



Chapter 1

ภาษา HTML

รศ.ดร. งามนิจ อางอินทร์
ผศ.ดร. ชีระยุทธ ทองเครือ

➡ HTML: Hypertext Markup Language

- ภาษาสำหรับจัดโครงสร้างของหน้าเว็บ
- คำสั่งอยู่ในรูปแบบการ **กำกับ** (Markup) ส่วนต่าง ๆ ของหน้าเว็บ เช่น ส่วนหัว ส่วนเนื้อหา ส่วนรายการ
- Web Browser สามารถแปลภาษา HTML ได้

➡ ทำไมต้องกำกับ (Markup)

กำกับให้รู้ว่าข้อความนี้คือ title

ข้อความ

```
<title>IT@KKU</title>
```

กำกับให้รู้ว่าข้อความนี้คือชื่อไฟล์ image

```

```

ข้อความ

```
<form>Full Name <input type="textbox"></form>
```

ข้อความ

กำกับให้รู้ว่าข้อความนี้ใช้
เป็นแบบฟอร์ม

➡ แท็ก (Tag)

- ใช้กำกับโครงสร้างเอกสาร
- อยู่ภายใต้เครื่องหมาย < > เช่น <body>
- รูปแบบ

<แท็กเปิด>เนื้อหา</แท็กปิด>

- ตัวอย่าง

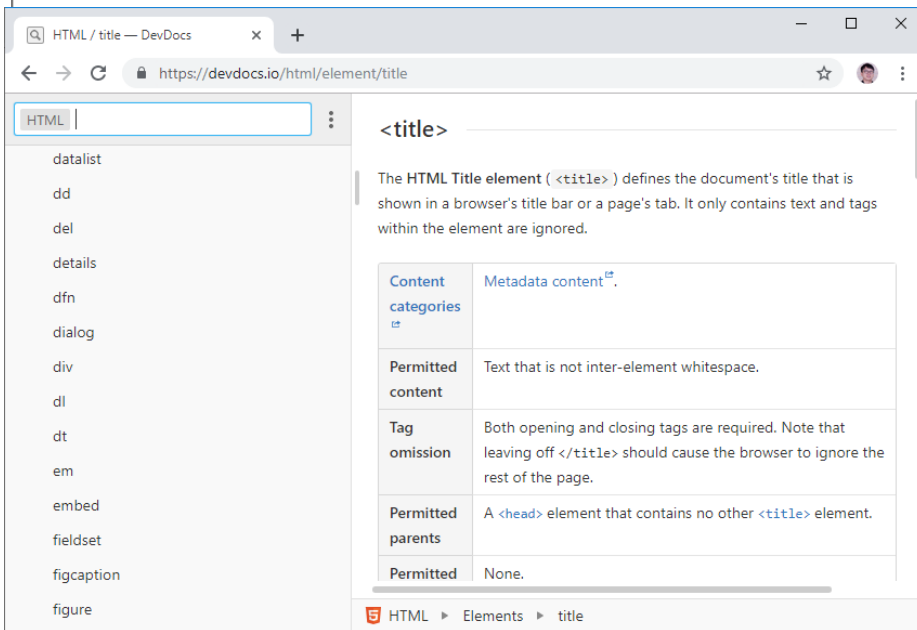
<h1>Web Development</h1>

↑
แท็กชื่อ h1 หมายถึง header ระดับ 1 ใช้กำหนดข้อความที่เป็นหัวข้อหลักใหญ่ของหน้าเว็บ

↑

➡ HTML มีแท็กอะไรบ้าง

- ดูจากคู่มือ (documentation, reference)

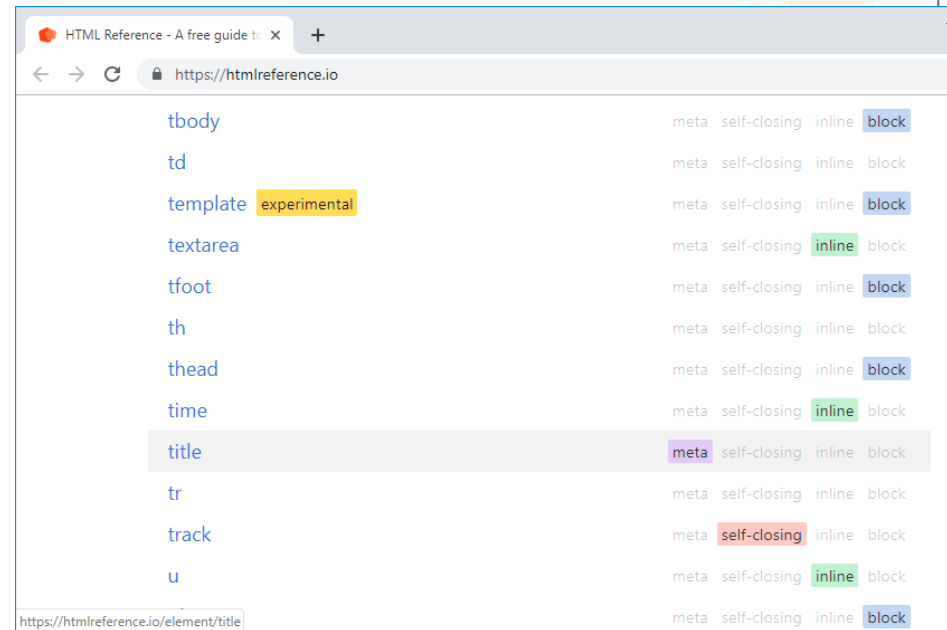


The screenshot shows the DevDocs website with the URL <https://devdocs.io/html/element/title>. The page displays the documentation for the `<title>` HTML element. On the left, there is a search bar with "HTML" entered and a list of HTML elements including `datalist`, `dd`, `del`, `details`, `dfn`, `dialog`, `div`, `dl`, `dt`, `em`, `embed`, `fieldset`, `figcaption`, and `figure`. The main content area shows the `<title>` element documentation, including a description: "The HTML Title element (`<title>`) defines the document's title that is shown in a browser's title bar or a page's tab. It only contains text and tags within the element are ignored." Below this is a table with the following content:

Content categories	Metadata content
Permitted content	Text that is not inter-element whitespace.
Tag omission	Both opening and closing tags are required. Note that leaving off <code></title></code> should cause the browser to ignore the rest of the page.
Permitted parents	A <code><head></code> element that contains no other <code><title></code> element.
Permitted	None.

At the bottom, there is a breadcrumb trail: `HTML` ▶ `Elements` ▶ `title`.

<https://devdocs.io>



The screenshot shows the HTML Reference website with the URL <https://htmlreference.io>. The page displays a list of HTML elements and their categories. The elements listed are: `tbody`, `td`, `template` (marked as experimental), `textarea`, `tfoot`, `th`, `thead`, `time`, `title` (highlighted), `tr`, `track`, and `u`. To the right of each element, its categories are listed: `meta`, `self-closing`, `inline`, and `block`. The `title` element is highlighted in a grey box, and its categories are `meta`, `self-closing`, `inline`, and `block`.

<https://htmlreference.io>

➡ ตัวอย่างแท็ก

<html>

<head>

<body>

<title>

<h1> ถึง <h6>

<form>

<i>

<div>

<table>

<a>

<input>

<select>

<option>

➡ ประเภทของแท็ก

- แท็กที่ต้องมีแท็กเปิดและแท็กปิดเสมอ

`<h1>Web Development</h1>`

- แท็กที่ไม่ต้องมีแท็กปิด (เรียกว่า Empty Tag)

``

Responsive Design`
` by CSS

↑
คำสั่งกำหนดให้ปิดบรรทัดใหม่

➡ Empty Tag ในภาษา HTML

<area>

<base>

<hr>

<input>

<link>

<meta>

<param>

<source>

<track>

<wbr>

➡ HTML เป็น case-insensitive

- คำสั่งในภาษา HTML จะใช้ตัวพิมพ์เล็ก หรือตัวพิมพ์ใหญ่ ก็ได้

`
` หรือ `
` หรือ `
`

- ใช้ชื่อ Tag เป็นตัวพิมพ์เล็กทั้งหมด ช่วยให้อ่านง่าย

➡ แอตทริบิวต์ (Attribute)

- ส่วนที่แทรกภายในแท็ก
- ใช้อธิบายข้อมูลเพิ่มเติมให้กับ Tag
- รูปแบบ

<tag ชื่อแอตทริบิวต์="ค่า">

- ตัวอย่าง

Attribute Name

Attribute Value

*หมายเหตุ บาง attribute อาจระบุได้เช่น controls ในแท็ก <video>

➡ อีลิเมนต์ (Element)

- Element ใช้เรียกรวม ส่วนที่เป็น Tag, Attribute และเนื้อหา

```
<div id="p1" class="red-alert">
```

```
    Web Programming
```

```
</div>
```

} 1 Element

➡ เครื่องมือสำหรับการพัฒนาเว็บ

- Text Editor
 - Notepad, Visual Studio Code, Sublime, Atom
- Web browser ใช้ทดสอบแอปพลิเคชันที่สร้างขึ้น
 - Chrome, Edge, Firefox, Safari, IE
- คู่มือ (Document/Reference) เช่น
 - <https://devdocs.io>
 - <https://htmlreference.io>

➡ โครงสร้างของเว็บ 1 หน้า



กำหนดข้อมูลที่เกี่ยวข้องกับเว็บ
จะทำงานก่อนการแสดงผลเสมอ

ส่วนแสดงผลหน้าเว็บทั้งหมด

➡ โครงสร้างของเว็บ 1 หน้าที่ต้องมีเสมอ

```
<!DOCTYPE html>
```

ระบุชนิดของไฟล์ หรือเอกสาร

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Title of page</title>
```

```
</head>
```

```
<body>
```

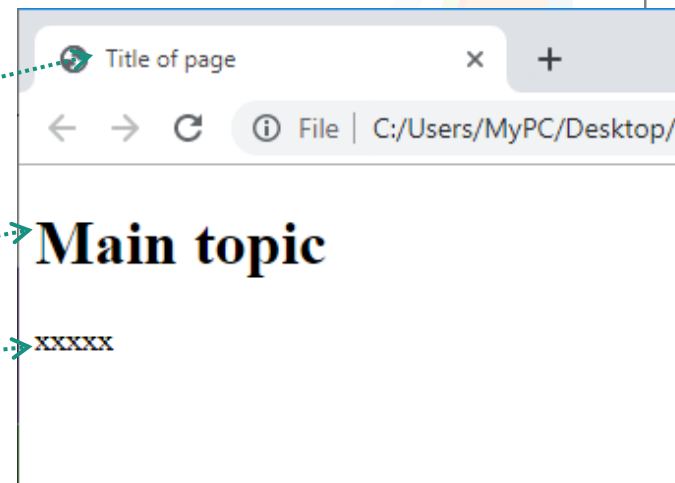
```
  <!-- Content -->
```

```
  <h1>Main topic</h1>
```

```
  xxxxx
```

```
</body>
```

```
</html>
```



Browser จะแปล และ
แสดงผลตามลำดับคำสั่ง

<head> Element

- <head> คือ Element ที่บรรจุ
 - โค้ดภาษา CSS หรือ JavaScript
 - คำสั่งเชื่อมโยงไปยังไฟล์ภายนอก เช่น template, library
 - ระบุข้อมูลเกี่ยวกับเว็บ (Meta Information)
- แท็กที่อยู่ภายใต้ <head> ได้แก่ <title>, <style>, <meta>, <link>, <script>

<meta> Element

- Metadata คือ ข้อมูลที่อธิบายข้อมูล (Information about data)
- แท็ก <meta> จะไม่แสดงผลใดๆ ให้คนเห็น
- ใช้เขียนคำอธิบาย, กำหนด keyword, กำหนดชื่อผู้เขียนเว็บ
- Metadata จะถูกใช้โดย browser หรือ Search Engines
- แท็ก <meta> จะอยู่ในส่วน <head> เท่านั้น

➡ ตัวอย่างการใช้ <meta>

- กำหนด Keyword

```
<meta name="keywords" content="web, programming, html, css">
```

- กำหนดคำอธิบาย

```
<meta name="description" content="เว็บเพื่อการเรียนรู้และพัฒนา ...">
```

- กำหนดชุดอักขระ ป้องกันภาษาต่างดาว

```
<meta charset="UTF-8">
```

- กำหนดให้มีการรีโหลดหน้าใหม่ทุกๆ วินาที

```
<meta http-equiv="refresh" content="30">
```

➡ ประโยชน์ของการใช้แท็ก <meta>

```
3
4 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
  transitional.dtd">
5 <html xmlns="http://www.w3.org/1999/xhtml">
6
7 <head>
8 <meta http-equiv="Content-Type" content="text/html;
  charset=UTF-8" />
9 <meta http-equiv="CACHE-CONTROL" content="Public" />
10
11
12 <meta name="description" content="Thai
  Airways International - online reservations and airline
  ticket purchase, flight schedule, and packages." />
13
14
15 <meta name="key
  content="tg, thai, air, thailand, k
  e, airline, airway" />
```

www.google.co.th/#scient=psy-ab&q=thai%20airways&oq=&

+You Search Images Maps Play Gmail Drive Calendar Translate Photos

Google thai airways

Web Images Maps More Search tools

About 13,400,000 results (0.17 seconds)

Thai Airways
www.thaiairways.com/ ▾
Thai Airways International - online reservations and airline ticket purchase, flight schedule, and packages.

<title> Element

- ใช้แท็ก <title> ในการนิยามชื่อ Tab บน browser
- ข้อความที่อยู่ภายในแท็ก title จะถูกใช้เป็น
 - ชื่อเรียกบน Bookmark
 - หัวข้อในผลลัพธ์ของ Search Engine

Comment

- การใส่คำอธิบายให้กับส่วนต่างๆของเอกสาร จะช่วยให้ง่ายต่อความเข้าใจ มีรูปแบบดังนี้

`<!-- This is a comment -->`

- สามารถใส่ได้ในทุกส่วนของเอกสาร HTML
- Browser จะไม่นำส่วนนี้ไปแสดงผล

➡ หัวข้อ

- การกำหนดหัวข้อจะใช้ <h1> ถึง <h6>

```
1 <!doctype html>
2 <html>
3   <body>
4     <h1>This is a heading</h1>
5     <h2>This is a heading</h2>
6     <h3>This is a heading</h3>
7   </body>
8 </html>|
```

This is a heading

This is a heading

This is a heading

- ใช้ Heading ในการกำหนดโครงสร้างหัวข้อภายในเว็บตามลำดับความสำคัญ <h1> สำคัญที่สุด <h6> น้อยที่สุด
- ไม่ควรใช้เพื่อขยายขนาดตัวอักษร เพราะ Search Engine จะใช้แท็ก Heading ในการทำดัชนี (Indexing)

➡ Paragraph

- ย่อหน้านิยามด้วย <p>

```
1 <!doctype html>
2 <html>
3   <body>
4     <p>This is a paragraph</p>
5     <p>This is another paragraph</p>
6   </body>
7 </html>
```

This is a paragraph

This is another paragraph

➡ Line Break

- ใช้แท็ก `
` ในการขึ้นบรรทัดใหม่

```
1 <!doctype html>
2 <html>
3   <body>
4     <p>This is<br>a para<br>graph with line breaks</p>
5   </body>
6 </html>
```

This is
a para
graph with line breaks

➡ รูปแบบตัวอักษร

- ตัวหนา ใช้ หรือ
- ตัวเอียง ใช้ <i> หรือ
- ตัวห้อย ใช้ <sub>, ยกกำลัง ใช้ <sup>

```
1 <!doctype html>
2 <html>
3   <body>
4     <p><b>This text is bold</b></p>
5     <p><i>This text is italic</i></p>
6     <p>This is<sub> subscript</sub> and
7       <sup>superscript</sup></p>
8   </body>
9 </html>
```

This text is bold

This text is italic

This is subscript and superscript

➡ Character Entity

- Character Entity คือ คำสั่งที่ใช้ในการแสดงอักขระที่ไม่มีในแป้นพิมพ์ หรืออักขระที่ถูกนำไปใช้เป็นคำสั่งใน HTML แล้ว เช่น
< >
- การแสดงอักขระเหล่านี้บนหน้าเว็บ จะใช้ Character Entity ซึ่งมีรูปแบบ ดังนี้

&entity_name; หรือ &#entity_number;

เช่น < หรือ < แทนเครื่องหมาย <

© หรือ © แทนเครื่องหมาย ©

➡ ช่องว่างในเอกสาร HTML

- การใช้ Space, Tab, Enter ในโค้ด HTML จะไม่ทำให้แสดงผลบน Browser เป็นไปตามนั้น เช่น

บนเอกสารใช้

Hello World

ผลลัพธ์บน Browser

Hello World

- หากต้องการใช้ Enter จะต้องใช้คำสั่ง
 เช่น

บนเอกสารใช้

Hello
 World

ผลลัพธ์บน Browser

Hello

World

- หากต้องการใช้ Space จะต้องใช้คำสั่ง เช่น

บนเอกสารใช้

Hello World

ผลลัพธ์บน Browser

Hello World

➡ การบังคับให้แสดงช่องว่างตามที่กำหนด

- `<pre>....</pre>` ย่อมาจาก Preformat Text
- ใช้กำหนดข้อความที่อยู่ระหว่างแท็ก `<pre>` มีการแสดงผลบน browser ตามรูปแบบที่เขียนอยู่ในแท็ก `<pre>` ทั้งการขึ้นบรรทัดใหม่และการกำหนดช่องว่างระหว่างข้อความ

```
1 <!doctype html>
2 <html>
3 <body>
4 <p>The pre tag is good for displaying computer code:</p>
5 <pre>
6 for i = 1 to 10
7     print i
8 next i
9 </pre>
10 </body>
11 </html>
```

The pre tag is good for displaying computer code:

```
for i = 1 to 10
    print i
next i
```

➡ รูปภาพ

- รูปแบบ

``

- ตัวอย่าง

``

ระบุชื่อไฟล์ภาพ หรือ อ้างอิงเป็น URL

ระบุความกว้างและความยาว

***หมายเหตุ**

- ความกว้างและความยาวของภาพ สามารถกำหนดเพียงค่าใดค่าหนึ่งก็ได้ หากไม่ทราบสัดส่วนของภาพ เพราะเมื่อกำหนด width อย่างเดียว browser จะคำนวณ height ที่ เป็นค่าที่คงไว้ซึ่งสัดส่วนเดิมอัตโนมัติ
- นอกจากนี้ การย่อภาพในโปรแกรมตกแต่งภาพให้ตรงตามขนาดที่ต้องการเลย จะช่วยลดขนาดไฟล์ ช่วยให้โหลดได้เร็ว

➡ ชนิดของไฟล์ภาพสำหรับเว็บ

- **.gif** สามารถใช้สีได้สูงสุด 256 สี ภาพที่มีจำนวนสีน้อยขนาดไฟล์จะยิ่งเล็กลงไปด้วย เหมาะกับ การ์ตูน โลโก้ต่างๆ ที่มีสีไม่มากนัก และเหมาะกับการทำ Animation
- **.png** ปรับปรุงข้อจำกัดของไฟล์ .gif ในหลายด้าน เช่น เพิ่มจำนวนสี ลดขนาดไฟล์
- **.jpg** หรือชนิด Jpeg สามารถแสดงสีได้สูงสุด 16.7 ล้านสี มีการบีบอัดเพื่อลดขนาดภาพ เหมาะกับภาพถ่ายทั่วไป

➡ แหล่งรวมไฟล์ภาพ

- <https://www.pexels.com>
- <https://stocksnap.io>
- <https://unsplash.com>
- <http://www.gratisography.com>

➡ วิดีโอ

- ไฟล์วิดีโอที่สามารถทำงานบน browser ได้แก่ MP4, WebM และ OGV

- การแทรกวิดีโอบนหน้าเว็บ

```
<video src="podcast.mp4" width="320" height="240" controls></video>
```



Attribute ของ <video>

ชื่อ Attribute	ค่า Attribute	คำอธิบาย
src	<i>URL</i>	ระบุตำแหน่งของไฟล์วิดีโอ
autoplay		กำหนดให้วิดีโอเล่นเมื่อเปิดหน้าเว็บ
controls		แสดงปุ่มควบคุมการเล่นวิดีโอ
height	<i>pixels</i>	กำหนดความสูงของวิดีโอ
width	<i>pixels</i>	กำหนดความกว้างของวิดีโอ
loop		กำหนดให้เล่นซ้ำเมื่อเล่นวิดีโอจบแล้ว
muted		กำหนดให้ปิดเสียง
poster	<i>URL</i>	ระบุตำแหน่งของภาพที่จะนำมาแสดงขณะที่กำลังโหลดวิดีโอ หรือจนกระทั่งผู้ใช้คลิกปุ่ม Play
preload	auto, metadata, none	กำหนดการเริ่มต้นโหลดวิดีโอ auto - ให้ดาวน์โหลดวิดีโอทันทีที่มีการเข้าหน้าเว็บ metadata - ดาวน์โหลดเฉพาะข้อมูลเกี่ยวกับวิดีโอ เช่น ระยะเวลา เฟรมแรก none - ไม่ต้องดาวน์โหลด จนกว่าผู้ใช้จะเริ่มดู

➡ การกำหนดวิดีโอทางเลือกสำรอง

- การรองรับชนิดวิดีโอของแต่ละ Browser ไม่เหมือนกัน ดังนั้นจึงควรใช้แท็ก `<source>` เพื่อกำหนดวิดีโอชนิดต่างๆ โดยเมื่อเปิดด้วย Browser ใดที่ไม่รองรับจะค้นหาวิดีโอในแท็ก `<source>` ตามลำดับ

```
<video controls>
```

```
<source src="podcast.mp4" type="video/mp4">
```

```
<source src="podcast.webm" type="video/webm">
```

```
<source src="podcast.ogv" type="video/ogg">
```

```
</video>
```

➡ แหล่งรวมไฟล์วิดีโอ

- <https://videos.pexels.com>
- <https://pixabay.com>

➡ เสียง

- ไฟล์เสียงที่สามารถทำงานบน browser ได้แก่ MP3, WAV และ OGG
- การแทรกเสียงบนหน้าเว็บ

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

Attribute ของ <audio>

ชื่อ Attribute	ค่า Attribute	คำอธิบาย
src	URL	ระบุตำแหน่งของไฟล์เสียง
autoplay		กำหนดให้เล่นเสียงเมื่อเปิดหน้าเว็บ
controls		แสดงปุ่มควบคุมการเล่นเสียง
loop		กำหนดให้เล่นซ้ำเมื่อเล่นจบแล้ว
muted		กำหนดให้ปิดเสียง
preload	auto, metadata, none	กำหนดการเริ่มต้นโหลดเสียง auto - ให้ดาวน์โหลดทันทีที่มีการเข้าหน้าเว็บ metadata - ดาวน์โหลดเฉพาะข้อมูลเกี่ยวกับเสียง เช่น ระยะเวลา none - ไม่ต้องดาวน์โหลด จนกว่าผู้ใช้จะเริ่มดู

➡ การกำหนดไฟล์เสียงทางเลือกสำรอง

```
<audio controls>
```

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

```
  <source src="song.ogg" type="audio/ogg">
```

```
</audio>
```

<iframe> Element

- การนำเว็บเพจอื่นมาแสดงซ้อนในเว็บเพจของตนเองจะใช้แท็ก <iframe>
- กำหนดความกว้างและสูงด้วย Attribute width และ height
- ตัวอย่าง

```
<iframe src="https://www.kku.ac.th" width="500" height="200"
frameborder="0"></iframe>
```

*หมายเหตุ บางเว็บไซต์อาจไม่อนุญาตให้เว็บของตนเองไปใช้ในแท็ก <iframe> ได้ เช่น google.com

➡ การแทรกคลิปจาก YouTube

- ตัวอย่าง

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/MyWyvUje2Mw"
frameborder="0" allowfullscreen></iframe>
```

ใช้ URL จาก YouTube ที่ต้องการนำมาแสดง

อนุญาตให้ดูแบบเต็มจอได้

Hyperlinks (Links)

- Hyperlinks คือ จุดหนึ่งบนเว็บที่สามารถคลิก และเปิดเว็บในหน้าที่ต้องการเชื่อมโยง
- ใช้แท็ก <a> ในการกำกับส่วนที่ต้องการให้ผู้ใช้งานคลิกได้
- แท็ก <a>.... จะครอบข้อความ หรือรูปภาพก็ได้
- ส่วนที่ครอบด้วยแท็ก <a> ตัวชี้จะเปลี่ยนจากลูกศรเป็นรูปมือแทน

Hyperlinks (Links)

- រូបបែប

`Link text`



- ตัวอย่าง

[Computer Science KKU](http://www.cs.kku.ac.th)

[More...](detail.html)

`My Photo`

[Download](gallery.zip)

ผลลัพั

Computer Science KKU

➡ Hyperlinks (Links)

- กำหนดวิธีการเปิด Link ด้วย Attribute target

```
<a href="http://www.cs.kku.ac.th" target="_blank">CS KKU</a>
```

- `_blank` คือ การเปิด Link ที่ระบุบน Tab ใหม่
- กรณีที่ไม่มีการกำหนด Attribute target จะเปิดที่หน้าเดิม

➡ การสร้าง Link ด้วยรูปภาพ

- ใช้แท็ก ซ่อนภายใต้แท็ก <a>

```
<a href="http://www.google.com"></a>
```

➡ ชนิดของ URL

- **Absolute URL** - URL แบบเต็ม ใช้ในการอ้างอิงไปยังโฟลเดอร์และไฟล์ที่อยู่บน server กับหน้าเว็บ

<http://www.cs.kku.ac.th/comsc/index.php>

- **Relative URL** - URL ที่อ้างอิงโดยใช้ตำแหน่งปัจจุบัน เพื่อเชื่อมโยงไปยังโฟลเดอร์และไฟล์ที่อยู่ใน server เดียวกันกับหน้าเว็บ

</comsc/index.php>

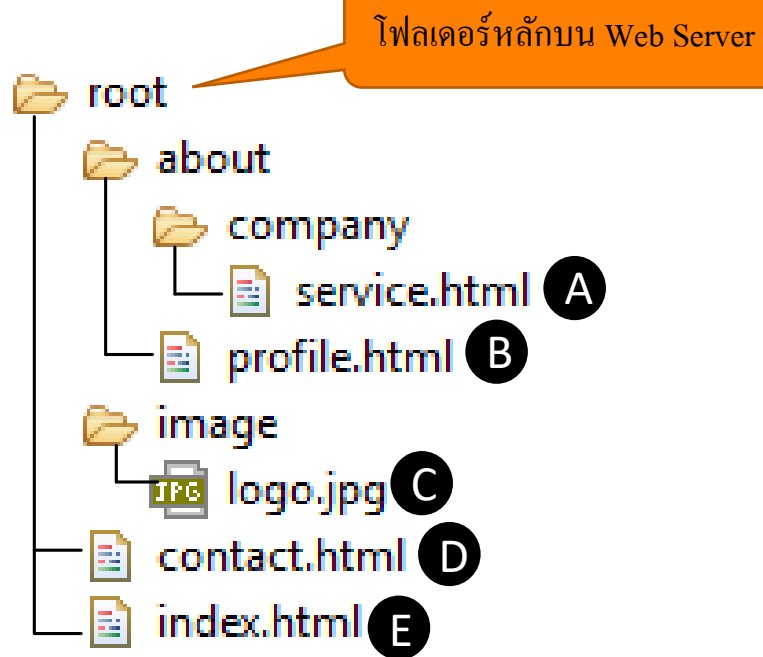
➡ การอ้างอิงไฟล์ด้วย Relative URL

- Relative URL สามารถใช้ได้กับทุก attribute ที่อ้างอิงไฟล์
 - รูปภาพ ``
 - Link ``
 - วิดีโอ `<video src="[relative url]">`
- ประโยชน์ของ Relative URL
 - สั้น เพราะไม่ต้องอ้างอิง path ทั้งหมด
 - เมื่อมีการเปลี่ยน domain ไม่ต้องตามไปแก้ไขทั้งหมด
 - ทดสอบการทำงานได้แม้ไม่ได้อยู่บนอินเทอร์เน็ต

➡ สัญลักษณ์สำหรับใช้กับ Relative URL

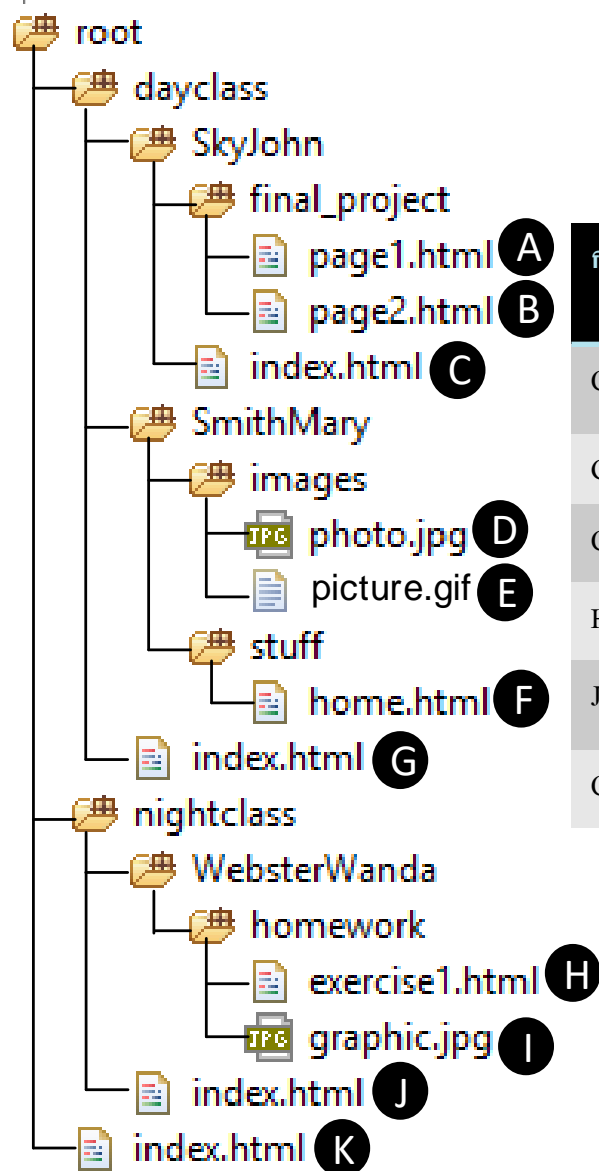
ขึ้นต้นด้วย	ความหมาย	คำอธิบาย	ตัวอย่าง
./	โฟลเดอร์ปัจจุบัน	อ้างอิงไฟล์หรือโฟลเดอร์ปัจจุบัน <u>หรือ สามารถละสัญลักษณ์นี้ แล้วระบุชื่อโฟลเดอร์ลูกที่ต้องการเข้าถึงได้เลย</u>	./image/dog.jpg หรือ image/dog.jpg
../	โฟลเดอร์แม่	อ้างอิงไฟล์ที่อยู่เหนือโฟลเดอร์ปัจจุบัน 1 ระดับ	../project/chart.png
../..	โฟลเดอร์เหนือโฟลเดอร์แม่	อ้างอิงไฟล์ที่อยู่เหนือโฟลเดอร์ปัจจุบัน 2 ระดับ	../..page/content.pdf
/	Root Folder	อ้างอิงโฟลเดอร์ระดับบนสุด หรือเป็นตัวแทนของชื่อ domain	/index.html

➡ ตัวอย่าง



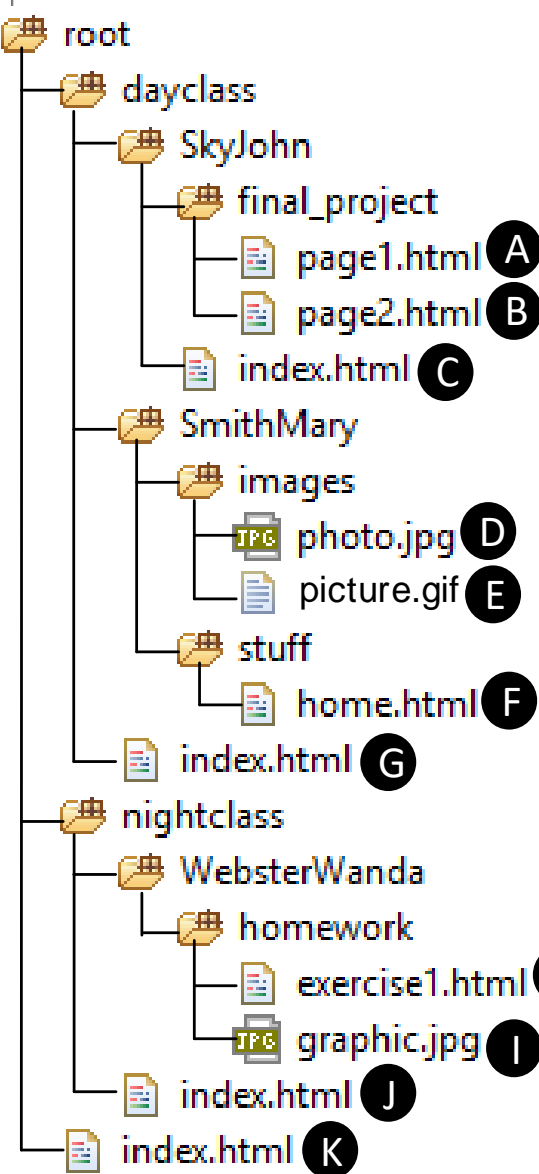
การอ้างอิง	Absolute URL (สมมติว่าเว็บไซต์อยู่ที่ http://www.aa.com)	Relative URL
ที่หน้าเว็บ E สร้างลิงค์ไปยัง D	<code></code>	<code></code>
ที่หน้าเว็บ E ดึงภาพ C มาแสดง	<code></code>	<code></code>
ที่หน้าเว็บ A สร้างลิงค์ไปยัง B	<code></code>	<code></code>
ที่หน้าเว็บ A สร้างลิงค์ไปยัง D	<code></code>	<code></code>
ที่หน้าเว็บ B ดึงภาพ C มาแสดง	<code></code>	<code></code>

กิจกรรม



การอ้างอิง	Absolute URL (สมมติว่าเว็บไซต์อยู่ที่ http://www.aa.com)	Relative URL
C ลิงค์ไปยัง A	http://www.aa.com/dayclass/SkyJohn/final_project/page1.html	final_project/page1.html
G ลิงค์ไปยัง A		SkyJohn/final_project/page1.html
C ลิงค์ไปยัง G		../index.html
H ลิงค์ไปยัง J		../../index.html
J ดึงภาพ E		../dayclass/SmithMary/images/picture.jpg
C ดึงภาพ E		../SmithMary/images/photo.jpeg

➔ กิจกรรม (ตรวจคำตอบจากการทำแล็บ)



การอ้างอิง	Absolute URL (สมมติว่าเว็บไซต์อยู่ที่ http://www.aa.com)	Relative URL
G ลิงค์ไปยังหน้า F		
F แสดงภาพ D บนเว็บ		
H ลิงค์ไปยังหน้า B		
K แสดงภาพ E บนเว็บ		
H ลิงค์ไปยังหน้า K		

➡ การอ้างไฟล์ที่ไม่ควรทำเป็นอย่างยิ่ง

``

`XXXX`

ปัญหาจะเกิดขึ้น หากมีการย้ายที่เก็บไฟล์ เพราะต้องแก้ไข source code

➡ Unordered List

- Unordered List คือ List ที่ไม่มีการกำหนดลำดับ จะครอบด้วยแท็ก `` โดยอธิบายแต่ละรายการด้วยแท็ก ``

```
1 <!doctype html>
2 <html>
3   <body>
4     <ul>
5       <li>Coffee</li>
6       <li>Milk</li>
7     </ul>
8   </body>
9 </html>
```

- Coffee
- Milk

➡ Ordered List

- Ordered List คือ List ที่มีการกำหนดลำดับ จะครอบด้วยแท็ก `` โดยอธิบายแต่ละรายการด้วยแท็ก ``

```
1 <!doctype html>
2 <html>
3   <body>
4     <ol>
5       <li>Coffee</li>
6       <li>Milk</li>
7     </ol>
8   </body>
9 </html>
10 |
```

1. Coffee
2. Milk

➡ Description List

- Description List คือ List จำกัดความ มักถูกนำมาใช้ในการ จำกัดความ เช่น ในพจนานุกรม รูปแบบ List แบ่งออกเป็น 2 ส่วน ส่วนแรก คือ คำศัพท์ (Term) ใช้ <dt> ส่วนที่สองคือคำ จำกัดความ ใช้ <dd> โดยทั้งหมดจะครอบด้วยแท็ก <dl>

```
1 <!doctype html>
2 <html>
3   <body>
4     <dl>
5       <dt>Coffee</dt>
6       <dd>black hot drink</dd>
7       <dt>Milk</dt>
8       <dd>white cold drink</dd>
9     </dl>
10   </body>
11 </html>
```

Coffee	black hot drink
Milk	white cold drink

➔ Description List

```
<html>
<body>
<dl>
  <dt>ต้นจำปี</dt>
  <dd>จำปี (ชื่อวิทยาศาสตร์: Michella alba DC.) มีมากกว่า 50 ชนิด</dd>
  <dt>ลักษณะทางพฤกษศาสตร์</dt>
  <dd>เป็นไม้ยืนต้นขนาดกลาง สูงประมาณ 10-20 เมตร ไม่ผลัดใบ</dd>
</dl>
</body>
</html>
```

ต้นจำปี
จำปี (ชื่อวิทยาศาสตร์: Michella alba DC.) มีมากกว่า 50 ชนิด
ลักษณะทางพฤกษศาสตร์
เป็นไม้ยืนต้นขนาดกลาง สูงประมาณ 10-20 เมตร ไม่ผลัดใบ

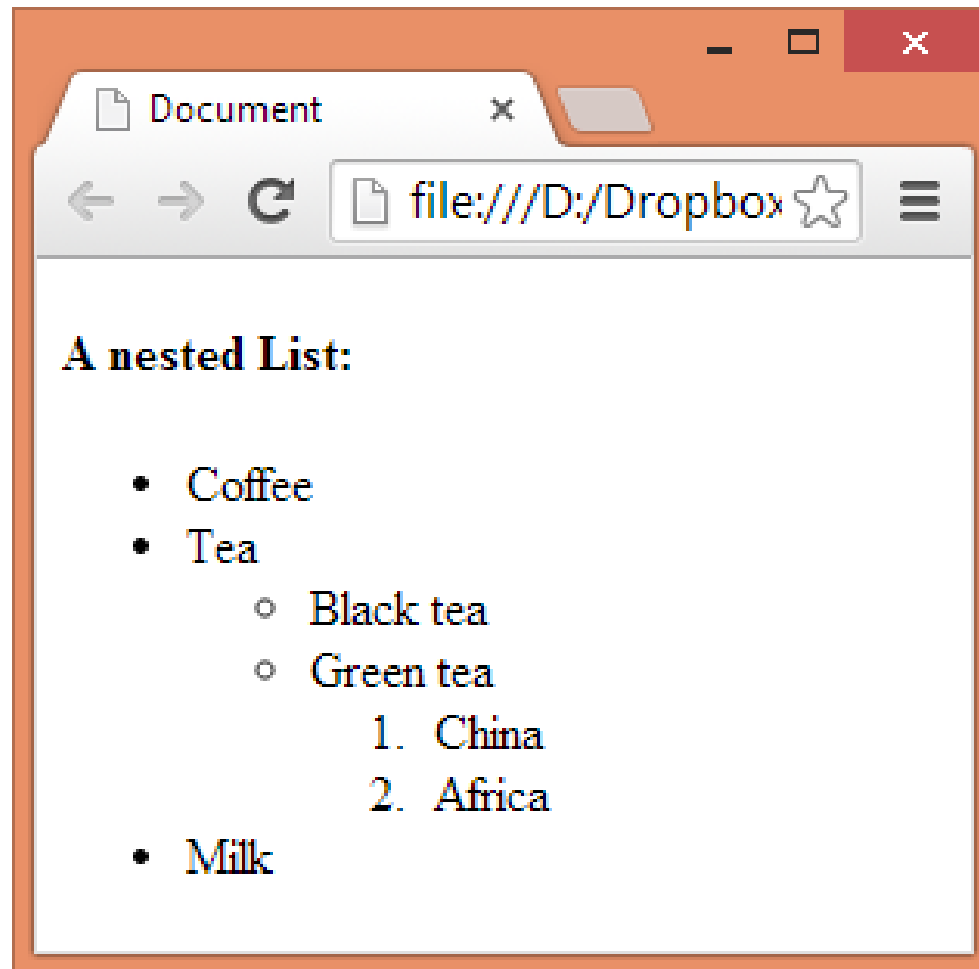
➡ List ซ่อน List

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <ol>
5     <li>Library Collections</li>
6     <ul>
7         <li> Books </li>
8         <li> Journals </li>
9     </ul>
10    <li> Library Catalog </li>
11    <li> Electronic Resources </li>
12    <ul>
13        <li> CD-ROMs </li>
14        <li> Abstracts & Indexes </li>
15    </ul>
16 </ol>
```

1. Library Collections
 - Books
 - Journals
2. Library Catalog
3. Electronic Resources
 - CD-ROMs
 - Abstracts & Indexes

➡ กิจกรรม

- จงสร้างรายการที่มีรูปแบบดังนี้



ตาราง

- ใช้ `<table>` ในการกำหนดตาราง
- ตารางจะแบ่งออกเป็น แถว (แท็ก `<tr>`) ในแถวจะแบ่งเป็น คอลัมน์ (แท็ก `<td>`)
- แท็ก `<td>` คือ ข้อมูล 1 ช่อง
- ข้อมูลใน `<td>` จะเป็นตัวอักษร, ภาพ, Link, ฟอর্মก็ได้

➡ ตัวอย่างตาราง

- ถ้าไม่มีการกำหนด Attribute border ตารางจะไม่มีกรอบ
- Attribute border ถูกยกเลิกไปแล้ว ใน HTML version 5

`<table border="1">`

`<tr>`

`<td>row 1, cell 1</td>`

`<td>row 1, cell 2</td>`

`</tr>`

`<tr>`

`<td>row 2, cell 1</td>`

`<td>row 2, cell 2</td>`

`</tr>`

`</table>`

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

➡ การกำหนดหัวตาราง

- ใช้แท็ก <th> ในการกำหนดแถวที่เป็นหัวของตาราง
- Browser จะแสดงข้อมูลในแท็ก <th> เป็นตัวหนา และอยู่กึ่งกลางเซลล์

```
<table border="1">  
<tr>  
  <th>Header 1</th>  
  <th>Header 2</th>  
</tr>  
<tr>  
  <td>row 1, cell 1</td>  
  <td>row 1, cell 2</td>  
</tr>  
<tr>  
  <td>row 2, cell 1</td>  
  <td>row 2, cell 2</td>  
</tr>  
</table>
```

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

➡ การผสานเซลล์ในตาราง

- การผสานในแนวนอนใช้ Attribute colspan
- การผสานในแนวตั้งใช้ Attribute rowspan

```
<table border="1">
<tr>
  <th>Name</th>
  <th colspan="2">Telephone</th>
</tr>
<tr>
  <td>Bill Gates</td>
  <td>555 77 854</td>
  <td>555 77 855</td>
</tr>
</table>
```

Name	Telephone	
Bill Gates	555 77 854	555 77 855

```
<table border="1">
<tr>
  <th>First Name:</th>
  <td>Bill Gates</td>
</tr>
<tr>
  <th rowspan="2">Telephone:</th>
  <td>555 77 854</td>
</tr>
<tr>
  <td>555 77 855</td>
</tr>
</table>
```

First Name:	Bill Gates
Telephone:	555 77 854
	555 77 855

➡ กิจกรรม

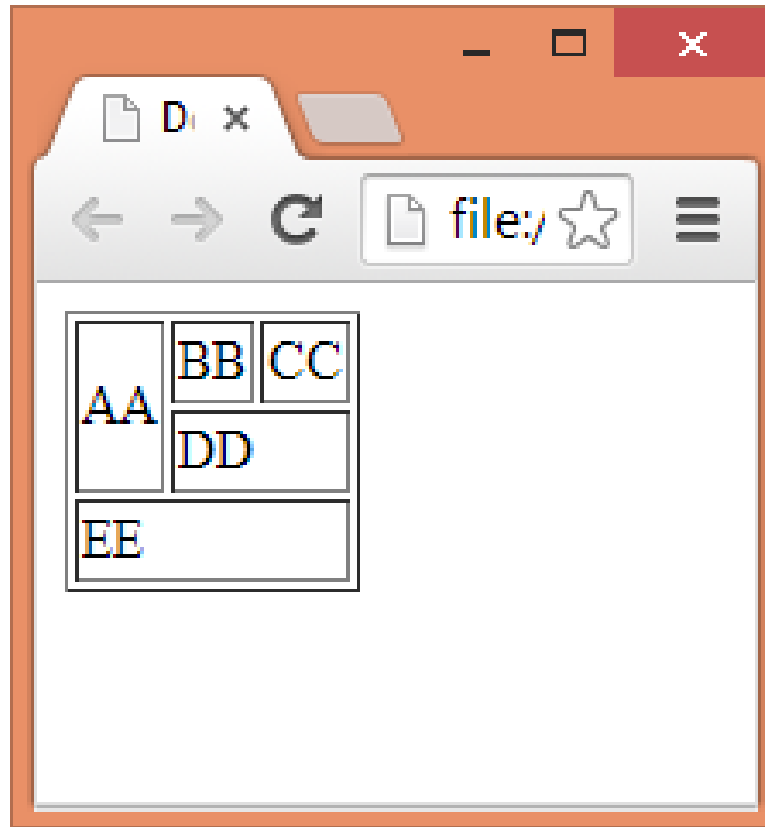
- จงสร้างตารางที่มีรูปแบบดังนี้

Users Info	1	John Carter	johncarter@mail.com
	2	Peter Parker	peterparker@mail.com
	3	John Rambo	johnrambo@mail.com

Users Info		
1	John Carter	johncarter@mail.com
2	Peter Parker	peterparker@mail.com
3	John Rambo	johnrambo@mail.com

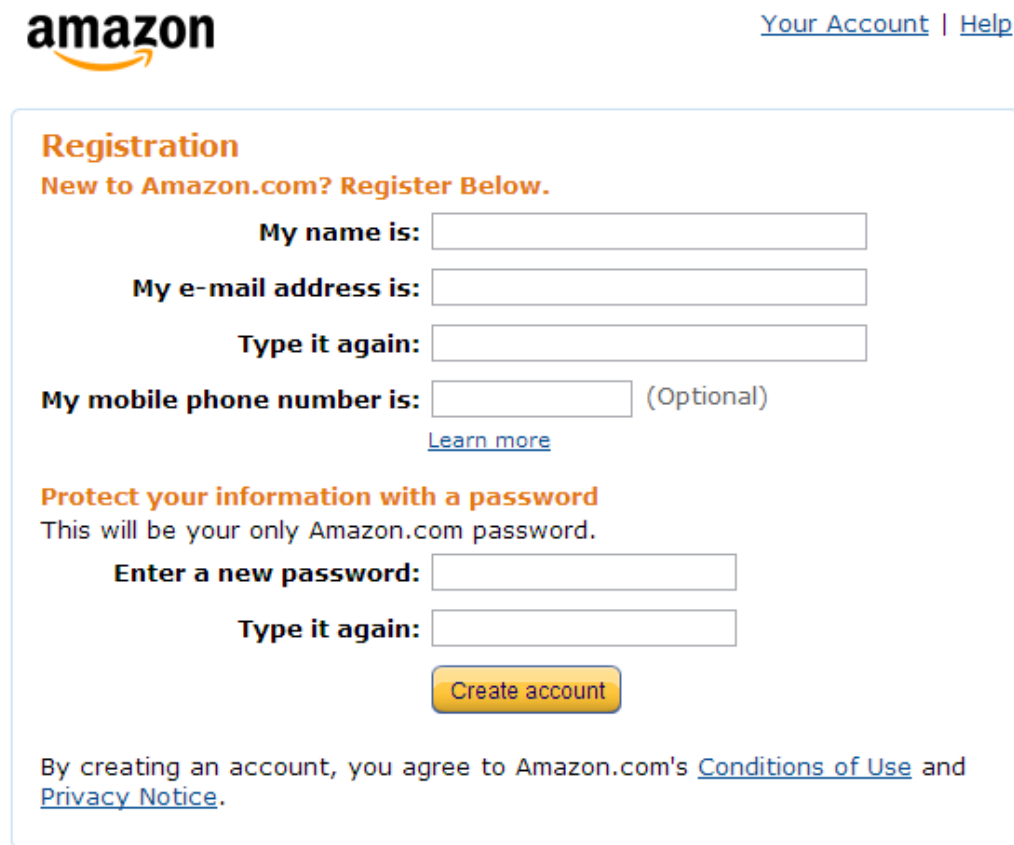
➡ กิจกรรม

- จงสร้างตารางที่มีรูปแบบดังนี้



➡ แบบฟอร์มบนเว็บ

- แบบฟอร์ม คือ การรับข้อมูลจากผู้ใช้งานผ่านเว็บไซต์ เพื่อส่งไปยัง Web Server



The image shows a screenshot of the Amazon registration page. At the top left is the Amazon logo, and at the top right are links for "Your Account" and "Help". The main content area is titled "Registration" and "New to Amazon.com? Register Below." It contains several input fields: "My name is:", "My e-mail address is:", "Type it again:", "My mobile phone number is:" (with an "Optional" label), and a "Learn more" link. Below these is a section titled "Protect your information with a password" with the text "This will be your only Amazon.com password." It includes two password input fields labeled "Enter a new password:" and "Type it again:", followed by a "Create account" button. At the bottom, it states "By creating an account, you agree to Amazon.com's Conditions of Use and Privacy Notice."

amazon [Your Account](#) | [Help](#)

Registration
New to Amazon.com? Register Below.

My name is:

My e-mail address is:

Type it again:

My mobile phone number is: (Optional)
[Learn more](#)

Protect your information with a password
This will be your only Amazon.com password.

Enter a new password:

Type it again:

By creating an account, you agree to Amazon.com's [Conditions of Use](#) and [Privacy Notice](#).

➡ HTML Form

- เว็บเพจแต่ละหน้าจะมีก็ฟอร์มก็ได้ แต่ในแต่ละฟอร์มจะต้องครอบด้วย `<form>` แยกส่วนกัน

`<form>`

`<!-- ใส่ input ต่างๆ -->`

`</form>`

➡ Text Field

- `<input type="text">` ใช้ในการรับ input ข้อความแบบ
บรรทัดเดียว

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       First name: <input type="text" name="firstname"><br>
6       Last name: <input type="text" name="lastname">
7     </form>
8   </body>
9 </html>
```

First name:

Last name:

➡ Submit Button

- Submit Button คือ ปุ่มสำหรับสั่งให้เริ่มส่งข้อมูลที่ใช้กรอกหรือเลือกในแบบฟอร์มไปยัง Server โดยไฟล์ที่คอยรับข้อมูลที่อยู่บน Server จะระบุไว้ใน Attribute action

```
1 <!doctype html>
2 <html>
3   <body>
4     <form name="input" action="html_form_action.php" method="get">
5       Username: <input type="text" name="user">
6       <input type="submit" value="Submit">
7     </form>
8   </body>
9 </html>
```

Username:

➡ Button

- Button คือ ปุ่มทั่วไปที่ใช้ในการกำหนดให้ทำงานเมื่อมีการใช้เมาส์คลิกที่ปุ่ม

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <input type="button" value="Hello world!">
6     </form>
7   </body>
8 </html>
```

Hello world!

➡ Password

- `<input type="password">` ใช้ในการรับ input ที่เป็น password

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Password: <input type="password" name="pwd">
6     </form>
7   </body>
8 </html>
```

Password:

➡ email

- `<input type="email">` ใช้ในการรับข้อมูลที่อยู่ในรูปแบบอีเมลเท่านั้น

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       E-mail: <input type="email" name="usermail">
6       <input type="submit">
7     </form>
8   </body>
9 </html>
```

E-mail: porche@

! Please enter an email address.

➡ URL

- `<input type="url">` ใช้ในการรับข้อมูลที่อยู่ในรูปแบบ URL เท่านั้น

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Add your homepage:
6       <input type="url" name="homepage">
7       <input type="submit">
8     </form>
9   </body>
10 </html>
```

Add your homepage:

www.google.com|

Submit

! Please enter a URL.

➡ Number

- `<input type="number">` ใช้ในการรับข้อมูลที่เป็นตัวเลขเท่านั้น ประกอบด้วย Attribute ดังนี้

max - ค่าสูงสุด

min - ค่าต่ำสุด

step - ช่วงห่างแต่ละค่า

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Quantity (between 1 and 5):
6       <input type="number" name="quantity" min="1" max="5">
7       <input type="submit">
8     </form>
9   </body>
10 </html>
```

Quantity (between 1 and 5):

! Value must be less than or equal to 5.

➡ Radio Buttons

- `<input type="radio">` ใช้ในการรับ input แบบตัวเลือกอย่างใดอย่างหนึ่ง ซึ่งเรียกว่า radio button

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <input type="radio" name="sex" value="male">Male<br>
6       <input type="radio" name="sex" value="female">Female
7     </form>
8   </body>
9 </html>
```

☐ Male
☒ Female

Checkbox

- `<input type="checkbox">` ใช้ในการรับ input แบบตัวเลือก ซึ่งมีการเลือกตั้งแต่ 0 ตัวเลือกขึ้นไป

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <input type="checkbox" name="vehicle" value="Bike">I have a bike<br>
6       <input type="checkbox" name="vehicle" value="Car">I have a car
7     </form>
8   </body>
9 </html>
```

☒ I have a bike
☒ I have a car

File

- `<input type="file">` ใช้ในการรับ input ที่เป็นไฟล์

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Select your photo : <input type="file" name="photo">
6     </form>
7   </body>
8 </html>
```

Select your photo : No file chosen

➡ Drop-Down List

- Drop-Down List คือ การแสดงรายการให้ผู้เลือกใช้

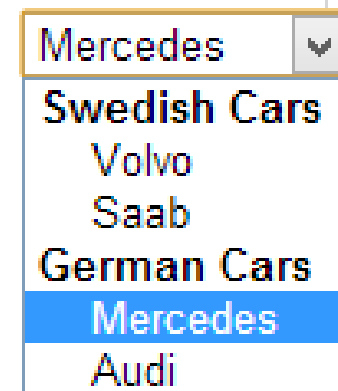
```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <select name="cars">
6         <option value="volvo">Volvo</option>
7         <option value="saab">Saab</option>
8         <option value="fiat">Fiat</option>
9         <option value="audi">Audi</option>
10      </select>
11    </form>
12  </body>
13 </html>
```



➡ การจัดกลุ่ม Drop-Down List

- ใช้ `<optgroup>` ในการกำหนดชื่อกลุ่มที่อยู่ในรายการของ Drop-Down List

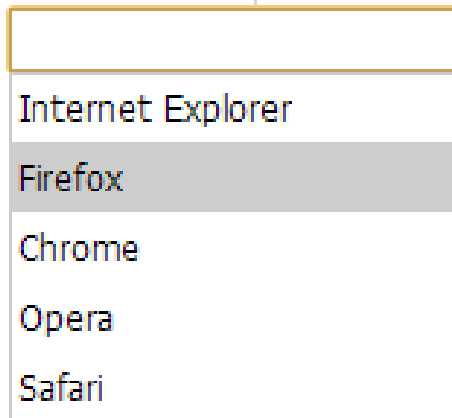
```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <select>
6         <optgroup label="Swedish Cars">
7           <option value="volvo">Volvo</option>
8           <option value="saab">Saab</option>
9         </optgroup>
10        <optgroup label="German Cars">
11          <option value="mercedes">Mercedes</option>
12          <option value="audi">Audi</option>
13        </optgroup>
14      </select>
15    </form>
16  </body>
17 </html>
```



➡ Data List

- ใช้ในการแสดงรายการที่อยู่ในรูปแบบ Text Field โดยจะแสดงรายการเมื่อผู้ใช้ใส่ตัวอักษรที่ตรงตามรายการที่มีอยู่ ซึ่งการทำงานส่วนนี้จะจะเป็นแบบเดียวกับ autocomplete

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <input list="browsers">
6
7       <datalist id="browsers">
8         <option value="Internet Explorer">
9         <option value="Firefox">
10        <option value="Chrome">
11        <option value="Opera">
12        <option value="Safari">
13      </datalist>
14    </form>
15  </body>
16 </html>
```



Internet Explorer
Firefox
Chrome
Opera
Safari

➡ Textarea

- ใช้ในการรับ input ข้อความแบบหลายบรรทัด

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <textarea rows="10" cols="30">
6         The cat was playing in the garden.
7       </textarea>
8     </form>
9   </body>
10 </html>
```

The cat was playing in the garden.

➡ Range

- ใช้ในการรับค่าตัวเลข โดยวิธีเลื่อนเลือกตามช่วงที่กำหนด
- Attribute ที่เกี่ยวข้อง

max - ค่าสูงสุด, min - ค่าต่ำสุด

step - ช่วงห่างแต่ละค่า

value – ค่าเริ่มต้น

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Points:
6       <input type="range" name="points" min="1" max="10">
7     </form>
8   </body>
9 </html>
```

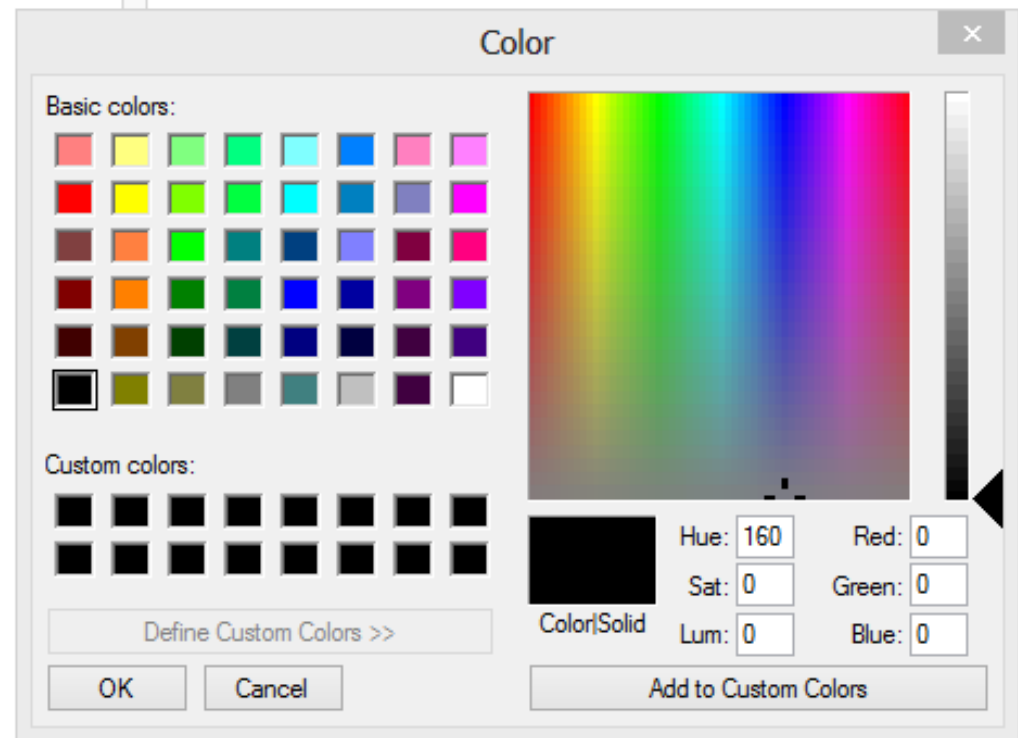
Points: 

➡ Color

- ใช้ในการรับค่าสี โดยให้ผู้ใช้เลือกสีที่ต้องการ

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Select your favorite color:
6       <input type="color" name="favcolor">
7     </form>
8   </body>
9 </html>
```

Select your favorite color:



➡ Date

- ใช้ในการรับค่าวันที่ โดยให้ผู้ใช้งานเลือกวันที่ต้องการจากปฏิทิน

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Birthdate: <input type="date" name="bday">
6     </form>
7   </body>
8 </html>
```

Birthdate: mm/dd/yyyy

October 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

➡ Fieldset

- Fieldset คือ การจัดกลุ่มของ input ต่างๆ ด้วยการตีกรอบและเขียนคำอธิบาย

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       <fieldset>
6         <legend>Personal information:</legend>
7         Name: <input type="text" size="30"><br>
8         E-mail: <input type="text" size="30"><br>
9         Date of birth: <input type="text" size="10">
10      </fieldset>
11    </form>
12  </body>
13 </html>
```

Personal information:

Name:

E-mail:

Date of birth:

➡ Attribute ที่ใช้ใน <form> และ <input>

- required
- placeholder
- autocomplete
- novalidate
- autofocus
- multiple

➡ required

- กำหนดให้ต้องกรอก หรือต้องเลือก input นั้น

```
1 <!doctype html>
2 <html>
3   <body>
4     <form action="demo_form.asp">
5       Username:
6       <input type="text" name="user" required>
7       <input type="submit">
8     </form>
9   </body>
10 </html>
```

Username:

! Please fill out this field.

➡ placeholder

- แสดงคำอธิบายให้กับ Text Field ชนิดต่างๆ

```
1 <!doctype html>
2 <html>
3   <body>
4     <form action="demo_form.asp">
5       <input type="text" placeholder="First name"><br>
6       <input type="text" placeholder="Last name"><br>
7       <input type="submit">
8     </form>
9   </body>
10 </html>
```

autocomplete

- ใช้ในการระบุใน `<form>` หรือ `<input>` เพื่อเปิดหรือปิด autocomplete

```
<form autocomplete="on">
```

```
  First name:<input type="text" name="fname"><br>
```

```
  Last name: <input type="text" name="lname"><br>
```

```
  E-mail: <input type="email" name="email" autocomplete="off"><br>
```

```
  <input type="submit">
```

```
</form>
```

novalidate

- ใช้ในการปิดตรวจสอบค่าในฟอร์มทั้งหมด

```
<form novalidate>
```

```
  E-mail: <input type="email" name="user_email">
```

```
  <input type="submit">
```

```
</form>
```

autofocus

- กำหนดให้ Focus ไปที่ Input

First name: `<input type="text" name="fname" autofocus>`

multiple

- กำหนดให้ File หรือ Email สามารถกรอกหรือเลือกได้หลายค่า

```
<form>
```

```
  Select images: <input type="file" name="img" multiple="multiple">
```

```
  <input type="email" multiple>
```

```
  <input type="submit">
```

```
</form>
```

➡ การกำหนดรูปแบบของ input

- ใช้ attribute pattern เพื่อกำหนดรูปแบบของ input ซึ่งจะใช้ Regular Expression ในการกำหนด

```
1 <!doctype html>
2 <html>
3   <body>
4     <form>
5       Country code:
6       <input type="text" name="code"
7         pattern="[A-Za-z]{3}"
8         title="Three letter country code">
9       <input type="submit">
10    </form>
11  </body>
12 </html>
```

Country code:

! Please match the requested format.
Three letter country code

➔ Regular Expression

- Regular Expression เรียกว่า Regex คือ ประโยคสัญลักษณ์ที่ใช้ระบุรูปแบบ (Pattern) ที่ต้องการ
- ประโยชน์ของ Regular Expression เช่น
 - ตรวจสอบ input จากผู้ใช้ ว่าถูกต้องตามต้องการหรือไม่ เช่น เบอร์โทรศัพท์ รหัสไปรษณีย์
 - ตรวจสอบว่าข้อความนั้นมีคำสั่ง HTML ปะปนมาด้วยหรือไม่

➔ Literal Characters

- Pattern กำหนดให้เป็นตัวอักษรตามที่ต้องการเท่านั้น

Regex : is

Input : this ✗

Regex : KK

Input : kK ✗

Regex : is

Input : is ✓

Regex : KK

Input : KK ✓

✗ - ไม่ถูกต้องตามกฎ

✓ - ถูกต้องตามกฎ

➡ อักขระพิเศษ

- อักขระที่สงวนสำหรับการเขียนกฎ regex จะต้องนำหน้าด้วยเครื่องหมาย \

[\ ^ \$. | ? * + ()

Regex : 1\+1=2

Input : 1+1=2 ✓

Regex : 1+1=2

Input : 1+1=2 ✗

➡ Period

- สัญลักษณ์จุด . แทนอักขระ 1 ตัว

Regex : a.boy

Input : a boy ✓

Regex : a.boy

Input : aboy ✗

➔ Character Classes

- Character Classes คือ การ Match กับอักขระตัวใดตัวหนึ่งที่อยู่ภายในเครื่องหมาย []

Regex : [Gg]r[ae]y

Input : Gray ✓

Regex : [Gg]r[ae]y

Input : grey ✓

Regex : [Gg]r[ae]y

Input : Ggry ✗

Regex : [Gg]r[ae]y

Input : rey ✗

➡ Character Classes

- ใช้เครื่องหมายลบ – ภายในเครื่องหมาย [] เพื่อกำหนดช่วง

Regex : [0-9a-fA-F]

Input : A ✓

Regex : cus[0-9]

Input : cus1 ✓

Regex : [0-9a-fA-F]

Input : 0a ✗

Regex : cus[0-9]

Input : cus ✗

➔ Character Classes

- ใช้ `[^]` เพื่อ Match กลุ่มคำที่ไม่ต้องการให้มีในข้อความ

Regex : `q[^u]`

Input : `qi` ✓

Regex : `q[^u]`

Input : `qu` ✗

➔ Shorthand Character Classes

- การกำหนดรูปแบบของ Character Classes อาจใช้สัญลักษณ์อื่นได้

`\d` – แทน `[0-9]`

`\w` – แทน `[A-Za-z0-9_]`

`\s` – แทน whitespace หรือ `[\t\r\n]` (space, tab, CR, LF)

Regex : `\d\s\w`

Input : 1 a ✓

Regex : `\d\s\w`

Input : 1 a3 ✗

➔ Shorthand Character Classes

- การกำหนดรูปแบบของ Character Classes อาจใช้สัญลักษณ์อื่นได้

`\D` – แทน non-digit หรือ `[^\d]`

`\W` – แทน non-word หรือ `[^\w]`

`\S` – แทน non-whitespace หรือ `[^\s]`

Regex : `\D\D\d`

Input : mp3 ✓

Regex : `\D\D\d`

Input : k25 ✗

➡ Repetition

- ใช้สัญลักษณ์ * เพื่อบอกจำนวนอักขระว่ามีได้ตั้งแต่ 0 ตัวขึ้นไป

Regex : [A-Za-z][A-Za-z0-9]*5

Input : Html5 ✓

Regex : [A-Za-z][A-Za-z0-9]*5

Input : Html57 ✗

➡ Repetition

- ใช้สัญลักษณ์ + เพื่อบอกจำนวนอักขระว่าต้องมีตั้งแต่ 1 ตัวขึ้นไป

Regex : [A-Za-z][A-Za-z0-9]+

Input : h1 ✓

Regex : [A-Za-z][A-Za-z0-9]+

Input : h ✗

➔ Repetition

- ใช้สัญลักษณ์ ? เพื่อบอกจำนวนอักขระว่ามีได้เพียง 0 ตัว หรือ 1 ตัวเท่านั้น

Regex : [A-Za-z][A-Za-z0-9]?

Input : i ✓

Regex : [A-Za-z][A-Za-z0-9]?

Input : Hat ✗

➡ Anchors

- ใช้ `^` เพื่อ Match โดยเริ่มจากจุดเริ่มต้นของข้อความ

Regex : `^vac`

Input : vacation ✓

Regex : `^vac`

Input : evacuation ✗

Anchors

- ใช้ \$ เพื่อ Match โดยเริ่มจากจุดสิ้นสุดของข้อความ

Regex : \$tion

Input : evacuation ✓

Regex : \$tion

Input : national ✗

➔ Alternation

- ใช้ | กันข้อความ 2 ข้อความขึ้นไป เมื่อมีตัวเลือก Match หลายตัว

Regex : (cat|dog)+

Input : dog ✓

Regex : (cat|dog)+

Input : catastrophe ✗

➔ Limiting Repetition

- กำหนดจำนวนการเกิดอักขระด้วย { }

Regex : \d{2,4}

Input : 111 ✓

Regex : \w{2,}

Input : you ✓

Regex : \w{3}

Input : rat ✓

Regex : \d{2,4}

Input : 1 ✗

Regex : \w{2,}

Input : y ✗

Regex : \w{3}

Input : bigga ✗

➡ จงเขียน Regex ตามรูปแบบที่กำหนด

รูปแบบที่ต้องการ	Regex
เป็นสระในภาษาอังกฤษเท่านั้น และมีได้ 3 ตัว	
ขึ้นต้นด้วยตัวพิมพ์ใหญ่ตัวแรก ที่เหลือเป็น ตัวอักษรพิมพ์เล็กตั้งแต่ 1 ตัวขึ้นไป	
ขึ้นต้นด้วย ord หรือ order ลงท้ายด้วยตัวเลข 5 ตัว	
เป็นคำว่า frog หรือ blog หรือ clog เท่านั้น	

➡ จงเขียนข้อความที่ถูกต้องตามกฎ Regex

Regex	ตัวอย่างข้อความที่ถูกต้องตามกฎ
ABC	
[A-Z]	
[0-9]	
[09]	
[0][9]	
[A-Z][a-z]	
...	
[0-9A-Z].555	