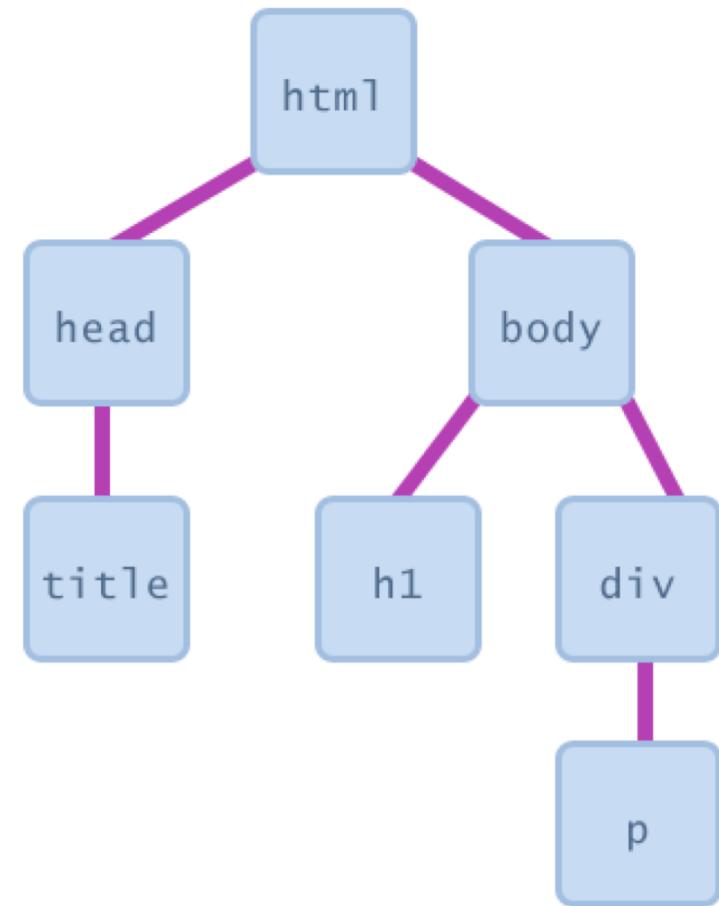
The six global DOM objects

- Every JavaScript program can refer to the following global objects

| name | description |
|-------------|--|
| document | current HTML page and its content |
| history | list of pages the user has visited |
| location | URL of the current HTML page |
| navigator | info about the web browser you are using |
| screen | info about the screen area occupied by the browser |
| window | the browser window |

The DOM tree

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



เมื่อเราต้องการสร้าง element หรือ node ใหม่

- ทำไมเราถึงไม่ควรทำแบบนี้?

```
<div id="main"></div>
```

```
window.onload = pageLoad;

function pageLoad() {
    addParagraph();
}

function addParagraph() {
    document.getElementById("main").innerHTML+=
        "<p>A paragraph!</p>";
}
```

- สมมติว่า element ที่เราจะเพิ่มเข้าไปมันดูวุ่นวายมาก ดังตัวอย่าง
- มีโอกาสเกิดความผิดพลาดจากการเขียน code ได้สูงมาก

```
function addParagraph() {  
    document.getElementById("main").innerHTML += "<p style='color: red;  
margin-left: 50px';> A paragraph!</p>";  
}
```

A paragraph!

การสร้าง nodes หรือ elements ใหม่

- เราจะทำการสร้าง node ใหม่ ก่อนที่เราจะเพิ่ม node นั้นลงใน document
- หลังจากสร้างแล้ว เราต้อง add ลงไปใน document ด้วย

| name | description |
|---------------------------------|---|
| document.createElement("tag") | creates and returns a new empty DOM node representing an element of that type |
| document.createTextNode("text") | creates and returns a text node containing given text |

การเพิ่ม/ลด nodes ใน document

- เราสามารถเพิ่ม/ลดโดยใช้ method ต่อไปนี้

| name | description |
|-----------------------|--|
| appendChild(node) | places given node at end of this node's child list |
| insertBefore(new,old) | places the given new node in this node's child list just before old child |
| removeChild(node) | removes given node from this node's child list |
| replaceChild(new,old) | replaces given child with new node |

```
function addParagraph(){
    var p = document.createElement("p");
    p.innerHTML= "A paragraph!";
    document.getElementById("main").appendChild(p);
```

การกำหนด Style โดยใช้ DOM object

- The Style object represents an individual style statement.

```
function addParagraph(){  
    var p = document.createElement("p");  
    p.innerHTML= "A paragraph!";  
    document.getElementById("main").appendChild(p);  
    document.getElementById("main").style.color = "red";  
}
```

A paragraph!

- Set margin-left

```
function addParagraph(){  
    var p = document.createElement("p");  
    p.innerHTML= "A paragraph!";  
    document.getElementById("main").appendChild(p);  
    document.getElementById("main").style.color = "red";  
    document.getElementById("main").style.marginLeft = "50px";  
}
```

A paragraph!

Selecting groups of DOM objects

methods in document and other DOM objects (* = HTML5):

| name | description |
|----------------------|--|
| getElementsByTagName | returns array of descendants with the given tag, such as "div" |
| getElementsByName | returns array of descendants with the given name attribute (mostly useful for accessing form controls) |
| querySelector* | returns the first element that would be matched by the given CSS selector string |
| querySelectorAll* | returns an array of all elements that would be matched by the given CSS selector string |

ตัวอย่าง

```
<p>This is my add</p>
<div id="address">
    <p>1234 Street</p>
    <p>Atlanta, GA</p>
</div>
```

```
window.onload = pageLoad;
function pageLoad(){
    var addrParas = document.querySelectorAll("p");
    //var addrParas = document.querySelectorAll("#address p");
    for (var i= 0; i< addrParas.length; i++) {
        addrParas[i].style.backgroundColor= "yellow";
    }
}
```

This is my add
1234 Street
Atlanta, GA
Hello World,

Getting/setting CSS classes

- JS DOM's className property corresponds to HTML class attribute

```
.highlight{  
    color:pink;  
    font-size: 30px;  
    background-color: green;  
}
```

```
<button id="clickme">Click me!</button>  
<div id="main"> </div>
```

```
function addParagraph(){
    var p = document.createElement("p");
    p.innerHTML= "A paragraph!";
    document.getElementById("main").appendChild(p);
    document.getElementById("main").style.color = "red";
    document.getElementById("main").style.marginLeft = "50px";
}

window.onload = pageLoad;

function pageLoad() {
    addParagraph();
    var x = document.getElementById("clickme");
    x.onclick = highlightField;
}

function highlightField(){
    var text = document.getElementById("main");
    if (!text.className){
        text.className = "highlight";
    }else if (!text.classList.contains("invalid")){
        text.className = "";
    }
}
```

Getting/setting CSS classes with classList

- classList collection has methods **add**, **remove**, **contains**, **toggle** to manipulate CSS classes
- เมื่อนอกับ className DOM property, แต่สามารถใส่ class ได้หลายตัวโดยที่ไม่ต้องกำหนดให้ แต่ละ class ทำการเว้นวรรค

```
.highlight2{  
    text-align: center;  
    width: 50%;  
}
```

```
function highlightField(){
    var text = document.getElementById("main");
    if (!text.className){
        text.className = "highlight";
    }else if (!(text.classList.contains("invalid"))){
        text.classList.add("highlight2");
    }
}
```

Traversing the DOM tree manually



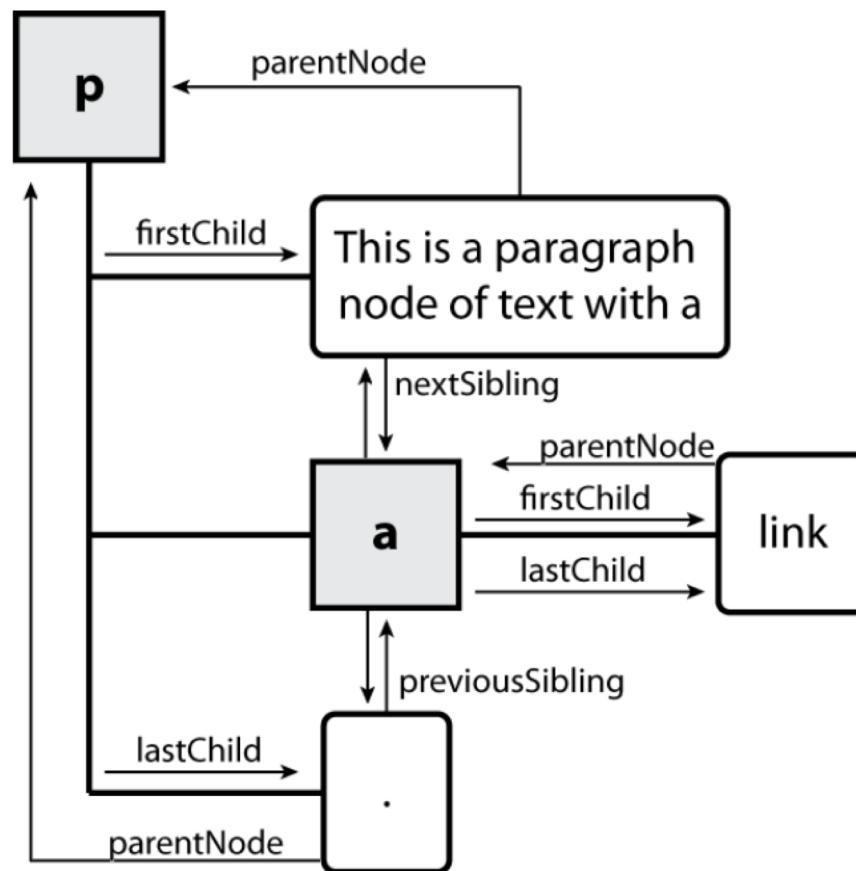
- ຖືກ node ຂອງ DOM object ຈະມີ properties ຕ່ອໄປນີ້

| name(s) | description |
|------------------------------|---|
| firstChild, lastChild | start/end of this node's list of children |
| childNodes | array of all this node's children |
| nextSibling, previousSibling | neighboring nodes with the same parent |
| parentNode | the element that contains this node |

ตัวอย่าง DOM tree properties

< p id="foo">This is a paragraph of text with a

< a href="/path/to/another/page.html">link</p>



Removing a node from the page

- `obj.parentNode.removeChild(obj);`

```
<button id="clickme2">Click me!</button>
<div id="main2">
  <h2 id="removeMe"> Hello World, </h2>
</div>
```

```
window.onload = pageLoad;

function pageLoad() {
  var x = document.getElementById("clickme2");
  x.onclick = removeMe;
}

function removeMe(){
  var x = document.getElementById("removeMe");
  x.parentNode.removeChild(x);
}
```

JavaScript events (object)

- The **event** object

```
function name(event) {  
    // an event handler function ...  
}
```

| property name | description |
|------------------|---|
| type | what kind of event, such as "click" or "mousedown" |
| target | the element on which the event occurred |
| timeStamp | when the event occurred |

Mouse events

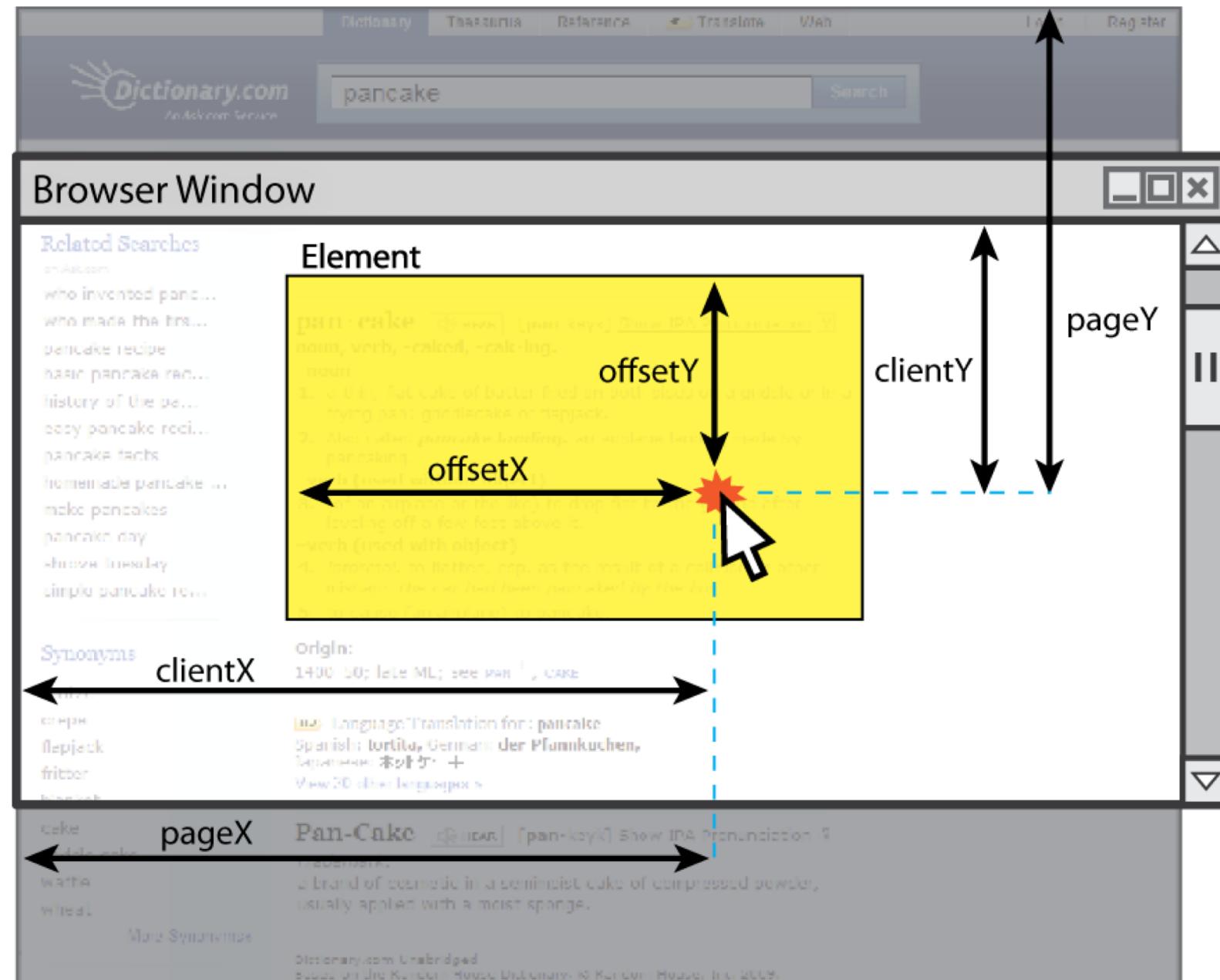
| | |
|------------------|---|
| click | user presses/releases mouse button on the element |
| dblclick | user presses/releases mouse button twice on the element |
| mousedown | user presses down mouse button on the element |
| mouseup | user releases mouse button on the element |

| | |
|------------------|--|
| mouseover | mouse cursor enters the element's box |
| mouseout | mouse cursor exits the element's box |
| mousemove | mouse cursor moves around within the element's box |

Mouse event objects

- The event passed to a mouse handler has these properties:

| property/method | description |
|--|---|
| <code>clientX</code> <code>clientY</code> | coordinates in <i>browser window</i> |
| <code>screenX</code> <code>screenY</code> | coordinates in <i>screen</i> |
| <code>offsetX</code> <code>offsetY</code> | coordinates in <i>element</i> (non-standard) |
| <code>button</code> | integer representing which button was pressed (0=Left, 1=Middle, 2=Right) |



ตัวอย่าง

```
<div id="target">Move the mouse over me!</div>
```

```
window.onload= pageLoad;

function pageLoad() {
    var target = document.getElementById("target");
    target.onmousemove= target.onmousedown= showCoords;
};

function showCoords(event) {
    document.getElementById("target").innerHTML =
    "screen : (" + event.screenX+ ", " + event.screenY+ ")\n"+
    "client : (" + event.clientX+ ", " + event.clientY+ ")\n" +
    "button : " + event.button;
}
```



สอบ Quiz วันที่ 13 มี.ค. 2563 (30 นาที)

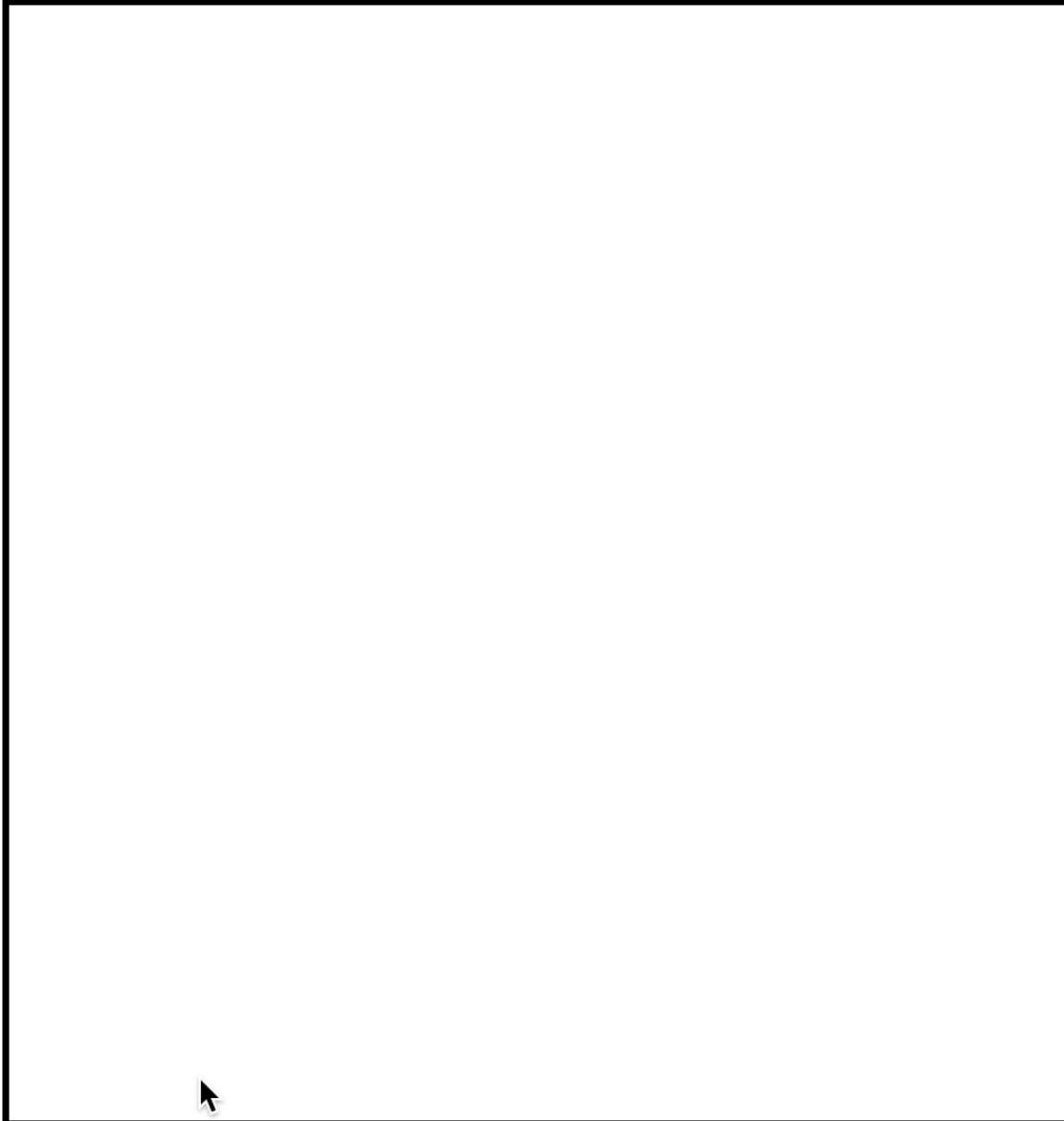
- 10 คะแนน (ใช้คอมพิวเตอร์ได้)
- ห้ามใช้อินเทอร์เน็ต
- ห้ามคุยกับเพื่อน
- เปิดหนังสือ sheet หรือ slide ได้
- เริ่มสอบ 9:15 กับ 13:45

Assignment 7:

- Download assignment7.zip file
- ทำการสร้าง square game โดยในเกมนี้ เราจะต้องกำหนด ก่อนว่าเราจะให้มีสีเหลี่ยมทั้งหมดกี่อัน และสีของสีเหลี่ยม เมื่อเรากด Start แล้ว เราจะต้องกดลบสีเหลี่ยมทั้งหมด ภายในเวลา 30 วินาที ถึงจะชนะในเกมนี้
- ส่งท้ายคำ
- **การบ้าน** ทำการตั้งแต่ง square game ของเรา ให้สวยงาม

Control

Number of box: Red Timer:



References

- <https://www.w3schools.com/>
- <https://courses.cs.washington.edu/courses/cse154/17sp/>
- *Web Programming Step by Step* by Marty Stepp, Jessica Miller, and Victoria Kirst,
<http://www.webstepbook.com/>