

# HTML5 with CSS3

16 Aug 2025

ผศ.วิญญู นรินทล้าพงศ์ (winyu.nir@dpu.ac.th)

2

## HTML5 with CSS3 (ตอนที่ 1)

# หัวข้อการบรรยายเกี่ยวกับ CSS ตอนที่ 1

3

- รู้จักกับ CSS3
- ตำแหน่งของ CSS
- เทคนิคการประยุกต์ใช้งานสไต์ชีท
  - ▣ การจัดรูปแบบข้อความและรูปภาพ
  - ▣ การเพิ่มสีสันให้กับ link และพื้นหลัง
  - ▣ การกำหนดรูปแบบของ Scroll Bar

# รู้จักกับ CSS

4

- CSS (Cascading Style Sheet) เป็นภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS จะกำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ style) ของเนื้อหาในเอกสาร
- ใช้กำหนดรูปแบบการจัดหน้าเว็บเพจและจัดวางตำแหน่งของรูปหรือออบเจ็คได้ตามความต้องการ
- หลักสำคัญคือ การแยกการจัดรูปแบบและการตกแต่งเว็บเพจ ออกจาก เนื้อหา (content) ด้วยการกำหนดคุณสมบัติ (property) ให้กับแท็กต่างๆ
- W3C ได้กำหนด CSS3 เป็นเวอร์ชันล่าสุด

# รู้จักกับ CSS

5

- CSS จะมีไวยากรณ์ (syntax) คือ

selector {property:value}

โดยที่ selector คือ แท็กของ HTML ที่ต้องการกำหนดรูปแบบ

property คือ แอททริบิวต์ของแท็กที่ต้องการกำหนด

value คือ ค่าที่ต้องการกำหนดให้แต่ละ property

- รูปแบบการใช้งาน CSS3 มี 3 รูปแบบ

- Inline

- Embedded

- External แยกเป็น Linking กับ Import

# รู้จักกับ CSS

6

## □ ตัวอย่าง CSS เช่น

- ต้องการกำหนด ให้แท็ก <body> มีสี Color เป็นสีดำ (black)

```
body {color:black}
```

- ต้องการกำหนดให้แท็ก <p> มีรูปแบบตัวอักษรเป็น sans serif

```
p {font-family:"sans serif"}
```

- ต้องการกำหนดให้แท็ก <p> มีรูปแบบตัวอักษรเป็น arial และมีสีดำ

```
p {
```

```
    color:black;
```

```
    font-family:"sans serif"
```

```
}
```

# ตำแหน่งของ CSS

7

## □ แบบที่ 1 กำหนดสไตล์ชีทแบบ Embedding

เป็นการกำหนดสไตล์ชีทฝังไว้ในไฟล์โค้ด HTML โดยสามารถกำหนดไว้ในส่วนต่างๆ ดังนี้

### ▣ ส่วนหัว (head)

```
<head>
```

```
<style>
```

```
font { color: blue; font-size: 16px; }
```

```
</style>
```

```
</head>
```

# ตำแหน่งของ CSS

8

□ แบบที่ 1 กำหนดสไตล์ชีทแบบ Embedding

▣ ส่วนเอกสาร (body)

```
<body>
```

```
<style>
```

```
font { color: blue; font-size: 16px; }
```

```
</style>
```

```
</body>
```



# ตำแหน่งของ CSS

9

## □ แบบที่ 1 กำหนดสไตล์ชีทแบบ Embedding

### ▣ ส่วนสคริปต์

```
<head>  
  <script language="JavaScript">  
    document.write("  
      <font style=color: blue; font-size: 16px;>");  
  </script>  
</head>
```

### ▣ กำหนดไว้ในแท็กแบบ inline

```
<body style="color: blue; font-size: 16px;">  
  .....  
</body>
```

# ตำแหน่งของ CSS

10

## □ แบบที่ 2 กำหนดสไตล์ชีตแบบ Linking

เป็นการกำหนดสไตล์ชีตไว้ในอีกไฟล์หนึ่ง ซึ่งเป็นไฟล์กลางมีส่วนขยาย หรือนามสกุลเป็น .css โดยที่ในไฟล์ HTML จะมีการใช้แท็ก link เพื่อเชื่อมโยงหรืออ้างอิงถึงไฟล์ที่เก็บสไตล์ชีตอีกทีหนึ่ง

### ▣ ตัวอย่าง

```
<head>  
  <link href="style001.css" rel="stylesheet"  
  type="text/css">  
</head>
```

# ตำแหน่งของ CSS

11

## □ แบบที่ 3 กำหนดสไตล์ชีทแบบ Import

เป็นการกำหนดสไตล์ชีทไว้ในอีกไฟล์หนึ่ง แยกต่างหากจากไฟล์ HTML โดยไฟล์ HTML จะดึงข้อกำหนดเหล่านั้นมาใช้ด้วยแท็กคำสั่ง `@import` ซึ่งในไฟล์ HTML ต้องสร้างแท็ก `<style>...</style>` ไว้รองรับสามารถกำหนดสไตล์เฉพาะที่ใช้ในเพจนี้เพิ่มได้ตามต้องการ

### ▣ ตัวอย่าง

```
<head>
  <style>
    @import URL(style1.css);
    div { font-weight: bolder; };
  </style>
</head>
```

# หน่วยวัดที่ใช้ในการจัดทำหน้าเว็บเพจ

12

- หน่วยวัดที่ใช้ในการจัดทำหน้าเว็บเพจ
  - ▣ การอ้างอิงหน่วยวัดแบบ Relative Length – อ้างอิงแบบสัมพันธ์กัน หรืออ้างอิงแบบอัตราส่วน มีวิธีการอ้างอิงที่นิยมกันดังนี้
    - % - (percent) เป็นการอ้างอิงกับหน้าจอภาพให้สัมพันธ์กัน โดย content ทั้งหมดจะมีการคิดเพิ่มหรือขยายตามอัตราส่วนของหน้าจอภาพ เช่น 30% 50% 70%
    - px – (pixel) เป็นการอ้างอิงกับความละเอียดของหน้าจอภาพ โดยระบุให้สัมพันธ์กับสัดส่วนของจอภาพ เช่น 5px 10px 30px
    - em – (emphasize) คือ ความสูงของตัวอักษร เป็นการระบุการอ้างอิงเป็นจำนวนเท่าของ font ที่กำหนดไว้ให้กับแท็ก body ex (ex ย่อมาจาก x-height) เช่น 1.5ex 2ex

# หน่วยวัดที่ใช้ในการจัดทำหน้าเว็บเพจ

13

## □ หน่วยวัดที่ใช้ในการจัดทำหน้าเว็บเพจ

▣ การอ้างอิงหน่วยวัดแบบ Absolute Length – เป็นการอ้างอิงหน่วยวัดแบบสัมพัทธ์ หรือเป็นกำหนดแบบตายตัวกับหน่วยวัดในความเป็นจริง ได้แก่

■ in (inches) – หน่วยเป็นนิ้ว(ฟุต)

■ cm (centimeters) – หน่วยเป็นเซนติเมตร(เมตรริก)

■ mm (millimeters) – หน่วยเป็นมิลลิเมตร(เมตรริก)

(1in = 2.54cm, 1cm = 10mm)

■ pt (points) – หน่วยของฟอนต์ในสื่อสิ่งพิมพ์ โดย 1pt = 1/72 inches และ 10pt = 12px

■ pc (picas) – หน่วยของฟอนต์ในสื่อสิ่งพิมพ์เช่นกัน โดย 1pc = 12pt

และ 6pc = 72pt

# การจัดรูปแบบข้อความและรูปภาพ

14

## □ คุณสมบัติที่สำคัญในการแสดงผลข้อความ

Property	คำอธิบาย	ค่าที่สามารถกำหนดได้
text-align	ใช้กำหนดตำแหน่งของข้อความ	left, right, center, justify
text-decoration	ใช้สำหรับตกแต่งข้อความ	None, underline, overline, line-through และ blink
text-indent	ใช้กำหนดย่อหน้าให้กับข้อความ	เลขจำนวนเต็มพร้อมหน่วยที่ใช้
text-transform	ใช้เปลี่ยนแปลงรูปแบบของข้อความให้เป็นแบบตัวพิมพ์เล็ก-ใหญ่	none, capitalize, uppercase และ lowercase
word-spacing	ใช้กำหนดระยะห่างระหว่างข้อความ	เลขจำนวนเต็มพร้อมหน่วยที่ใช้
letter-spacing	ใช้กำหนดระยะห่างระหว่างตัวอักษร	เลขจำนวนเต็มพร้อมหน่วยที่ใช้
line-height	ใช้กำหนดระยะห่างของบรรทัด	เลขจำนวนเต็มพร้อมหน่วยที่ใช้

# การจัดรูปแบบข้อความและรูปภาพ

15

## □ คุณสมบัติที่ใช้กำหนดระยะห่างระหว่าง Element

Property	คำอธิบาย	ค่าที่สามารถกำหนดได้
margin	กำหนดระยะห่างระหว่าง <b>tag</b> นั้น ๆ กับ <b>tag</b> ที่อยู่รอบ ๆ ทั้ง 4 ด้าน	<b>auto</b> , และเลขจำนวนเต็มพร้อมหน่วยที่ใช้
margin-top margin-bottom margin-left margin-right	กำหนดระยะห่างระหว่าง <b>tag</b> นั้น ๆ กับ <b>tag</b> ที่อยู่ด้านบน ล่าง ซ้าย ขวา ตามลำดับ	<b>auto</b> , และเลขจำนวนเต็มพร้อมหน่วยที่ใช้
padding	กำหนดระยะห่างระหว่างข้อความกับ <b>tag</b> ทั้ง 4 ด้าน	<b>auto</b> , และเลขจำนวนเต็มพร้อมหน่วยที่ใช้
padding-top padding-bottom padding-left padding-right	กำหนดระยะห่างระหว่างข้อความกับ <b>tag</b> ด้านบน ล่าง ซ้าย ขวา ตามลำดับ	<b>auto</b> , และเลขจำนวนเต็มพร้อมหน่วยที่ใช้

# การจัดรูปแบบข้อความและรูปภาพ

16

## □ คุณสมบัติการกำหนดการแสดงผลของรูปภาพ

Property	คำอธิบาย	ค่าที่สามารถกำหนดได้
width	ใช้กำหนดความกว้างให้กับ <b>Element</b>	<b>auto</b> , และเลขจำนวนเต็มพร้อมหน่วยที่ใช้
height	ใช้กำหนดความสูงให้กับ <b>Element</b>	<b>auto</b> , และเลขจำนวนเต็มพร้อมหน่วยที่ใช้
float	ใช้กำหนดตำแหน่งให้กับ <b>Element</b> ซิดซ้ายหรือขวา	<b>left</b> และ <b>right</b>



# การจัดรูปแบบข้อความและรูปภาพ

17

## □ การเพิ่มสีสันให้กับข้อความ

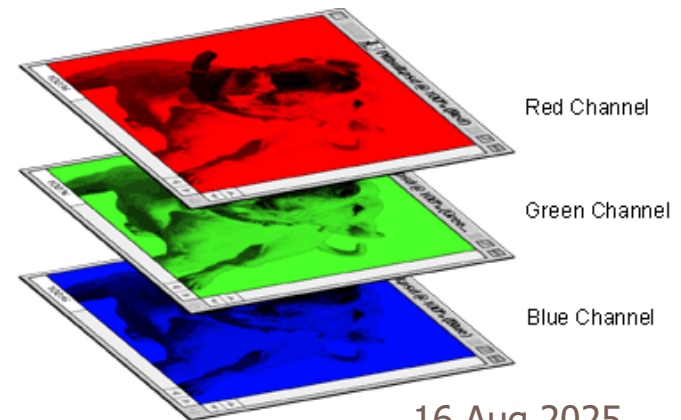
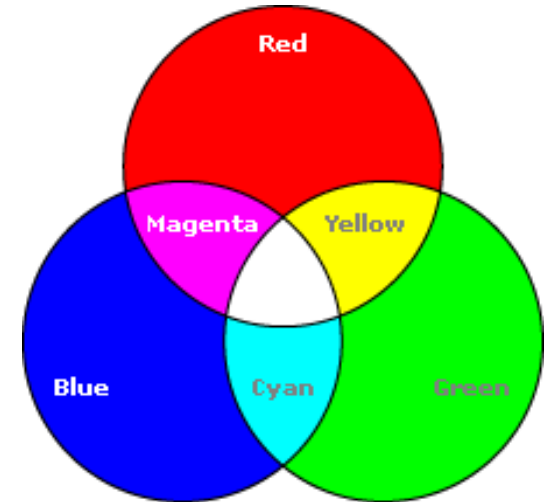
- การเพิ่มสีสันสรรให้กับข้อความถือว่าเป็นวิธีหนึ่งที่มีความน่าสนใจมากขึ้น โดยใช้ Property “color” ในสไตล์ชีทกำหนดสีให้กับข้อความ ซึ่งมีสีมาตรฐาน 16 สี ที่สามารถแสดงผลค่าสี ได้เหมือนกันทุก web browser ได้แก่ black, silver, gray, white, maroon, red, purple, fuchsia, green, lime, olive, yellow, navy, blue, teal, และ aqua
- นอกจากนั้นแล้ว ยังสามารถกำหนดสีให้กับข้อความโดย  
color : rgb (ค่าสีแดง, ค่าสีเขียว, ค่าสีน้ำเงิน)  
color : hsl (ค่าสี hue, ค่าสี saturation, ค่าสี light)

# การจัดรูปแบบข้อความและรูปภาพ

18

## □ ค่าสี RGB

- คือ ค่าสีของแสง แดง เขียว น้ำเงิน
- กำหนดเป็นเลขจำนวนเต็ม 0-255
  - เช่น color : rgb (255, 255, 0)
- กำหนดเป็นสัดส่วน %
  - เช่น color : rgb (100%, 0%, 100%)
- กำหนดเป็นเลขฐาน 16
  - เช่น background-color : #FFCF00



# การจัดรูปแบบข้อความและรูปภาพ

19

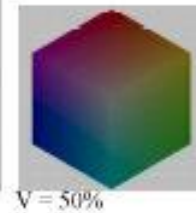
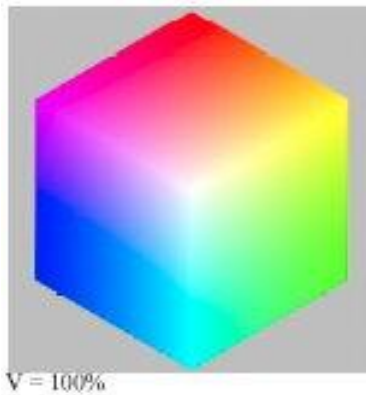
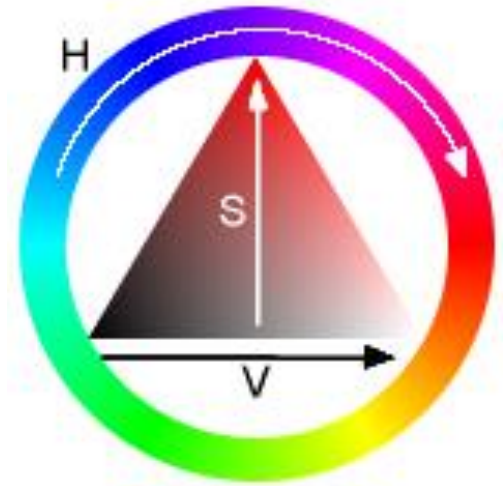
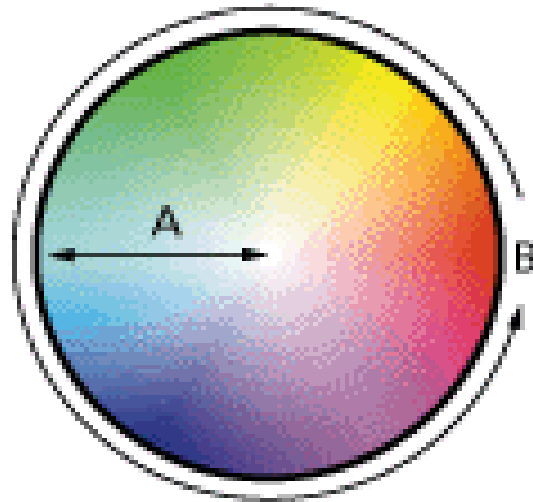
## □ ค่าสี HSL

- คือ ค่าสีของแสงเช่นกัน แต่ปรับการใช้งาน RGB ให้สอดคล้องกับการใช้งานของมนุษย์ (ที่มองสีได้ครั้งละ...สีในคราวเดียวกัน) (บางครั้งในบางวงการจะเรียกโมเดลสีแบบนี้ว่า HSV หรือ HSB แล้วแต่การอธิบาย) มี 3 องค์ประกอบคือ
  - H : HUE – คือ เนื้อสี เป็นสีบริสุทธิ์ที่ได้ตามสเปกตรัมของแสง สำหรับค่าสีจะกำหนดเป็นองศาตามตำแหน่งในวงกลมสี (roughly, dominant wavelength)
  - S : Saturation – คือ ความสดหรือความบริสุทธิ์ของจุดสี (purity) มีค่าเป็นเปอร์เซ็นต์ตั้งแต่ 0-100% โดยค่า 0% จะได้เป็นสีเทาหรือสีขาว ค่าที่ 100% จะได้เป็นสีบริสุทธิ์ที่สุด
  - L : Light (บางครั้งจะเป็น V : Value หรือ B : Brightness แทน) – คือ ความสว่างของจุดสี มีค่าเป็นเปอร์เซ็นต์ ตั้งแต่ 0-100% โดยค่า 0% จะได้เป็นสีดำ และค่า 100% จะได้เป็นสีขาว

# การจัดรูปแบบข้อความและรูปภาพ

20

## □ ค่าสี HSL



# การจัดรูปแบบข้อความและรูปภาพ

21

- การกำหนดความโปร่งใส (Transparency)
  - ▣ เป็นการกำหนดความโปร่งใสให้กับสี โดยมีค่าตั้งแต่ 0 ถึง 1 เช่น  
color : rgba (255, 0, 0, 0.5)  
หมายถึง สีแดงที่มีความโปร่งใส 0.5 หรือ 50%
- แผนผังสี (color scheme)
  - ▣ <http://kuler.adobe.com>
  - ▣ เลือกสีจากการผสมสีหลัก
  - ▣ เลือกสีจากรูปภาพ

# การเพิ่มสีสันให้กับ Link และพื้นหลัง

22

- a:link เป็นการกำหนดสไตล์สีให้กับ tag ในขณะที่ยังไม่ได้ถูกคลิก
- a:visited เป็นการกำหนดสไตล์สีให้กับ tag หลังจากถูกคลิกแล้ว
- a:hover เป็นการกำหนดสไตล์สีให้กับ tag ในขณะที่เลื่อนเมาส์ไปวางบริเวณที่กำหนดให้เป็น Link
- a:active เป็นการกำหนดสไตล์สีให้กับ tag ในขณะที่กำลังคลิก

# การเพิ่มสีให้กับ Link และพื้นหลัง

23

- background-color: สีที่ต้องการ; ใช้กำหนดสีของพื้นหลัง
- โดยปกติแล้วเราสามารถกำหนดสีของพื้นหลังได้
  - ▣ สีของพื้นหลังของเอกสาร กำหนดที่ <body>
  - ▣ สีของพื้นหลังของส่วนต่าง ๆ ก็ให้กำหนดที่ tag นั้นๆ ได้เลย

# การเพิ่มสีสันให้กับ Link และพื้นหลัง

24

- การกำหนดรูปภาพเป็นพื้นหลังชนิดของรูปภาพที่นิยมใช้ได้แก่ GIF, JPG, PNG
- คุณสมบัติที่ใช้กำหนดรูปภาพให้เป็นพื้นหลังเว็บเพจ

Property	คำอธิบาย	ค่าที่สามารถกำหนดได้
background	ใช้กำหนดสไตล์สีให้กับพื้นหลังของเว็บเพจโดยสามารถกำหนดสไตล์สีทั้งหมดที่เกี่ยวข้องกับพื้นหลังได้	ทุกค่า
background-image	ใช้กำหนดพื้นหลังเป็นรูปภาพ	url(ที่อยู่ของรูปภาพ) none
background-repeat	ใช้กำหนดให้แสดงแบบวนซ้ำ	repeat, repeat-x, repeat-y, no-repeat



# การเพิ่มสีสันให้กับ Link และพื้นหลัง

25

- การกำหนดรูปภาพเป็นพื้นหลังชนิดของรูปภาพที่นิยมใช้ได้แก่ GIF, JPG, PNG
- คุณสมบัติที่ใช้กำหนดรูปภาพให้เป็นพื้นหลังเว็บเพจ

Property	คำอธิบาย	ค่าที่สามารถกำหนดได้
background-position	ใช้กำหนดตำแหน่งของพื้นหลัง โดยเปรียบเทียบกับมุมบนซ้าย	top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right. x-% y-%, x-pos y-pos
background-attachment	ใช้กำหนดรูปภาพที่เป็นภาพพื้นหลังให้อยู่กับที่ หรือเลื่อนไปพร้อม ๆ กับการเลื่อนของจอภาพ	fix scroll

# การกำหนดรูปแบบของ Scroll Bar

26

- Scroll Bar มีส่วนประกอบที่สำคัญ 3 ส่วนดังนี้
  - ▣ Scroll Box เป็นกล่องสี่เหลี่ยมที่อยู่บนแถบ Scroll Bar ทำหน้าที่เลื่อนหน้าจอซ้าย-ขวา และบน-ล่าง เพื่อแสดงข้อมูลที่ซ่อนไว้
  - ▣ Scroll Arrow เป็นสัญลักษณ์ลูกศรขึ้น-ลง และซ้าย-ขวา
  - ▣ Track เป็นแถบ Scroll Bar ที่ Scroll Box สามารถเลื่อนขึ้น-ลง และซ้าย-ขวาได้

# การกำหนดรูปแบบของ Scroll Bar

27

□ Scroll Bar มีส่วนประกอบที่สำคัญ 3 ส่วนดังนี้

Property	คำอธิบาย
scrollbar-3dlight-color	ใช้กำหนดสีให้กับขอบซ้ายและขอบบนของ <b>Scroll Bar</b> และ <b>Scroll Arrow</b> จึงทำให้เห็นเป็นภาพ 3 มิติ
scrollbar-arrow-color	ใช้กำหนดสีของลูกศรขึ้น-ลงและซ้าย-ขวาของ <b>Scroll Bar</b>
scrollbar-base-color	ใช้กำหนดสีให้กับพื้นที่ของ <b>Scroll Box</b> และ <b>Track</b> ยกเว้นขอบล่างและขอบขวา
scrollbar-darkshadow-color	ใช้กำหนดสีให้กับขอบขวาและขอบล่างของ <b>Scroll Box</b> ซึ่งตรงข้ามกับ <b>scrollbar-3dlight-color</b>
scrollbar-face-color	ใช้กำหนดสีให้กับ <b>Scroll Box</b> และ <b>Track</b> โดยที่สีของ <b>Scroll Box</b> จะเข้มกว่า <b>Track</b>
scrollbar-highlight-color	ใช้กำหนดสีกับขอบซ้ายและขอบบนของ <b>Scroll Box</b> รวมถึงสีของ <b>Track</b> ด้วย
scrollbar-shadow-color	ใช้กำหนดสีกับขอบขวาและขอบล่างของ <b>Scroll Box</b> โดยจะมีผลถึงขอบขวาและขอบล่างของ <b>Scroll Arrow</b> ด้วย แต่ไม่มีผลกับลูกศรที่อยู่ภายใน
scrollbar-track-color	ใช้กำหนดสีให้กับ <b>Track</b> เพียงอย่างเดียว

# การกำหนดรูปแบบของ Scroll Bar

28

- ฝึกปฏิบัติการกำหนดรูปแบบให้ Scroll Bar
  - ▣ ใช้ news.html ที่ได้สร้างไว้แล้วมาปรับปรุงรูปแบบของ Scroll Bar
  - ▣ ให้ใช้ property ที่ได้เรียนมา
  - ▣ ดูผลการกำหนดรูปแบบของ Scroll Bar



# หัวข้อการบรรยายเกี่ยวกับ CSS ตอนที่ 2

30

- เทคนิคการประยุกต์ใช้งานสไตล์ชีท (ต่อ)
  - ▣ Properties ของสไตล์ชีทที่ใช้กำหนดรูปแบบของตาราง
  - ▣ การใช้สไตล์ชีทในการจัดรูปแบบฟอร์ม
  - ▣ CSS Position และ Layout
- การประยุกต์ใช้ CSS แบบ class
- การประยุกต์ใช้ CSS แบบ id
- แบบฝึกปฏิบัติ C : การตกแต่งเว็บเพจโดยใช้ CSS

# สไตล์ชีทที่ใช้กำหนดรูปแบบของตาราง

31

## □ การกำหนดค่า Property “Border”

Property	คำอธิบาย
Hidden	ซ่อนเส้นตาราง
Dotted	ขอบเป็นจุด
Dashed	ขอบเป็นเส้นสีสลับขาว
Solid	ขอบเป็นเส้นหนาสีอ่อน
Double	ขอบเป็นเส้นคู่
Groove	ขอบเป็นเส้นแบบมีเงา
Ridge	ขอบเป็นเส้นแบบไม่มีเงา
Inset	ขอบบนและซ้ายเป็นสีทึบ
Outset	ขอบบนและซ้ายเป็นสีอ่อน

# สไตลชีทที่ใช้กำหนดรูปแบบของตาราง

32

## □ การกำหนดค่า Property “Border”

### ▣ ตัวอย่าง 1

```
table{ border : 2 px solid #8894F7; }
```

/\* เป็นการกำหนดเส้นขอบให้ตารางขนาด 2 px แบบ solid สี #8894F7 \*/

### ▣ ตัวอย่าง 2

```
table th, table td{ border : 1 px ridge blue; }
```

/\*เป็นการกำหนดเส้นขอบให้กับ th + td ให้มีขนาด 1 px แบบ ridge สี blue\*/

### ▣ ตัวอย่าง 3

```
table{ border-collapse : collapse; }
```

/\*เป็นการกำหนดให้ border ที่ซ้อนกันให้เหลือแค่เส้นเดียว\*/



# สไตลชีทที่ใช้กำหนดรูปแบบของตาราง

33

## □ เพิ่มความสวยงามให้กับตารางโดยการเพิ่มสี

### ▣ ตัวอย่าง 1

```
table{ background : #CCFFFF; }
```

### ▣ ตัวอย่าง 2

```
.name{ text-align : left ; background : #FFFF99; }
```

## □ เพิ่มการตอบสนองต่อการเลื่อนเมาส์ในตารางด้วย “hover”

### ▣ ตัวอย่าง

```
table tr:hover { background : FFFF99; }
```

### ▣ ตัวอย่าง

```
table td:hover { background : FFFF99; }
```

# การใช้สไตล์ชีทในการจัดรูปแบบฟอร์ม

34

## □ คุณสมบัติที่ใช้ในการจัดรูปแบบฟอร์ม

Property	คำอธิบาย	ค่าที่สามารถกำหนดได้
border	ใช้กำหนดเส้นขอบของวัตถุ	ตัวเลขจำนวนเต็มพร้อมหน่วย <b>px</b>
color	ใช้สำหรับตกแต่งสีฟอร์ม	#ค่าสีเป็นเลขฐาน 16
display	ใช้กำหนดการวางตำแหน่งของฟอร์ม	<b>Inline</b>
width	ใช้กำหนดความกว้าง	ตัวเลขจำนวนเต็มพร้อมหน่วย <b>px</b>
height	ใช้กำหนดความสูง	ตัวเลขจำนวนเต็มพร้อมหน่วย <b>px</b>
background-color	ใช้สำหรับตกแต่งสีพื้น	#ค่าสีเป็นเลขฐาน 16
overflow	ใช้กำหนดให้มี <b>scroll</b>	<b>Auto</b> , ตัวเลขจำนวนเต็มพร้อมหน่วย <b>px</b>

# CSS Position

35

- คุณสมบัติที่ใช้กำหนดระยะตำแหน่งของแท็ก

Property	คำอธิบาย	ค่าที่สามารถกำหนดได้
position	กำหนดตำแหน่งของข้อมูลหรือ tag	Static, Absolute, Relative
margin	กำหนดระยะห่างระหว่าง tag กับ tag parent	auto, หรือกำหนดเลขจำนวนเต็ม พร้อมหน่วยที่ใช้
width และ height	กำหนดความกว้างและความสูงของข้อมูลหรือ tag	Auto, หรือกำหนดเลขจำนวนเต็ม พร้อมหน่วยที่ใช้ หรือเปอร์เซ็นต์
text-align และ align และ float	กำหนดตำแหน่งของข้อมูลหรือ tag ให้ชิดซ้าย ชิดขวา หรืออยู่กลาง	center, left, right
padding	กำหนดระยะห่างระหว่างข้อความกับ tag ทั้ง 4 ด้าน	auto, และเลขจำนวนเต็มพร้อมหน่วยที่ใช้
z-index	กำหนดการซ้อนทับกันของข้อมูล	เลขจำนวนเต็มพร้อมหน่วยที่ใช้
clear	คำสั่งยกเลิกการจัดวางข้อความ (ส่วนใหญ่จะยกเลิกการใช้ float)	left, right, both

# การใช้ CSS แบบ class

36

- การใช้ Class Selector – การกำหนด style ของ CSS อาจใช้ class selector ในการกำหนด style ที่แตกต่างกันของ Tag HTML ในแท็กชนิดเดียวกันได้
- เช่นต้องการให้มีรูปแบบการจัดย่อหน้าในเอกสาร 2 รูปแบบ ได้แก่ ย่อหน้าชิดขวา และย่อหน้าอยู่กึ่งกลาง จะกำหนดใน CSS เป็น

```
p.right {text-align:right}
```

```
p.center {text-align:center}
```

- เวลาใช้งาน โดยสามารถกำหนดให้กับแท็กได้หลาย class แต่จะสามารถกำหนดรูปแบบได้เพียง 1 คลาสต่อ 1 แท็ก ไม่สามารถกำหนดรูปแบบหลายคลาสให้กับแท็กเดียวกันได้

# การใช้ CSS แบบ class

37

## □ ตัวอย่างเช่น

```
<p class="right">
```

This paragraph will be right-aligned.

```
</p>
```

```
<p class="center">
```

This paragraph will be center-aligned.

```
</p>
```

# การใช้ CSS แบบ class

38

- กรณีที่ต้องการกำหนดรูปแบบให้กับแท็กอื่นๆ โดยไม่ต้องการระบุแท็กเป็นการเฉพาะ ก็ไม่จำเป็นต้องใส่ชื่อแท็กเลยก็ได้ ตัวอย่างเช่น

```
.center {text-align:center}
```

- การเรียกใช้ CSS แบบ class แบบนี้จะเรียกใช้เป็นแอททริบิวต์ของแท็ก

```
<h1 class="center">
```

This heading will be center-aligned.

```
</h1>
```

```
<p class="center">
```

This paragraph will be center-aligned.

```
</p>
```

# การใช้ CSS แบบ id

39

- การใช้ ID Selector – เป็นการใช้งานโดยการระบุ id ซึ่ง id ที่กำหนดนั้นจะต้องเป็นค่าที่ไม่ซ้ำกัน และ ID Selector สามารถกำหนดให้กับแท็ก HTML ได้เพียง 1 แท็กเท่านั้น เช่น ต้องการกำหนด style ที่มี id เป็น “kim99” ให้มีสีเขียว (green)

```
#kim99 {text-align:center; color:green} /*line1*/
```

```
h1#kim99 {text-align:center; color:red} /*line2*/
```

- ตัวอย่างการนำไปใช้

```
<h1 id=“kim99”> Heading </h1>
```

```
<p id=“kim99”> Paragraph </p>
```





# หัวข้อการบรรยายเกี่ยวกับ HTML Tag

41

- รู้จักกับแท็กอื่นๆ ใน XHTML
  - ▣ <meta>
  - ▣ <frame> (ยกเลิกใน HTML5)
  - ▣ <iframe>
  - ▣ <embed>
  - ▣ การจัดแบ่งหน้าด้วยแท็ก <section>, <article>

# การระบุข้อมูลของหน้าเว็บเพจด้วย MetaData

42

- MetaData เป็นการระบุข้อมูลเพิ่มเติม หรืออธิบายความเป็นมาของข้อมูลหลักไว้ในส่วนของข้อมูลย่อย (metadata : เมต้าดาต้า)
- ข้อมูลที่เกี่ยวข้องของหน้าเว็บเพจ จะอธิบายไว้ในแท็ก <meta> โดยใส่ข้อมูลนี้ไว้ในส่วนแรกของหน้าเว็บ (ส่วน Head) ได้แก่
  - ▣ รายละเอียดเพิ่มเติมของหน้าเว็บเพจ, ผู้สร้าง, วันที่แก้ไขล่าสุด
  - ▣ คีย์เวิร์ด (สำหรับให้ search engine ใช้ในการค้นหา)
  - ▣ ข้อมูลสำหรับ Web Browser ในการโหลดหน้าเว็บหน้านี้ เช่น รูปแบบข้อมูลของหน้าเว็บเพจ, ระบบภาษาที่ใช้ในหน้าเว็บเพจ
  - ▣ คำสั่งอื่นๆ เพิ่มเติม เช่น สั่งให้รีเฟรชทุกๆ 10 วินาที, สั่งให้โหลดหน้าเว็บเพจอื่น เมื่อครบเวลาที่กำหนด, ฯลฯ

# การระบุข้อมูลของหน้าเว็บเพจด้วย MetaData

43

- การกำหนดข้อมูลที่เกี่ยวข้องกับหน้าเว็บเพจ

```
<meta name="description" content="Kimmy's homepage, Kimmy personal page">
```

```
<meta name="description" content="DPU's IT-school">
```

# การระบุข้อมูลของหน้าเว็บเพจด้วย MetaData

44

- การกำหนดคีย์เวิร์ดสำหรับให้ Search Engine ค้นหา

```
<meta name="keywords" content="school,it,ict,information  
technology,web engineering">
```

- การกำหนดตัวอักขระที่ใช้ในเพจ

```
<meta http-equiv="Content-Type" content="text/html;  
charset=UTF-8">
```

# การระบุข้อมูลของหน้าเว็บเพจด้วย MetaData

45

- การสั่งให้เพจทำการโหลดตัวเองซ้ำทุก 10 วินาที

```
<meta http-equiv="Refresh" content="10">
```

- การสั่งให้เพจทำการโหลดเพจอื่นขึ้นมาแทนที่ เมื่อถึงเวลาที่กำหนดไว้

```
<meta http-equiv="Refresh" content="10;  
URL=book.html">
```

# การสร้าง Frame (ยกเลิกใน HTML5)

46

- เป็นการแบ่งหน้าจอของ web browser ออกเป็นส่วนย่อย ๆ ทำให้สามารถใช้พื้นที่ทั้งหมดบนหน้าจอได้อย่างมีประสิทธิภาพ
- วิธีการสร้างเฟรม
  - ▣ แบ่งโครงสร้างส่วนต่าง ๆ ในหน้าเว็บเพจ ด้วยการสร้างเฟรมเพจ
  - ▣ กำหนดเว็บเพจที่ต้องการให้แสดงในเฟรม
- <frameset> ใช้กำหนดลักษณะของเฟรมบนหน้าจอว่าจะแบ่งแต่ละส่วนเป็นแนวตั้ง (cols) หรือแนวนอน (rows)
- การสร้างเฟรมเพจให้ใช้ <frameset> แทน <body>

# การสร้าง Frame (ยกเลิกใน HTML5)

47

## □ ตัวอย่าง 1

```
<frameset rows="20%, 30%, *">
```

```
  <frame name="frame1" src="page1.html" />
```

```
  <frame name="frame2" src="page2.html" />
```

```
  <frame name="frame3" src="page3.html" />
```

```
  <noframes>
```

```
    <body>
```

เว็บเพจที่สร้างเพื่อไว้ในกรณีที่ browser ดูเฟรมไม่ได้

```
    </body>
```

```
  </noframes>
```

```
</frameset>
```

# การสร้าง Frame (ยกเลิกใน HTML5)

48

## □ ตัวอย่าง 2

```
<frameset cols="20%, 30%, *">  
    <frame name="frame1" src="page1.html" />  
    <frame name="frame2" src="page2.html" />  
    <frame name="frame3" src="page3.html" />  
</frameset>
```



# การสร้าง Frame (ยกเลิกใน HTML5)

49

## □ ตัวอย่าง 3

```
<frameset rows="20%, *">
```

```
  <frame name="frame1" src="page1.html" />
```

```
  <frameset cols="30%,*">
```

```
    <frame name="frame2" src="page2.html" />
```

```
    <frame name="frame3" src="page3.html" />
```

```
  </frameset>
```

```
</frameset>
```

# การสร้าง Frame (ยกเลิกใน HTML5)

50

- การกำหนดลักษณะของกรอบเฟรม

```
<frameset rows="20%, *" border="5" frameborder="yes">  
  <frame name="frame1" src="page1.html" scrolling="No"  
  noresize="noresize" />  
  <frameset cols="30%,*">  
    <frame name="frame2" src="page2.html" scrolling="No"  
    noresize="noresize" />  
    <frame name="frame3" src="page3.html" scrolling="No"  
    noresize="noresize" />  
  </frameset>  
</frameset>
```

# การสร้าง Frame (ยกเลิกใน HTML5)

51

- การสร้าง Link แบบระบุเฟรม
  - ▣ ระบุชื่อเฟรมลงไปใน Attribute Target ของ `<a href ...>` ได้เลย

# การสร้าง Frame (ยกเลิกใน HTML5)

52

## □ การแทรกเฟรมภายในเว็บเพจด้วย <iframe>

### □ รูปแบบ

```
<iframe src="ชื่อไฟล์ HTML หรือ URL"> ... </iframe>
```

### □ width : เป็นการกำหนดความกว้างของ iframe

### □ height : เป็นการกำหนดความสูงของ iframe

### □ ตัวอย่าง

```
<body>
```

```
  <iframe src="http://www.it.dpu.ac.th/" width="300"  
  height="200"> </iframe>
```

```
</body>
```

# การเรียก “ปลั๊กอิน” สำหรับการเล่นไฟล์มัลติมีเดีย

53

- ใช้แท็ก <embed> ไม่ใช่ Tag มาตรฐานที่ใช้ในเอกสาร XHTML แต่ Web Browser ส่วนใหญ่สามารถแสดงผลได้
- เป็น Tag ที่บอกให้ Web Browser เรียก “ปลั๊กอิน” ของไฟล์มัลติมีเดียที่ต้องการเล่นขึ้นมาแสดงบนเว็บเพจ (เพื่อเปิดไฟล์นั้นๆ –ปลั๊กอินขึ้นอยู่กับว่าเครื่องคอมพิวเตอร์นั้นๆ ติดตั้งปลั๊กอินอะไรอยู่ที่สามารถเล่นไฟล์ดังกล่าวได้)
- วิธีการเรียกใช้

```
<embed src="ชื่อไฟล์" width="300" height="300"
autostart="true">
```

# การจัดแบ่งหน้าเว็บเพจออกเป็นส่วนๆ

54

- เป็นการแบ่งส่วนของหน้าเว็บเพจ เพื่อให้สามารถกำหนดส่วนที่สอดคล้องกันให้อยู่ในกลุ่มเดียวกัน (สะดวกต่อการดูแลในภายหลัง)
- การตกแต่งเพิ่มเติม เช่น ย่อหน้า ควรใช้แท็ก `<p>` `<div>` และ list แทนการแบ่งส่วน
- แท็กที่เกี่ยวข้อง
  - ▣ `<section>` กำหนดส่วนหลักของหน้าเว็บเพจ
  - ▣ `<article>` กำหนดส่วนย่อยของหน้าเว็บเพจ
  - ▣ `<nav>` กำหนดส่วนที่จะวางลิงก์
  - ▣ `<aside>` กำหนดส่วนที่เป็นเนื้อหาข้างเคียง

# การจัดแบ่งหน้าเว็บเพจออกเป็นส่วนๆ

55

- แท็กที่เกี่ยวข้อง (ต่อ)
  - ▣ <header> กำหนดส่วนหัวของหน้าเว็บเพจ
  - ▣ <footer> กำหนดส่วนท้ายของหน้าเว็บเพจ
  - ▣ <h1>...<h6> กำหนดหัวข้อ
  - ▣ <hgroup> สำหรับการจัดกลุ่มแท็ก <h1> ... <h6> หากกำหนดหัวข้อแบบมีหลายชั้น
  - ▣ <address> สำหรับระบุข้อมูลสำหรับติดต่อผู้เขียน/ผู้สร้างเว็บเพจ (จะปรับให้เป็นตัวเอียงอัตโนมัติ)

# การจัดแบ่งหน้าเว็บเพจออกเป็นส่วนๆ

56

## □ ตัวอย่างการใช้งาน

```
<body>
```

```
  <article>
```

```
    <header> <h1> หนังสืออ่านเล่น </h1> </header>
```

```
    <section>
```

1. วาบ วาบ พิกพมหัสจรรย์ <br />
2. หอสมุดอุทัย : บทผู้ครองหนังสือต้องห้าม <br />

```
  </section>
```

```
  <header> <h1> หนังสือจิตวิทยา </h1> </header>
```

```
  <section>
```

1. นิยายแบบนี้เลือกกรุป B แน่ๆ <br />
2. ก็ยังดีที่ได้รัก <br />

```
  </section>
```

```
</article>
```

```
</body>
```

ความรู้เบื้องต้นเกี่ยวกับ HTML5 และ CSS

## หนังสืออ่านเล่น

1. วาบ วาบ พิกพมหัสจรรย์
2. หอสมุดอุทัย : บทผู้ครองหนังสือต้องห้าม

## หนังสือจิตวิทยา

1. นิยายแบบนี้เลือกกรุป B แน่ๆ
2. ก็ยังดีที่ได้รัก

16 Aug 2025



# การจัดแบ่งหน้าเว็บเพจออกเป็นส่วนๆ

57

## □ ตัวอย่างการใช้งานร่วมกับ CSS

```
<head>
```

```
  <title> Book List </title>
```

```
    <style type="text/css">
```

```
      .novel{background-color:yellow;}
```

```
      .mind{background-color:pink;}
```

```
    </style>
```

```
</head>
```

```
<body>
```

```
  <article>
```

```
    <header> <h1> หนังสืออ่านเล่น </h1> </header>
```

# การจัดแบ่งหน้าเว็บเพจออกเป็นส่วนๆ

58

## □ ตัวอย่างการใช้งานร่วมกับ CSS (ต่อ)

```
<section><div class="novel">
<ol>
  <li> วาบ วาบ พิกพมหัสจรรย์ </li>
  <li> หอสมุดลูเพียส : บทผู้ครองหนังสือต้องห้าม </li>
</ol> </div> </section>

<header> <h1> หนังสือจิตวิทยา </h1> </header>

<section><div class="mind">
<ul>
  <li> นิสัยแบบนี้เลือกกรุป B แน่ๆ </li>
  <li> ก็ยังดีที่ได้รัก </li>
</ul> </div> </section>

</article>

</body>
```

### หนังสืออ่านเล่น

1. วาบ วาบ พิกพมหัสจรรย์
2. หอสมุดลูเพียส : บทผู้ครองหนังสือต้องห้าม

### หนังสือจิตวิทยา

- นิสัยแบบนี้เลือกกรุป B แน่ๆ
- ก็ยังดีที่ได้รัก



# หัวข้อการบรรยาย Multimedia Tag

60

- Multimedia Tag in HTML5
  - ▣ แท็กเกี่ยวกับเสียง <Audio>
  - ▣ แท็กเกี่ยวกับวิดีโอ (Video)

# แท็กเกี่ยวกับเสียง (Sound)

61

- แท็กที่เกี่ยวกับการแทรกไฟล์เสียงใน HTML5 คือ <audio>
- ไฟล์เสียงที่ใช้งานได้ คือ .mp3, .wav, .ogg (ไฟล์ .ogg เหมาะสำหรับการใช้แบบสตรีมมิ่งหรือการกระจายเสียง ซึ่งจะมาแทน .mp3)
- วิธีการเรียกใช้ คือ

```
<audio src="ชื่อไฟล์"></audio>
```

- ตัวอย่างการใช้งาน

```
<audio src="song.mp3"></audio>
```

# แท็กเกี่ยวกับเสียง (Sound)

62

## □ แอทธิบิวส์ของแท็ก <audio>

- ▣ autoplay : เป็นการสั่งให้ไฟล์เพลงเล่นเองโดยอัตโนมัติเมื่อเริ่มโหลดเพจ เช่น

```
<audio src="song.mp3" autoplay> </audio>
```

- ▣ controls : เป็นการสั่งให้แสดงแถบควบคุมการเล่น โดยผู้ใช้งานจะสามารถสั่งให้เริ่มเล่น, หยุดเล่น และปรับค่าความดังของเสียงได้ เช่น

```
<audio src="song.mp3" controls> </audio>
```



# แท็กเกี่ยวกับเสียง (Sound)

63

## □ แอทธิบิวส์ของแท็ก <audio>

▣ preload : เป็นการสั่งให้ทำการโหลดไฟล์เสียงก่อนการกดปุ่มเล่น (ถ้าใช้ autoplay จะเป็นการเล่นไฟล์เสียงแบบสตรีมตั้งแต่ตอนโหลดเว็บเพจ) โดยสามารถกำหนดค่าการ preload ได้ 3 ค่า คือ

■ none : ไม่ต้องการให้ browser โหลดไฟล์เสียงตอนโหลดเว็บเพจ

■ metadata : กำหนดให้ browser โหลดไฟล์เสียงเฉพาะแค่ metadata ของไฟล์เสียงมาเท่านั้น ตอนที่โหลดเว็บเพจ

■ auto : กำหนดให้ browser โหลดไฟล์เสียงมาทั้งไฟล์ตอนโหลดเว็บเพจ

การใช้งาน preload

```
<audio src="song.mp3" autoplay="ค่า"> </audio>
```

# แท็กเกี่ยวกับเสียง (Sound)

64

ตัวอย่างการใช้งาน preload

```
<audio src="song.mp3" autoplay="none"> </audio>
```

```
<audio src="song.mp3" autoplay="metadata"> </audio>
```

```
<audio src="song.mp3" autoplay="auto"> </audio>
```

▣ loop : เป็นการสั่งให้ทำการเล่นไฟล์เสียงนั้นซ้ำ เมื่อเล่นจบรอบ เช่น

```
<audio src="song.mp3" loop> </audio>
```



# แท็กเกี่ยวกับเสียง (Sound)

65

- Web Browser ที่รองรับการใช้งานแท็ก <audio>

Browser	MP3	WAV	OGG
Chrome	Yes	No	Yes
Firefox	No	Yes	Yes
Internet Explorer 9*	Yes	No	No*
Opera	No	Yes	No
Safari	Yes	Yes	No

\* Microsoft announced that IE9 would be supporting the OGG format, but in the beta version of IE9, it did not.

# แท็กเกี่ยวกับเสียง (Sound)

66

- จากการที่ Web Browser ที่รองรับการใช้ไฟล์เสียงแตกต่างกันแต่ละฟอร์แมต ดังนั้นผู้สร้างเว็บเพจ อาจจะต้องใช้ไฟล์เสียงหลายๆ ชนิดร่วมกัน
- วิธีการคือ ใช้แท็ก <source> สำหรับเรียกไฟล์เสียงแต่ละชนิด

```
<audio controls>
```

```
  <source src="ชื่อไฟล์#1.wav">
```

```
  <source src="ชื่อไฟล์#2.mp3">
```

```
  <source src="ชื่อไฟล์#3.ogg">
```

```
</audio>
```

# แท็กเกี่ยวกับเสียง (Sound)

67

- เพื่อให้สามารถเล่นไฟล์เสียงได้อย่างรวดเร็ว อาจจะใช้ **แอททริบิวต์ type** ของแท็ก `<source>` สำหรับให้ web browser พิจารณานำไฟล์ฟอร์แมตไหน (ที่เบราว์เซอร์รองรับ) ขึ้นมาเล่น

```
<audio controls>
```

```
  <source src="ชื่อไฟล์#1.wav" type="audio/wave">
```

```
  <source src="ชื่อไฟล์#2.mp3" type="audio/mpeg">
```

```
  <source src="ชื่อไฟล์#3.ogg" type="audio/ogg">
```

```
</audio>
```

# แท็กเกี่ยวกับเสียง (Sound)

68

- ไฟล์เสียงบางไฟล์อาจจะมีการเข้ารหัส หรือมีการบีบอัดอยู่ เพื่อลดขนาดของไฟล์ให้เล็กลง (codec) อาจจะต้องใช้คำสั่ง codecs เพิ่มเพื่อระบุการใช้งานโปรแกรมสำหรับถอดรหัส

```
<audio controls>
```

```
    <source src="ชื่อไฟล์#1.wav" type="audio/wave;  
    codecs=wav">
```

```
    <source src="ชื่อไฟล์#2.mp3" type="audio/mpeg ;  
    codecs=mp3">
```

```
    <source src="ชื่อไฟล์#3.ogg" type="audio/ogg ;  
    codecs=vorbis">
```

```
</audio>
```

# แท็กเกี่ยวกับวิดีโอ (Video)

69

- มาตรฐานของไฟล์วิดีโอที่จะสามารถเล่นบนเว็บเพจได้
  - ▣ H.264 เป็นมาตรฐานการบีบอัดข้อมูลของสัญญาณภาพและเสียง ได้แก่ ไฟล์ชนิด MPEG-4 หรือ \*.mp4 และ \*.wav
  - ▣ OGG เป็นมาตรฐานการบีบอัดข้อมูลโดย xiph.org ได้แก่ ไฟล์ชนิด \*.ogv
  - ▣ WEBM เป็นมาตรฐานการบีบอัดข้อมูลโดย google ได้แก่ ไฟล์ชนิด \*.webm
  - ▣ 3GP เป็นมาตรฐานไฟล์วิดีโอและเสียงบนอุปกรณ์เคลื่อนที่ (โทรศัพท์มือถือ) ได้แก่ ไฟล์ชนิด \*.3gp

# แท็กเกี่ยวกับวิดีโอ (Video)

70

## □ คำสั่งที่ใช้

```
<video src="ชื่อไฟล์"></video>
```

## □ ตัวอย่างเช่น

```
<video src="kimkim.mp4"></video>
```

# แท็กเกี่ยวกับวิดีโอ (Video)

71

## □ แอททริบิวต์สำหรับแท็ก <video>

- poster : ในขณะที่ไฟล์วิดีโอซึ่งมีขนาดใหญ่กำลังโหลดไฟล์มาอยู่นั้น (และอินเทอร์เน็ตมีความเร็วช้าด้วย) อาจจะอนุญาตให้แสดงภาพได้ระหว่างที่รอการโหลดอยู่

```
<video src="kim.mp4" poster="message.png"></video>
```

- preload : อนุญาตให้มีการโหลดไฟล์วิดีโอมาในขณะที่โหลดเว็บเพจด้วย

```
<video src="kim.mp4" preload></video>
```

# แท็กเกี่ยวกับวิดีโอ (Video)

72

- แอททริบิวต์สำหรับแท็ก <video>
  - ▣ loop : อนุญาตให้เล่นวิดีโอซ้ำเมื่อครบรอบ

```
<video src="kim.mp4" loop></video>
```

```
<video loop>
```

```
<video loop="loop">
```

```
<video loop="">
```



# แท็กเกี่ยวกับวิดีโอ (Video)

73

□ แอทธิบิวส์สำหรับแท็ก <video>

▣ autoplay : อนุญาตให้เล่นวิดีโอได้ทันทีขณะที่มีการโหลดเว็บเพจเสร็จ

```
<video src="kim.mp4" autoplay></video>
```

```
<video autoplay>
```

```
<video autoplay="autoplay">
```

```
<video autoplay="">
```

# แท็กเกี่ยวกับวิดีโอ (Video)

74

□ แอททริบิวต์สำหรับแท็ก <video>

▣ controls : อนุญาตให้แสดงส่วนควบคุมวิดีโอ

```
<video src="kim.mp4" controls></video>
```

- กด play ให้เริ่มเล่นได้
- กด stop ให้หยุดเล่นได้
- ปรับระดับเสียงได้
- บอกเวลาของไฟล์วิดีโอได้
- เลื่อนตำแหน่งที่เริ่มเล่นได้

# แท็กเกี่ยวกับวิดีโอ (Video)

75

## □ แอทธิบิวส์สำหรับแท็ก <video>

### ▣ width/height : กำหนดขนาดหน้าจอของวิดีโอ

```
<video src="kim.mp4" width="352"  
height="288"></video>
```

## □ ตัวอย่างการใช้งานแท็ก <video>

```
<video width=30% src="video1.mp4" controls  
preload="auto"></video>
```

# แท็กเกี่ยวกับวิดีโอ (Video)

76

## □ การรองรับของเว็บเบราว์เซอร์

Browser	H.264	OGG	WebM	3GP
Chrome	No	Yes	Yes	No
Firefox	No	Yes	Unknown	No
Internet Explorer 9	*No	No	No	Yes
Opera	No	Yes	Yes	No
Safari	Yes	No	No	Yes
Safari Mobile	No	No	No	Yes

\* Microsoft announced that it would support H.264, but IE9 was unable to play MP4 format at the time of testing the beta version of the browser.

# แท็กเกี่ยวกับวิดีโอ (Video)

77

- จากการที่ Web Browser ที่รองรับการใช้ไฟล์วิดีโอแตกต่างกันแต่ละฟอร์แมต ดังนั้นผู้สร้างเว็บเพจ อาจจะต้องใช้ไฟล์หลายๆ ชนิดร่วมกัน
- วิธีการคือ ใช้แท็ก <source> สำหรับเรียกไฟล์วิดีโอแต่ละชนิด

```
<video controls>
```

```
<source src="ชื่อไฟล์#1.mp4">
```

```
<source src="ชื่อไฟล์#2.3gp">
```

```
<source src="ชื่อไฟล์#3.webm">
```

```
Your browser does not support the video tag!
```

```
</video>
```

# แท็กเกี่ยวกับวิดีโอ (Video)

78

- เพื่อให้สามารถเล่นไฟล์วิดีโอได้อย่างรวดเร็ว อาจจะใช้ **แอททริบิวต์ type** ของแท็ก `<source>` สำหรับให้ web browser พิจารณานำไฟล์ฟอร์แมตไหน (ที่เบราว์เซอร์รองรับ) ขึ้นมาเล่นได้ (เว็บเบราว์เซอร์จะพิจารณาจากด้านบนลงด้านล่าง)

```
<video controls>
```

```
  <source src="ชื่อไฟล์#1.mp4" type="audio/mp4">
```

```
  <source src="ชื่อไฟล์#2.3gpp" type="audio/3gpp">
```

```
  <source src="ชื่อไฟล์#3.webm" type="audio/webm">
```

```
Your browser does not support the video tag!
```

```
</video>
```

# แท็กเกี่ยวกับวิดีโอ (Video)

79

- ไฟล์วิดีโอบางไฟล์อาจจะมีการเข้ารหัส หรือมีการบีบอัดอยู่ เพื่อลดขนาดของไฟล์ให้เล็กลง (codec) อาจจะต้องใช้คำสั่ง codecs ร่วมด้วย เพื่อระบุการใช้งานโปรแกรมสำหรับถอดรหัส

```
<video controls>
```

```
  <source src="ชื่อไฟล์#1.mp4" type="video/mp4;  
  codecs=mp4v.20.8">
```

```
  <source src="ชื่อไฟล์#2.3gp" type="video/3gpp;  
  codecs=">
```

```
  <source src="ชื่อไฟล์#3.webm" type="video/webm;  
  codecs=">
```

```
    Your browser does not support the video tag!
```

```
</video>
```

# แท็กเกี่ยวกับวิดีโอ (Video)

80

- **MP4** = MPEG 4 files with H264 video codec and AAC audio codec
- **WebM** = WebM files with VP8 video codec and Vorbis audio codec
  - ▣ เฉพาะ Opera และ Chrome เท่านั้นที่ support ฟอร์แมตนี้ ถ้าต้องการแปลงไฟล์จากฟอร์แมต webm ไปเป็นฟอร์แมตอื่น แนะนำให้ลองเข้าไปที่ [www.mirovideoconverter.com](http://www.mirovideoconverter.com)
- **Ogg** = Ogg files with Theora video codec and Vorbis audio codec



## HTML5 with New API

16 Aug 2025

อ.วิญญู นีรนาทล้ำพงศ์ (winyou@gmail.com)

# หัวข้อการบรรยาย

82

- รู้จักกับ API
- New API ของ HTML 5
- Canvas

# รู้จักกับ API

83

- API (Application Programming Interface) คือ ช่องทางการเชื่อมต่อจากระบบหนึ่งไปยังอีกระบบหนึ่ง
  - ▣ เพื่อการเรียกใช้บริการจากระบบภายนอกจากระบบเดิม
  - ▣ เพื่อเพิ่มความสามารถให้กับระบบเดิม
  - ▣ เพื่อเพิ่มความสะดวกในการใช้บริการ
- API เป็นเสมือนออบเจ็คตัวหนึ่ง ที่มีหน้าที่ตอบสนองต่อการเรียกใช้ โดยจะรับค่าบางอย่างเข้ามาในตัว API และคืนค่ากลับไปในรูปแบบบริการต่างๆ เช่น Facebook API, Google Map API เป็นต้น

# APIs ของ HTML5

84

- ภาษา HTML 5 มี API ให้สามารถเรียกใช้ได้หลายตัว เพื่ออำนวยความสะดวกให้ผู้พัฒนาใช้งานได้สะดวกมากขึ้น ในที่นี้ขอยกตัวอย่างเช่น
  - Canvas
  - Drag & Drop
  - History
  - Inline Editing
  - Messaging
  - Offline Apps
  - Video & Audio
  - Geolocation
  - Local Storage
  - Selectors
  - Server Events
  - Web Sockets
  - Workers

# APIs ของ HTML5

85

□ ภาษา HTML 5 มี API ให้สามารถเรียกใช้ได้หลายตัว เพื่ออำนวยความสะดวกให้ผู้พัฒนาใช้งานได้สะดวกมากขึ้น ในที่นี้ขอยกตัวอย่างเช่น

□ Canvas

□ Drag & Drop

□ History

□ Inline Editing

□ Messaging

□ Offline Apps

□ Video & Audio

□ Geolocation

□ Local Storage

□ Selectors

□ Server Events

□ Web Sockets

□ Workers

ดูเพิ่มเติมที่ [www.whatwg.org/html5/](http://www.whatwg.org/html5/)

ดูเพิ่มเติมที่ [dev.w3.org/html5/](http://dev.w3.org/html5/)

# รู้จักกับ Canvas

86

- Canvas คือ กระดานแสดงภาพกราฟิกบนภาษา HTML5 (เปรียบเหมือนเป็นพื้นผ้าใบวาดภาพของจิตรกร)
- เป็น API ชนิดหนึ่ง สามารถเรียกใช้ Feature นี้ได้โดยใช้ร่วมกับภาษา Java Script และ CSS
- ประโยชน์
  - ใช้วาดภาพกราฟิก
  - สร้างเกม
  - แสดงกราฟ
  - ใช้แสดงภาพ
  - เพิ่มลูกเล่นในเว็บ

# รู้จักกับ Canvas

87

- Canvas เป็นความสามารถใหม่ของ HTML5 ที่จะนำมาใช้แทนปลั๊กอินได้
- Canvas มักจะใช้คู่กับ JavaScript
  - ▣ สร้าง UI element
  - ▣ ควบคุมการทำงานเกี่ยวกับรูปภาพ
  - ▣ สร้างภาพเคลื่อนไหว
  - ▣ ส่วนติดต่อกับเมาส์และคีย์บอร์ด
- Canvas ใช้ได้กับเบราว์เซอร์ที่สนับสนุน HTML5 เท่านั้น

# การใช้งาน Canvas วาดภาพ 2 มิติ

88

- แท็ก <canvas> เป็นตัวกลางสำหรับควบคุมและจัดการพื้นที่ canvas
- การสร้างพื้นที่ Canvas

**CSS :**

```
#myCanvas1{ border:solid 3px red; }  
#myCanvas2{ border:dashed 3px red; }
```

**Body :**

```
<canvas width="320" height="240" id="myCanvas1">  
    Your browser not support Canvas  
</canvas>
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

89

- วิธีตรวจสอบว่าเบราว์เซอร์รองรับ HTML5 หรือไม่ คือ การแทรกข้อความเข้าไประหว่าง `<canvas> ... </canvas>`
- เบราว์เซอร์ที่รองรับ HTML5 จะไม่เห็นข้อความดังกล่าว

```
1 <html>
2   <head>
3     <title>Canvas</title>
4   </head>
5
6   <body>
7     <canvas id="colorScheme" width="600"
8       height="100">If you see this, You <u>really</u>
9       need to get an HTML5 Compatible
       browser.</canvas>
6   </body>
1  </html>
```

# การใช้งาน Canvas วาดภาพ 2 มิติ

90

- ในหนึ่งเพจสามารถมี <canvas> ได้ไม่จำกัด
- รูปแบบคำสั่ง

```
<canvas id="....." width="....." height=".....">
```

```
</canvas>
```

- ▣ id คือ ชื่อของพื้นที่ที่จะทำงาน
- ▣ width และ height คือ ความกว้างและความสูงของพื้นที่ที่จะทำงาน

# การใช้งาน Canvas วาดภาพ 2 มิติ

91

## □ ใส่กรอบให้ Canvas

```
<canvas id="....." width="....." height="....." style="border: 1px solid  
#000000;"></canvas>
```

- ▣ 1px คือ ความกว้างเส้นขอบ
- ▣ solid คือ ลักษณะของเส้นขอบ (เส้นทึบ)
- ▣ #000000 คือ สีของเส้นขอบ (สีดำ)

# การใช้งาน Canvas วาดภาพ 2 มิติ

92

- วาดภาพในแคนวาสต้องใช้คำสั่ง JavaScript แทรกไว้หลังแท็ก

`<canvas>...</canvas>` ที่จะใช้งาน เช่น

```
<script type="text/javascript">
```

```
    var c = document.getElementById(“ชื่อแคนวาส”);
```

```
    var ctx = c.getContext(“2d”);
```

```
    //คำสั่งวาดรูป
```

```
</script>
```

# การใช้งาน Canvas วาดภาพ 2 มิติ

93

- แคนวาสมี 2 วิธีในการวาดภาพ
  - ▣ stroke : ใช้สำหรับลากเส้นตรงหรือเส้นขอบ โดยสามารถกำหนดความหนาของเส้น สี รูปร่าง และข้อความได้
  - ▣ fill : ใช้สำหรับเทสีลงในพื้นที่ปิดหรือตัวอักษรที่กำหนด โดยสีที่ใช้สามารถใช้ได้เป็น
    - สีทึบ (solid)
    - ไล่สีแบบเส้นตรง (linear gradient)
    - ไล่สีแบบรัศมี (radial gradient)
    - ใช้รูปภาพมาทำเป็นพื้นหลัง (image pattern)

# การใช้งาน Canvas วาดภาพ 2 มิติ

94

## □ คำสั่งวาดรูปสี่เหลี่ยมผืนผ้า

### ▣ แบบทึบ

```
ctx.fillStyle=".....";
```

```
ctx.fillRect(x, y, w, h);
```

- `fillStyle` คือ รูปแบบการทาสี
- `fillRect` คือ สั่งให้วาดรูปสี่เหลี่ยมมุมฉากที่ตำแหน่ง (x,y) โดยมีความกว้างเท่ากับ w และความสูงเท่ากับ h และใช้รูปแบบการทาสีตามที่ได้กำหนดไว้

# การใช้งาน Canvas วาดภาพ 2 มิติ

95

## □ คำสั่งวาดรูปสี่เหลี่ยมผืนผ้า

### ▣ แบบ โปร่ง

```
ctx.strokeStyle=".....";
```

```
ctx.strokeRect(x, y, w, h);
```

- `strokeStyle` คือ รูปแบบเส้นขอบ
- `strokeRect` คือ สั่งให้วาดรูปสี่เหลี่ยมมุมฉากที่ตำแหน่ง (x,y) โดยมีความกว้างเท่ากับ w และความสูงเท่ากับ h และใช้รูปแบบเส้นขอบตามที่ได้กำหนดไว้

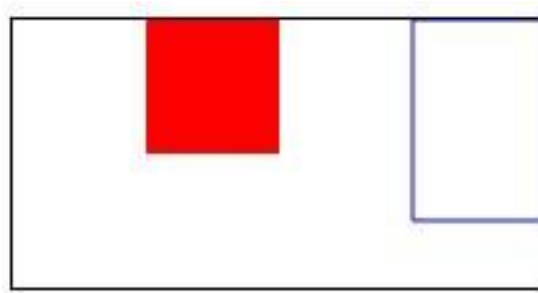
# การใช้งาน Canvas วาดภาพ 2 มิติ

96

## □ คำสั่งวาดรูปสี่เหลี่ยมผืนผ้า

### ▣ ตัวอย่างเช่น

- `ctx.fillStyle="#FF0000";`  
`ctx.fillRect(50, 0, 50, 50);`
- `ctx.strokeStyle="blue";`  
`ctx.strokeRect(150, 0, 50, 75);`





# การใช้งาน Canvas วาดภาพ 2 มิติ

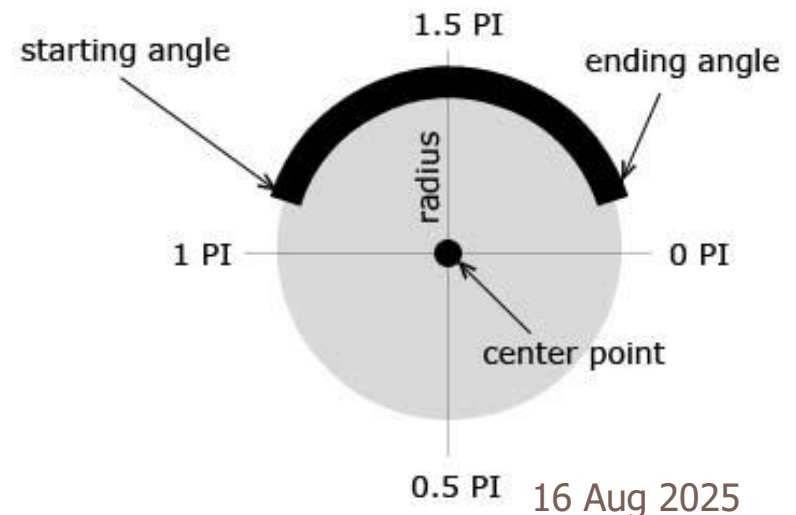
97

- คำสั่งวาดส่วนของวงกลม (arc) และวงกลม
  - ▣ หน่วยขององศาที่ใช้คือ เรเดียน (radian)

```
ctx.beginPath();
```

```
ctx.arc(x, y, r, องศาเริ่ม, องศาจบ);
```

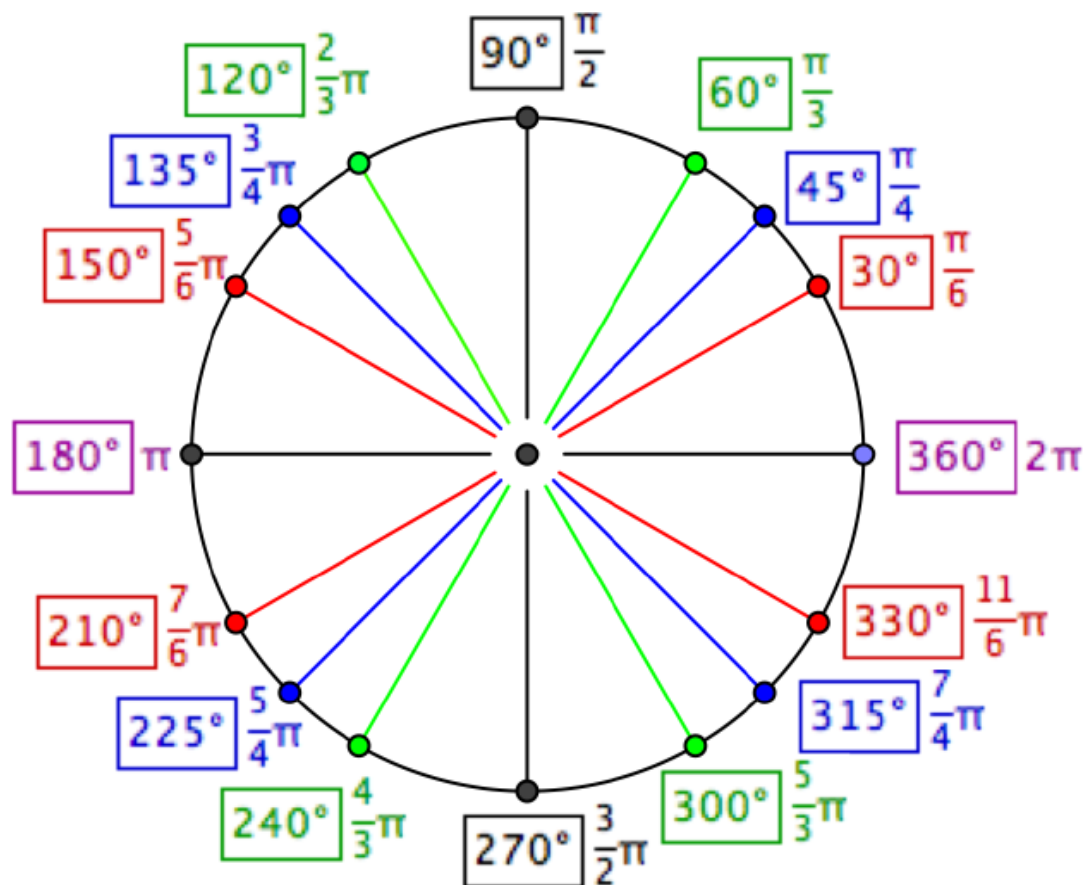
```
ctx.stroke();
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

98

- คำสั่งวาดส่วนของวงกลม (arc) และวงกลม

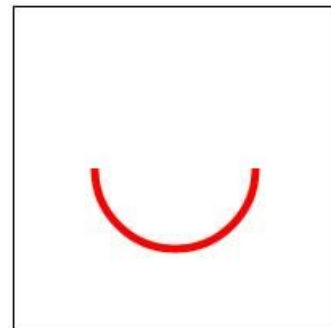


# การใช้งาน Canvas วาดภาพ 2 มิติ

99

- คำสั่งวาดส่วนของวงกลม (arc) และวงกลม
  - ▣ ตัวอย่างเช่น

```
ctx.beginPath();  
ctx.arc(100, 100, 50, 0, Math.PI);  
ctx.strokeStyle="red";  
ctx.lineWidth=5;  
ctx.stroke();
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

100

## □ คำสั่งวาดตัวอักษร

### ▣ แบบทึบ

```
ctx.font="ขนาดตัวอักษร ชนิดตัวอักษร";
```

```
ctx.fillText(".", x, y);
```

### ▣ แบบโปร่ง

```
ctx.font="ขนาดตัวอักษร ชนิดตัวอักษร";
```

```
ctx.strokeText(".", x, y);
```

# การใช้งาน Canvas วาดภาพ 2 มิติ

101

## □ คำสั่งวาดตัวอักษร

### ▣ ตัวอย่างเช่น

```
ctx.font="30px Arial";  
ctx.fillStyle="red";  
ctx.fillText("Hello World",10,50);  
  
ctx.strokeStyle="blue";  
ctx.strokeText("Hello World",10,100);
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

102

- คำสั่งสร้างงานไล่สีสำหรับนำไปใช้กับ element อื่นๆ
  - ▣ แบบเส้นตรง (Linear gradient)

```
var grd = ctx.createLinearGradient(x0, y0, x1, y1);
```

```
grd.addColorStop(ค่า offset, สี);
```

```
grd.addColorStop(ค่า offset, สี);
```

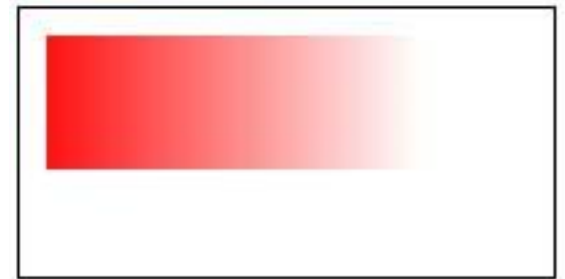
- ค่า offset คือ ระยะห่างจากจุดเริ่มต้น มีค่าอยู่ระหว่าง 0-1
- ต้องมี ColorStop อย่างน้อยสองค่า

# การใช้งาน Canvas วาดภาพ 2 มิติ

103

- คำสั่งสร้างงานไล่สีสำหรับนำไปใช้กับ element อื่นๆ
  - ▣ ตัวอย่างเช่น

```
var grd=ctx.createLinearGradient(0, 0, 150, 0);  
grd.addColorStop(0, "red");  
grd.addColorStop(1, "white");  
  
ctx.fillStyle=grd;  
ctx.fillRect(10, 10, 150, 50);
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

104

- คำสั่งสร้างงานไล่สีสำหรับนำไปใช้กับ element อื่นๆ
  - ▣ ตัวอย่างการใช้กับตัวอักษร เช่น

```
var grd=ctx.createLinearGradient(0, 0, 150, 0);  
grd.addColorStop(0, "red");  
grd.addColorStop(1, "white");  
  
ctx.fillStyle=grd;  
ctx.font="30px Arial";  
ctx.fillText("Hello World",10,50);
```



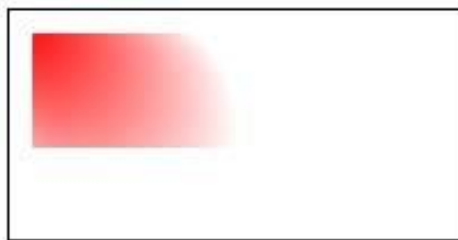


# การใช้งาน Canvas วาดภาพ 2 มิติ

105

- คำสั่งสร้างงานไล่สีสำหรับนำไปใช้กับ element อื่นๆ
  - ▣ ตัวอย่างเช่น

```
var grd=ctx.createRadialGradient(0, 0, 5, 50, 50, 50);  
grd.addColorStop(0, "red");  
grd.addColorStop(1, "white");  
  
ctx.fillStyle=grd;  
ctx.fillRect(10, 10, 150, 50);
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

106

- คำสั่งนำรูปภาพมาใช้งานร่วม

```
var image = new Image();  
image.onload = function()  
{  
    ctx.drawImage(image, x, y);  
};  
image.src = “ที่อยู่ไฟล์”;
```

# การใช้งาน Canvas วาดภาพ 2 มิติ

107

## □ คำสั่งนำรูปภาพมาใช้งานร่วม

```
1 <html>
2   <head>
3     <title>Canvas</title>
4     <script>
5       function drawImg()
6       {
7         var c = document.getElementById("myCanvas");
8         var ctx = c.getContext("2d");
9
10        var image = new Image();
11
12        image.onload = function()
13        {
14          ctx.drawImage(image,5,5);
15        };
16        image.src = "icq.jpg";
17      };
18    </script>
19  </head>
20  <body>
21    <form id="imageTest">
22      <input type="button" onClick="drawImg()"
23        value="Click me">
24    </form>
25    <canvas id="myCanvas" width="200"
26      height="100" style="border:1px solid
27        #000000;">Hello</canvas>
28  </body>
29 </html>
```

# การใช้งาน Canvas วาดภาพ 2 มิติ

108

- คำสั่งนำรูปภาพมาใช้งานร่วม
  - ▣ คำสั่งนำรูปภาพมาใช้งานและปรับขนาดภาพ

```
ctx.drawImage(image, x, y, w, h);
```

- ▣ คำสั่งนำรูปภาพมาทำเป็นพื้นหลัง (image pattern)

```
var pattern = ctx.createPattern(image,"repeat");
```

```
ctx.rect(0, 0, c.width, c.height);
```

```
ctx.fillStyle = pattern;
```

```
ctx.fill();
```

# การใช้งาน Canvas วาดภาพ 2 มิติ

109

## □ คำสั่งนำรูปภาพมาใช้งานร่วม

```
1 <html>
2   <head>
3     <title>Canvas</title>
4     <script>
5       function drawImg()
6       {
7         var c = document.getElementById("myCanvas");
8         var ctx = c.getContext("2d");
9
10        var image = new Image();
11
12        image.onload = function()
13        {
14          var pattern =
15            ctx.createPattern(image, "repeat");
16            ctx.rect(0,0,c.width,c.height);
17            ctx.fillStyle = pattern;
18            ctx.fill();
19        };
20        image.src = "bg_body.gif";
21      };
22    </script>
23  </head>
```

# การใช้งาน Canvas วาดภาพ 2 มิติ

110

- คำสั่งอื่นๆ
  - ▣ กำหนดค่าความโปร่งใสของ element

```
ctx.globalAlpha(. . .);
```

- (...)มีค่าตั้งแต่ 0-1 โดย
  - 0 คือ โปร่งใส 100%
  - 1 คือ ทึบแสง 100%

# การใช้งาน Canvas วาดภาพ 2 มิติ

111

## □ คำสั่งอื่นๆ

### ▣ ตัวอย่างเช่น

```
ctx.globalAlpha=0.2;
```

```
ctx.fillStyle="red";
```

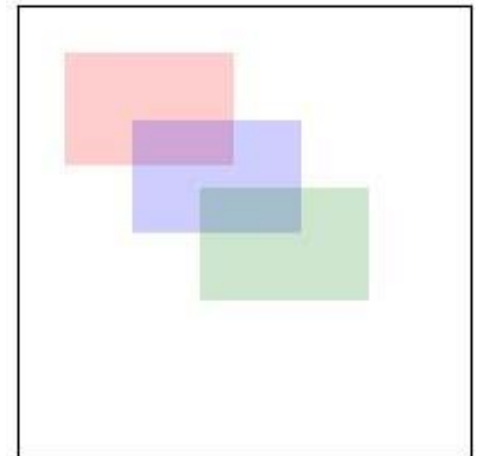
```
ctx.fillRect(20,20,75,50);
```

```
ctx.fillStyle="blue";
```

```
ctx.fillRect(50,50,75,50);
```

```
ctx.fillStyle="green";
```

```
ctx.fillRect(80,80,75,50);
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

112

## □ คำสั่งอื่นๆ

- ทุก element จะอ้างอิงตำแหน่งต่างๆ จากจุดกำเนิดที่ตำแหน่ง (0,0) แต่เราสามารถย้ายตำแหน่งจุดกำเนิดจาก (0,0) ไปที่ (x,y) ได้ โดยใช้คำสั่ง

```
ctx.translate(x,y);
```

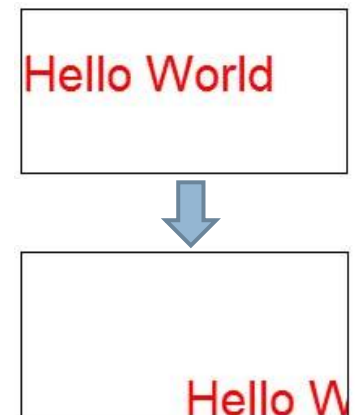
เช่น

```
ctx.translate(100,50);
```

```
ctx.font="30px Arial";
```

```
ctx.fillStyle="red";
```

```
ctx.fillText("Hello World",0,50);
```





# การใช้งาน Canvas วาดภาพ 2 มิติ

113

## □ คำสั่งอื่นๆ

- ▣ เราสามารถหมุน element รอบจุดกำเนิดได้ โดยใช้คำสั่ง

```
ctx.rotate(x,y);
```

เช่น

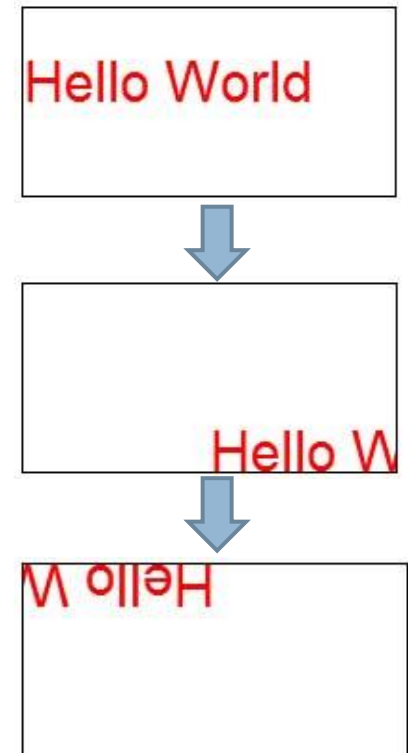
```
ctx.translate(100,50);
```

```
ctx.rotate(Math.PI);
```

```
ctx.font="30px Arial";
```

```
ctx.fillStyle="red";
```

```
ctx.fillText("Hello World",0,50);
```



# การใช้งาน Canvas วาดภาพ 2 มิติ

114

## □ คำสั่งอื่นๆ

- ▣ เราสามารถตรวจจับเหตุการณ์ที่เกิดขึ้นกับเมาส์บน canvas ได้ เช่น
  - click, dblclick : กดเมาส์หนึ่งครั้ง และกดเมาส์สองครั้ง
  - mousedown : กดเมาส์ค้างไว้
  - mouseup : ปล่อยเมาส์ที่กดไว้
  - mouseover : เลื่อนเมาส์ไปบนพื้นที่ที่กำหนดไว้ใน canvas
  - mouseout : เลื่อนเมาส์ออกจากพื้นที่ที่กำหนดไว้
  - mousemove : เคลื่อนเมาส์บนพื้นที่ที่กำหนด
  - mousewheel : ใช้ลูกกลิ้งเมาส์บนพื้นที่ที่กำหนด