

# Responsive Web Design: CSS Responsive

ผศ. ดร. เกื้อแก้ว ธีเนศวร

[kejkaew.tha@kmutt.ac.th](mailto:kejkaew.tha@kmutt.ac.th)

# การบ้าน

---

- สร้าง Portfolio web page ของนักศึกษาแต่ละคนสำหรับใช้เพื่อการสมัครงาน โดยใช้ HTML ง่าย ๆ และมีรายละเอียดข้อมูลดังนี้
- ทำการตกแต่ง Portfolio webpage ของนักศึกษา โดยใช้ CSS แบบ External CSS นะ
- ทำการตกแต่ง layout ของนักศึกษา เพิ่มเติม

# การบ้าน

---

- ทำ html पेจอีกหนึ่งเพจมีชื่อว่า project.html
- เนื้อหาของเพจเป็นเรื่องเกี่ยวกับ project ต่าง ๆ ที่เราเคยทำหรือคิดว่าจะทำ มีมากกว่า 1 อย่างก็ได้
  - เช่น project จากที่เคยทำในห้องเรียน
- ใช้ CSS file ร่วมกับ portfolio html file ของเรา

# ตารางเรียน

สัปดาห์	วันที่	หัวข้อ/รายละเอียด	การบ้าน (ผู้ตรวจ)	ส่ง Assignment
1	5/8/2568	Introduction to Web programming, Basic HTML 1	Assignment 1 (KT,KN)	1
	12/8/2568	หยุด		
2	19/8/2568	Basic HTML 2 + CSS 1	Assignment 2 (KT, KN)	1,2
3	26/8/2568	CSS 2	Assignment 3 (KT)	1 2 3
4	2/9/2568	Responsive Web design	Assignment 4 (KT, KN)	2,3,4
5	9/9/2568	JavaScript 1 + DOM	Assignment 5 (KT, KN)	3,4,5
6	16/9/2568	JavaScript 2 + GitHub	Assignment 6 (KT)	4,5,6
7	23/9/2568	JavaScript 3	Assignment 7 (KT, KN)	5,6,7
8	30/9/2568	AJAX + JSON	Assignment 8 (KT, KN)	6,7,8
9	7/10/2568	Node.js 1	Assignment 9 (KT)	7,8,9
10	14/10/2568	Node.js 2	Assignment 10 (KT, KN)	8,9,10
	21/10/2568	หยุด		
11	28/10/2568	Node.js 3	Assignment 11 (KT, KN)	9,10,11
12	4/11/2568	Node.js 4 + MySQL	Assignment 12 (KT, KN)	10,11,12
13	11/11/2568			11,12
14	18/11/2568			12
16	25/11/2568	ส่งโปรเจค (9:00-16:00)	KT	
17	/12/2568	สอบปลายภาค ในตาราง		

# วันนี้เราจะเรียนเรื่อง...

- CSS Animations
- CSS Website layout
- Responsive Web Design
- Assignment ในห้องเรียน
- Quiz
  - เปิดเนื้อหา สไลด์ สรุป code ต่างๆ ได้หมด
  - ห้ามใช้ AI ทุกประเภท ห้ามถามเพื่อน ห้ามแชท ห้ามคุย
  - สอบในรอบที่เรียน
    - 11.00 - 11:30 น..
    - 15:30 – 16.00 น.
    - อย่าลืม ดินสอ ปากกา

# CSS Animations

- CSS animations สามารถทำให้เราทำ animation ได้โดยไม่ต้องใช้ JavaScript
- @keyframes CSS property

Property	Description
<u>@keyframes</u>	Specifies the animation code
<u>animation</u>	A shorthand property for setting all the animation properties
<u>animation-delay</u>	Specifies a delay for the start of an animation
<u>animation-direction</u>	Specifies whether an animation should be played forwards, backwards or in alternate cycles

# CSS Animation Properties

Property	Description
<u>animation-duration</u>	Specifies how long time an animation should take to complete one cycle
<u>animation-fill-mode</u>	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
<u>animation-iteration-count</u>	Specifies the number of times an animation should be played
<u>animation-name</u>	Specifies the name of the @keyframes animation
<u>animation-play-state</u>	Specifies whether the animation is running or paused
<u>animation-timing-function</u>	Specifies the speed curve of the animation

## ตัวอย่างเช่น

```
<div class="rotate-ball"></div>
```

```
.rotate-ball{  
  width: 30%;  
  animation: moveball 2s linear  
    infinite;  
  position: relative;  
}  
  
@keyframes moveball{  
  0% {left: 0px; top:0;}  
  100%{left: 900px; top:0px;}  
}
```

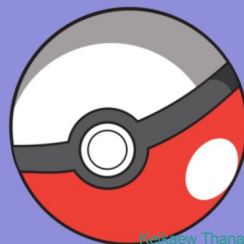




## ตัวอย่าง การหมุน

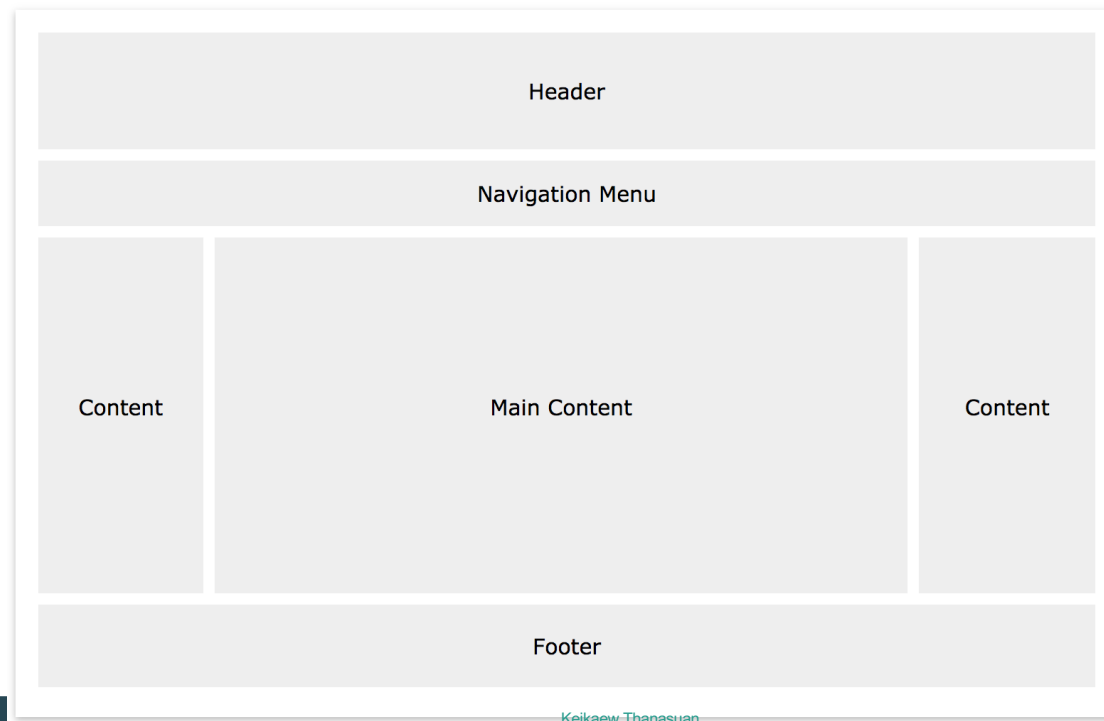
```
<div class="rotate-ball"></div>
```

```
.rotate-ball{  
  width: 30%;  
  animation: spin 2s linear  
    infinite;  
  margin: auto;  
}  
  
@keyframes spin {  
  0% { transform: rotate(0deg); }  
  100% { transform: rotate(360deg); }  
}
```



# Website Layout

- โดยปกติแล้ว website จะแบ่งออกเป็น headers, menus, content และ footer:



# Header

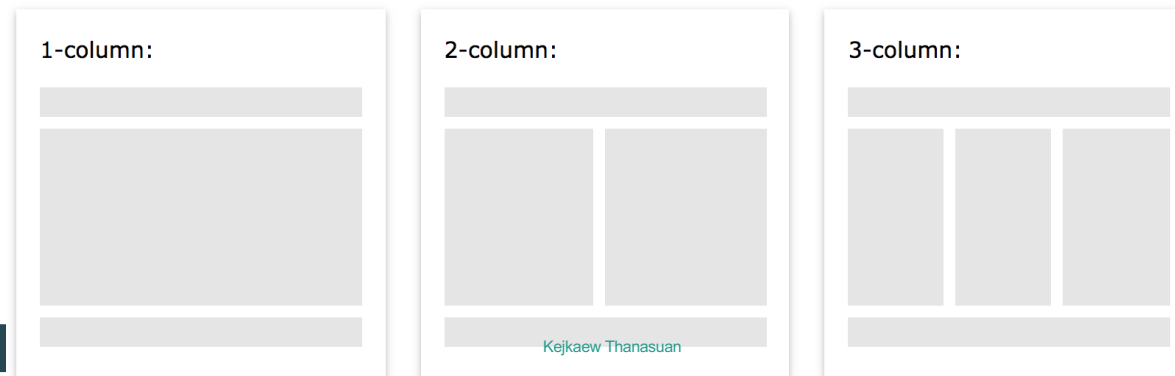
- A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:



# Content

---

- The layout in this section, often depends on the target users. The most common layout is one (or combining them) of the following:
- **1-column** (often used for mobile browsers)
- **2-column** (often used for tablets and laptops)
- **3-column** layout (only used for desktops)



# Footer

- The footer is placed at the bottom of your page. It often contains information like copyright and contact info:

REPORT ERROR	PRINT PAGE	FORUM	ABOUT
<b>Top 10 Tutorials</b>	<b>Top 10 References</b>	<b>Top 10 Examples</b>	<b>Web Certificates</b>
HTML Tutorial	HTML Reference	HTML Examples	HTML Certificate
CSS Tutorial	CSS Reference	CSS Examples	CSS Certificate
JavaScript Tutorial	JavaScript Reference	JavaScript Examples	JavaScript Certificate
W3.CSS Tutorial	W3.CSS Reference	W3.CSS Examples	jQuery Certificate
Bootstrap Tutorial	Bootstrap Reference	Bootstrap Examples	PHP Certificate
SQL Tutorial	Browser Statistics	HTML DOM Examples	Bootstrap Certificate
PHP Tutorial	PHP Reference	PHP Examples	XML Certificate
jQuery Tutorial	HTML Colors	jQuery Examples	
Angular Tutorial	jQuery Reference	Angular Examples	
How To Tutorial	AngularJS Reference	XML Examples	

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2018 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

# Responsive Web Design

- ทำไมเราถึงต้องมีการทำ Responsive Web Design?
- Responsive web design จะทำให้ website ของเรา ดูดีในทุกอุปกรณ์แสดงผล (โทรศัพท์ คอมพิวเตอร์ หรือ Tablet)
- Responsive web design จะใช้แค่ HTML and CSS ไม่มี JavaScript



Desktop



Tablet



Phone

# Setting The Viewport

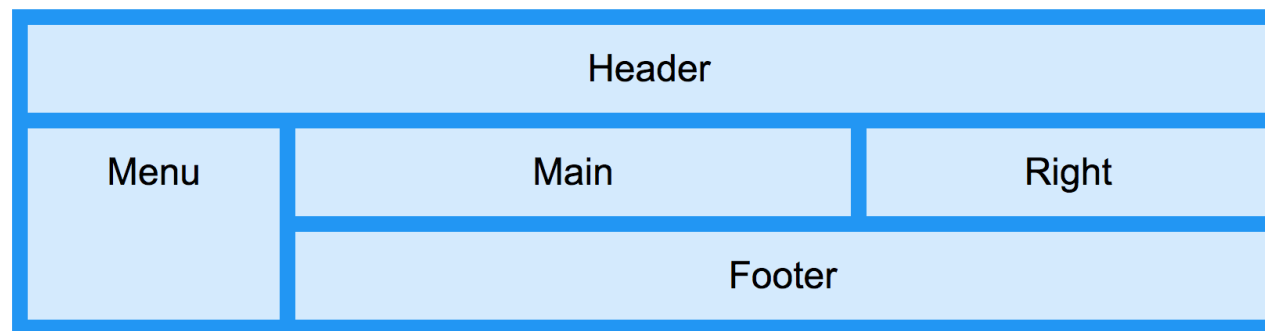
---

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- ใน header ของ HTML file

# CSS Grid Layout Module

- The CSS Grid Layout Module คือ การสร้าง rows และ columns บน web page ทำให้ง่ายในการ design web page โดยเราไม่จำเป็นต้องใช้ floats และ position เข้ามาช่วย
- สามารถทำได้โดยตั้งค่า display เป็น grid หรือ inline grid ใน container ใหญ่ และ element ที่อยู่ใน container นั้นจะกลายเป็น grid items





# CSS Grid Container

---

- The **grid-template-columns** property defines the number of columns in your grid layout, and it can define the width of each column.
- ตัวอย่างเช่น

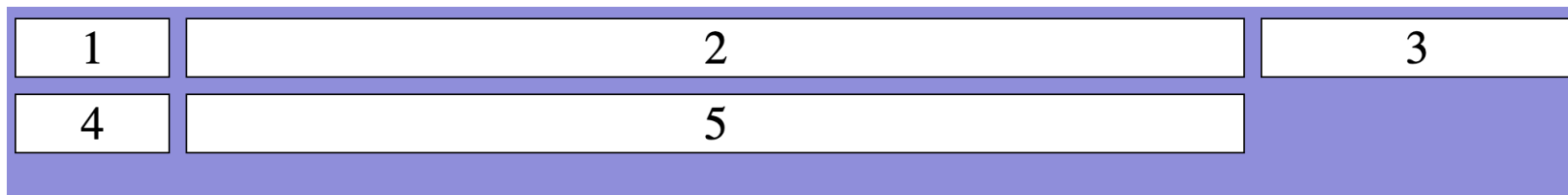
```
.grid-container {  
    display: grid;  
    grid-template-columns: auto auto auto auto;  
}
```

## ตัวอย่างเช่น

```
<div class="grid-container">
  <div id="header">1</div>
  <div id="menu">2</div>
  <div id="main">3</div>
  <div id="right">4</div>
  <div id="footer">5</div>
</div>
```

```
/* grid display */
.grid-container{
  display: grid;
  grid-template-columns: 10% auto 20%;
  grid-gap: 10px;
}

.grid-container > div{
  border: 1px solid black;
  font-size: 30px;
  text-align: center;
  background-color: white;
}
```



## The `grid-template-rows` Property

---

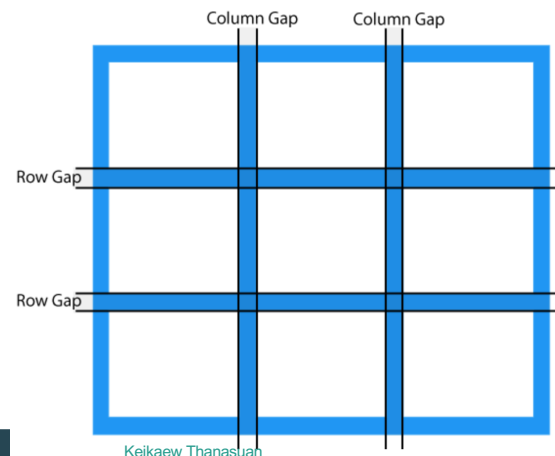
- The `grid-template-rows` property defines the height of each row (กำหนดความสูงของแต่ละแถว).

- ตัวอย่างเช่น

```
.grid-container {  
    display: grid;  
    grid-template-rows: 80px 200px;  
}
```

# Grid Gaps

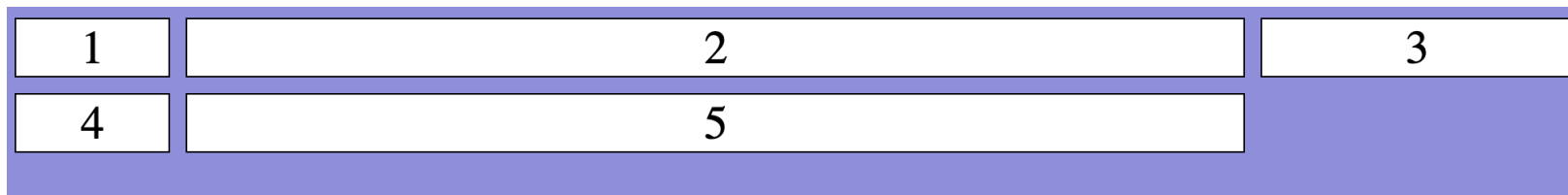
- The space between each column/row are called gaps.
- **grid-column-gap** : sets the gap between the columns
- **grid-row-gap** : sets the gap between the rows
- **grid-gap** : set the grid-column-gap and the grid-row-gap



# ตัวอย่าง

```
/* grid display */
.grid-container{
  display: grid;
  grid-template-columns: 10% auto 20%;
  grid-gap: 10px;
}

.grid-container > div{
  border: 1px solid black;
  font-size: 30px;
  text-align: center;
  background-color: white;
}
```



## CSS Grid Item (div ที่อยู่ใน Grid container)

---

- The grid-column Property: กำหนดว่า item ของเราจะเริ่มจาก column ที่เท่าไร ไปถึง column ที่เท่าไร
- The grid-row Property: กำหนดว่า item ของเราจะเริ่มจาก row ที่เท่าไร ไปถึง row ที่เท่าไร
- The grid-area Property : คือการนำ grid-column Property และ grid-row Property มารวมไว้ด้วยกัน

# ตัวอย่าง

เริ่มที่

ถึง **column**  
ที่ 2

```
#item1{  
  grid-column: 1 / 3;  
}
```

เริ่มที่

กว้าง

```
#item1{  
  grid-row: 1 / span 3;  
}
```

```
#item1{  
  grid-column: 1 / span 3;  
}
```

row

row

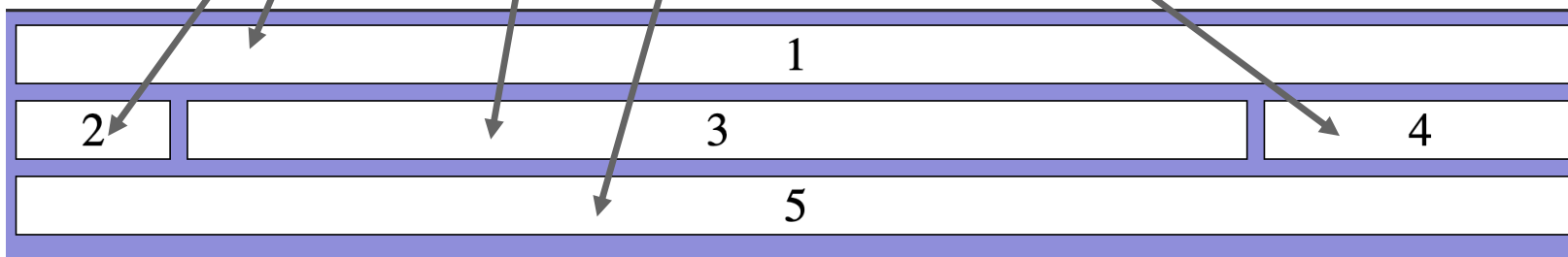
```
#item1{  
  grid-area: 1 / 1 / 3 / 3;  
}
```

column

column

# ตัวอย่าง

```
#header{  
  grid-column: 1 / span 3;  
}  
  
#menu{  
  grid-column: 1 / span 1;  
}  
  
#main{  
  grid-column: 2 / span 1;  
}  
#right{  
  grid-column: 3 / span 1;  
}  
  
#footer{  
  grid-column: 1 / span 3;  
}
```





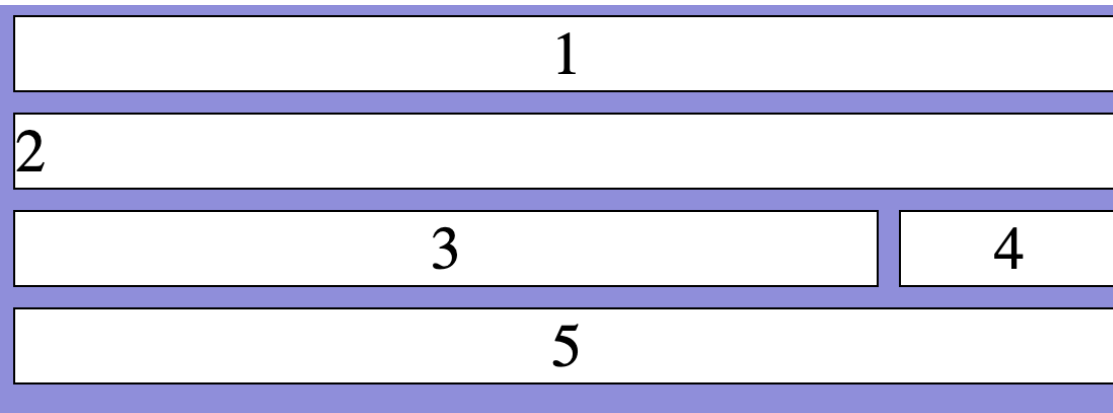
# CSS: Media Query

- Media query is a CSS technique introduced in CSS3.
- โดย CSS properties ที่อยู่ใน media จะทำงานก็ต่อเมื่อ เงื่อนไขใน @media rule เป็นจริง เช่น

rule

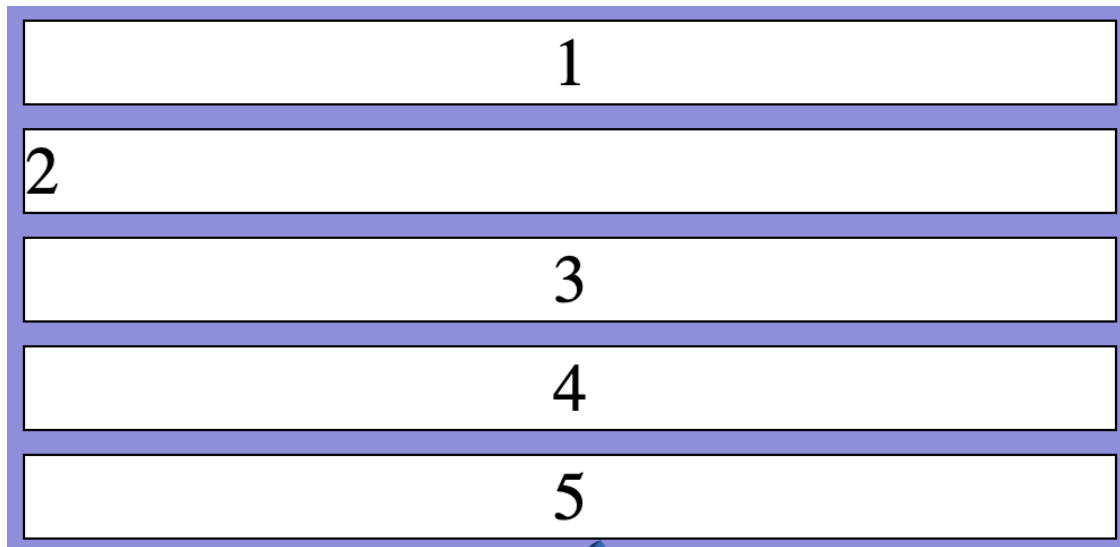
- @media only screen and (max-width: 500px) {  
    body {  
        background-color: lightblue;  
    }

## ตัวอย่าง 2 columns



```
@media only screen and (max-width: 767px){  
  .grid-container{  
    display: grid;  
    grid-template-columns: auto 20%;  
    grid-gap: 10px;  
  }  
  #header{  
    grid-column: 1 / span 2;  
  }  
  #menu{  
    grid-column: 1 / span 2;  
    text-align: left;  
  }  
  #main{  
    grid-column: 1 / span 1;  
  }  
  #right{  
    grid-column: 2 / span 1;  
  }  
  #footer{  
    grid-column: 1 / span 2;  
  }  
}
```

## ตัวอย่างสำหรับ mobile (1 column)



```
@media only screen and (max-width: 500px){  
  .grid-container{  
    display: grid;  
    grid-template-columns: auto;  
    grid-gap: 10px;  
  }  
  #header{  
    grid-column: 1 / span 1;  
  }  
  
  #menu{  
    grid-column: 1 / span 1;  
  }  
  
  #main{  
    grid-column: 1 / span 1;  
  }  
  #right{  
    grid-column: 1 / span 1;  
  }  
  
  #footer{  
    grid-column: 1 / span 1;  
  }  
}
```

# Pokémon web design

1		
2	3	4
5		

1



2

Home  
Calculators



3

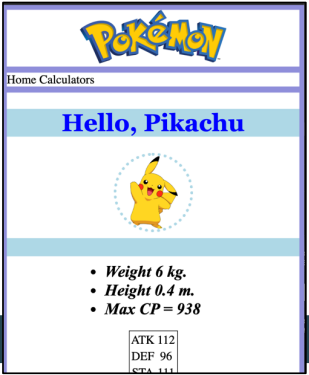
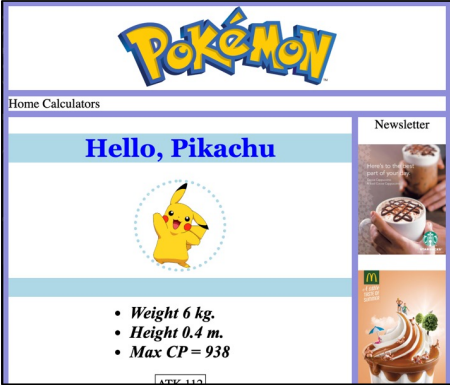
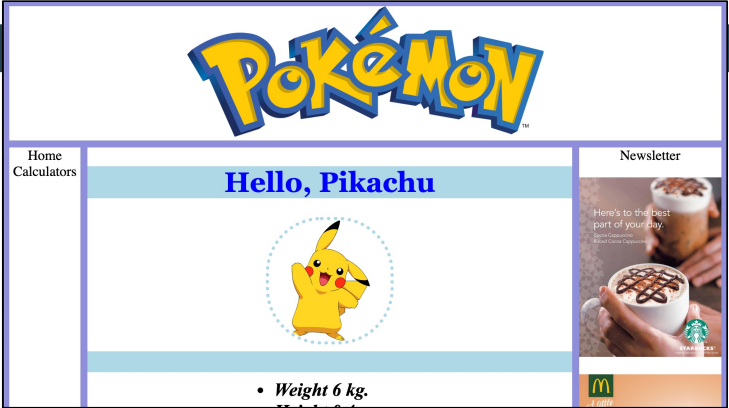


4

5

If you want to know more, please click!: [Reference](#)

Images from: [Reference](#)



# CSS Specificity

ถ้า element เดียวกัน แต่มีการระบุ CSS rule มากกว่า 1 แบบ จะต้องใช้ CSS ตัวใด

- **Inline styles** - Example: `<h1 style="color: #ffffff;">`.
- **IDs** - as `#navbar`.
- **Classes, attributes and pseudo-classes** - This category includes `.classes`, `[attributes]` and pseudo-classes such as `:hover`, `:focus` etc.
- **Elements and pseudo-elements** - This category includes element names and pseudo-elements, such as `h1`, `div`, `:before` and `:after`.

# How to Calculate Specificity?

---

เริ่มคำนวณจาก 0

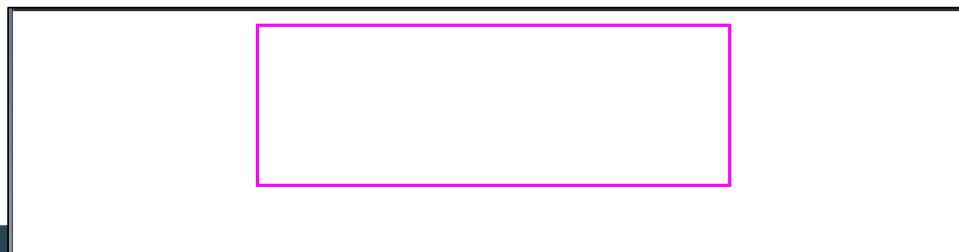
- + 1000 for style attribute
- + 100 for each ID
- + 10 for each attribute, class or pseudo-class
- + 1 for each element name or pseudo-element.
- สุดท้าย CSS rule แบบไหน มีค่ามากกว่า เลือกอันนั้น



## ตัวอย่าง

```
<div>  
  <p class="class1"></p>  
</div>  
<hr>
```

ผลที่ได้



```
div .class1{  
  border: 2px solid magenta;  
  margin: auto;  
  width: 50%;  
  height: 100px;  
}  
p{  
  border: 2px solid rgb(6, 58, 23);  
  margin: auto;  
  width: 30%;  
  height: 100px;  
}  
@media only screen and (max-width: 767px){  
  .class1{  
    border: 2px solid blue;  
  }  
}
```

# สอบ Quiz 20 นาที

## Week 4: Classroom game

---

- ทำ classroom game ใน LEB2 ก่อนสัปดาห์หน้า

## Assignment 4

- จาก Pizza dip ให้ตกแต่งเว็บเพจ โดยใช้
  - Grid display
  - พยายามทำ webpage ให้เป็น Responsive โดยใช้ Media Queries
  - พยายามกำหนด width เป็นแบบ percent
  - เปลี่ยนสี, padding, size etc.
- ลอง design layout สำหรับ grid display ก่อน (ลองทำใส่กระดาษ)



Desktop

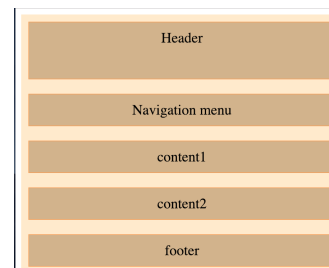
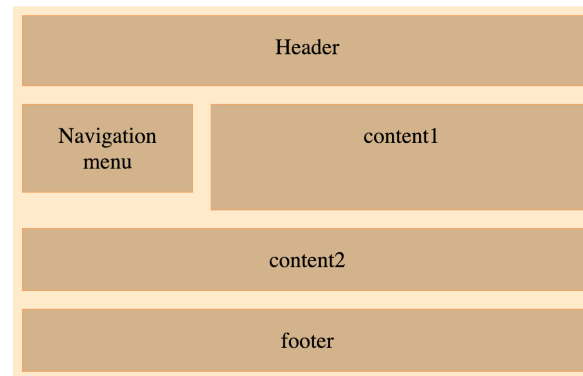
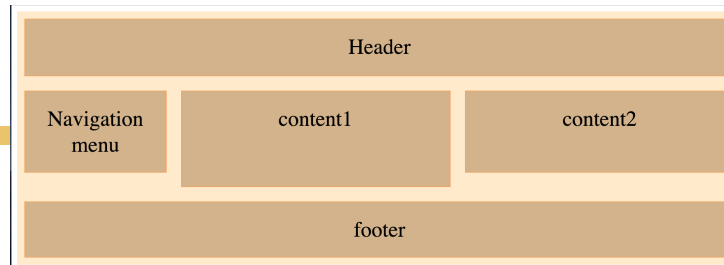


Tablet



Phone

# Assignment



# References

---

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