

# MDT312 Web Programming

ผศ. ดร. เกื้อแก้ว ธีเนศวร

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

# ทำไมเราต้องเรียนวิชานี้

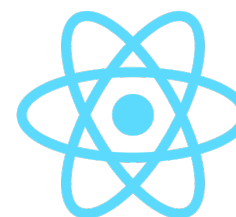


# ทำไมเราต้องเรียนวิชานี้

---



Bootstrap



รหัส/วิชา: MDT312 การเขียนเว็บโปรแกรม (Web programming)

วันและเวลา: วันอังคาร      กลุ่มที่ 1 8:30-13:20,  
  กลุ่มที่ 2 13:30-18:20

e-mail: [kejkaew.tha@gmail.com](mailto:kejkaew.tha@gmail.com)  
[,kejkaew.tha@kmutt.ac.th](mailto:kejkaew.tha@kmutt.ac.th)

## รายละเอียดรายวิชา

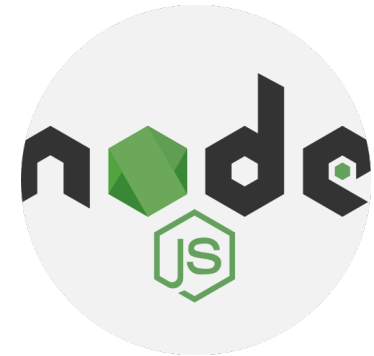
---

- การสร้างโปรแกรมประมวลผลบนระบบเว็บ การออกแบบส่วนติดต่อผู้ใช้ การสร้างเว็บแบบไดนามิก การใช้ประโยชน์โปรแกรมฝังลูกข่ายและแม่ข่าย การใช้โปรแกรมฝังลูกข่ายเพื่อควบคุมการทำงานของเว็บ กลไกคุกกี้และการสร้างเว็บที่เก็บสถานะ การใช้ทรัพยากรบนแม่ข่าย และในระบบเครือข่าย ระบบประมวลผลร่วมกับฐานข้อมูล ข้อคำนึงถึงด้านความปลอดภัยของระบบงาน

# สรุปง่าย ๆ วิชานี้เรียนเกี่ยวกับ...

---

- Internet คืออะไร
- Internet มีการทำงานอย่างไร
- เว็บไซต์ ทำงานอย่างไร
- และเราจะสร้างเว็บไซต์ได้อย่างไร



# สิ่งที่คาดหวังว่านักศึกษาจะได้จากวิชานี้

---

- Client – Server relationship
- Web programming skills
- Some of you will be a web developer.
- ความสนุกสนานจากการทำ website

## เอกสารประกอบการสอน:

---

- W3Schools website, <https://www.w3schools.com/>
- ครูก, ส., นิรันดร์ ประวิทย์ธนา, ซาลี พงษ์สง่างาน, & Krug, S. (2558). Don't make me think revisited แค่อ่ต้องคิดก็ผิดแล้ว ออกแบบเว็บไซต์ให้เข้าใจง่ายง่าย และไม่ต้องใช้สมอง (กลับมาอีกรั้ง). กรุงเทพฯ: แมกชินคิวิธพ์พับลิชซิ่ง.



## เกณฑ์การแบ่งคะแนน

ที่	วิธีการประเมิน	สัดส่วนของการประเมินผล
1	งานในห้องเรียน + Classroom Quiz (ทุกสัปดาห์)	25%
2	Portfolio Web Page: website + design (แต่ละคน)	15%
3	สอบ Quiz + สอบปลายภาค	35%
4	โปรเจกต์เกี่ยวกับ Web programming: website + design (กลุ่ม)	25%

# ข้อตกลงการใช้ Generative AI ในรายวิชา

- การใช้เพื่อการเรียนรู้ส่วนตัว
  - นักศึกษาสามารถใช้ Generative AI (เช่น ChatGPT, Copilot, Gemini ฯลฯ) เพื่อ
    - ทบทวนบทเรียน
    - อธิบายสิ่งที่ไม่เข้าใจ
    - ช่วยสรุปเนื้อหาเพื่อทำความเข้าใจ
  - ห้ามใช้ AI ตอบคำถามใน Classroom Quiz, Quiz, หรือสอบปลายภาค
  - ถ้ามีคำตอบเหมือนกันกับคนอื่น จะได้ 0 ทั้งคู่
  - หากตรวจพบการใช้ AI ในการสอบ ให้คะแนน 0 ในการสอบนั้นทันที
- การใช้เพื่อทำงานที่ได้รับมอบหมาย
  - นักศึกษาสามารถใช้ Generative AI ช่วย ทำงานสร้างสรรค์ เช่น การสร้างเว็บไซต์ งานเดี่ยวหรืองานกลุ่ม ได้
  - แต่ต้องระบุ Prompt หรือวิธีที่ใช้ AI ช่วยเหลือในงานที่ส่งทุกครั้ง (เพื่อความโปร่งใส)
- แนวทางจริยธรรม
  - ข้อมูลที่ได้จาก AI ควรมีการตรวจสอบความถูกต้องก่อนใช้งาน
  - หลีกเลี่ยงการคัดลอกผลงาน AI ทั้งหมดโดยไม่ปรับปรุงหรือทำความเข้าใจ

# 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		

# Web languages

---

- **Content:** Hypertext Markup Language (HTML): used for writing web pages
- **Style:** Cascading Style Sheets (CSS): stylistic info for web pages
- **Behavior:** JavaScript: interactive and programmable web pages
  - Asynchronous JavaScript and XML (Ajax): accessing data for web applications
  - Node.js: cross-platform and backend JavaScript runtime environment
- **Structured Query Language (SQL):** interaction with databases

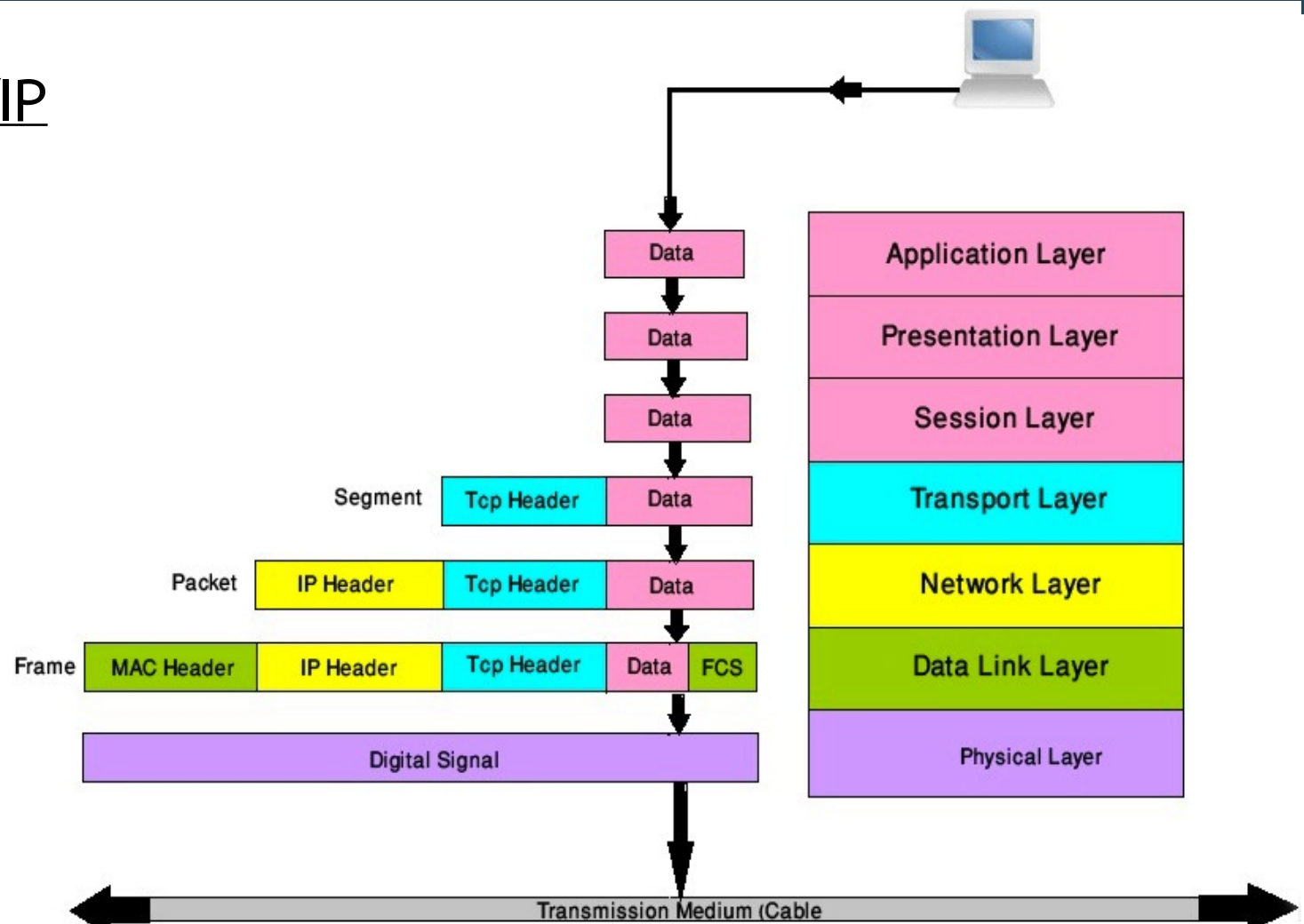
# Internet

---

- A connection of computer networks using the Internet Protocol (IP)
- Layers of communication protocols: IP → TCP/UDP → HTTP/FTP/POP/SMTP/SSH...



# โปรโตคอล TCP/IP



<http://routinglab.blogspot.com/2011/05/how-does-osi-model-function.html>

## การทำงานของคอมพิวเตอร์ในระบบเครือข่าย

---

- Client-Server
- เป็นระบบที่เครื่องคอมพิวเตอร์ทุกเครื่องมีฐานะการทำงานที่เหมือน ๆ กัน เท่าเทียมกันภายในระบบเครือข่าย แต่จะมีเครื่องคอมพิวเตอร์เครื่องหนึ่ง ที่ทำหน้าที่เป็นเครื่อง Server ที่ทำหน้าที่ให้บริการทรัพยากรต่าง ๆ กับ เครื่อง Client หรือเครื่องที่ขอใช้บริการ ซึ่งเครื่อง Server อาจจะต้องเป็นเครื่องที่มีประสิทธิภาพที่ค่อนข้างสูง

# World Wide Web

---

- web server: software that listens for web page requests (server)
  - Apache
  - Microsoft Internet Information Server (IIS)
  - Nginx
- web browser: fetches/displays documents from web servers (client)
  - Mozilla Firefox
  - Apple Safari
  - Microsoft Edge
  - Google Chrome
  - Opera



# The Internet: A brief history

---

- ค.ศ.1969 (พ.ศ.2512) - อินเทอร์เน็ต ถูกพัฒนาโดย ARPAnet = Advanced Research Projects Agency Network เปลี่ยนชื่อเป็นดาต้าเน็ต DARPA NET = Defense Advanced Research Projects Agency Network
- ค.ศ.1983 (พ.ศ.2526) ดาต้าเน็ตตัดสินใจนำ TCP/IP (Transmission Control Protocol/Internet Protocol) มาใช้กับเครื่องคอมพิวเตอร์ทุกเครื่องในระบบ
- ค.ศ.1986 (พ.ศ.2529) เริ่มใช้การกำหนดโดเมนเนม (Domain Name) เป็นการสร้างฐานข้อมูลแบบกระจาย (Distribution Database) อยู่ในแต่ละเครือข่าย และให้ ISP (Internet Service Provider) ช่วยจัดทำฐานข้อมูลของตนเอง จึงไม่จำเป็นต้องมีฐานข้อมูลแบบรวมศูนย์เหมือนแต่ก่อน

- ค.ศ.1991(พ.ศ.2534) ทิม เบอร์เนิร์ส ลี (Tim Berners-Lee) แห่งศูนย์วิจัย CERN ได้คิดค้นระบบไฮเปอร์เท็กซ์ขึ้น สามารถเปิดด้วย เว็บเบราว์เซอร์ (Web Browser) ตัวแรกมีชื่อว่า WWW (World Wide Web)
- โปรแกรม MOSAIC (โมเสค) โดย Marc Andreessen ซึ่งเป็นเว็บเบราว์เซอร์ระบบกราฟฟิก หลังจากนั้นทีมงานที่ทำโมเสคก็ได้ออกไปเปิดบริษัทเน็ตสเคป(Netscape)
- Amazon.com เริ่มต้นเมื่อ ปี 1995; Google เริ่มต้นเมื่อปี January 1996
- เริ่มมีการใช้ "Web 2.0" เมื่อ January 1999 โดย Darcy DiNucci. ซึ่งหมายความว่าเว็บไซต์ที่เน้นในการทำ User content, usability และ คำนึงถึงการใช้งานของ end users
- Facebook ก่อตั้งเมื่อ February 2004

# The Next Era of the Internet: Web3

- Web 3.0 มีการคาดการณ์ไว้ว่า จะเป็น The Next Era of the Internet ที่จะมีควมอัจฉริยะมากขึ้น สามารถทำงาน วิเคราะห์ข้อมูลต่าง ๆ ได้ใกล้เคียงกับมนุษย์มากขึ้น อีกทั้งยังสามารถทำงานได้แบบอัตโนมัติ และยุคนี้เองที่จะทำให้เทคโนโลยีอย่าง Machine Learning (ML), Big Data, AI, Blockchain และเทคโนโลยีอัจฉริยะอื่น ๆ สามารถทำงานได้อย่างเต็มที่ และวิวัฒนาการไปมีประสิทธิภาพมากยิ่งขึ้น
- Web 3.0 จะเป็นแหล่งกักเก็บ Data แบบ Decentralized หรือแบบกระจายออกจากศูนย์กลาง ซึ่งแตกต่างจาก Web 2.0 ที่ Data ส่วนใหญ่จะถูกจัดเก็บไว้แบบ Centralized หรือที่ศูนย์กลางมากกว่า

## Key Features: Web3

---

- Ubiquity = การมีความสามารถในการอยู่ทุก ๆ ที่ ทุก ๆ แห่ง ในเวลาเดียวกัน
- Semantic web = คล้ายมนุษย์ ฉลาดมากขึ้น และรองรับเทคโนโลยีอัจฉริยะได้
- Artificial Intelligence
- Spatial Web and 3D Graphics = การปิดช่องว่างระหว่างโลกดิจิทัล กับโลกจริง โดยการปฏิวัติเทคโนโลยีด้านกราฟิก ให้สามารถออกแบบ และสร้างโลก 3D เสมือนจริงขึ้นมาได้

## คำถาม

- 
- ใครคือผู้ควบคุมและดูแล เรื่องที่เกี่ยวกับ Internet

## ผู้ควบคุมและดูแลระบบอินเทอร์เน็ต

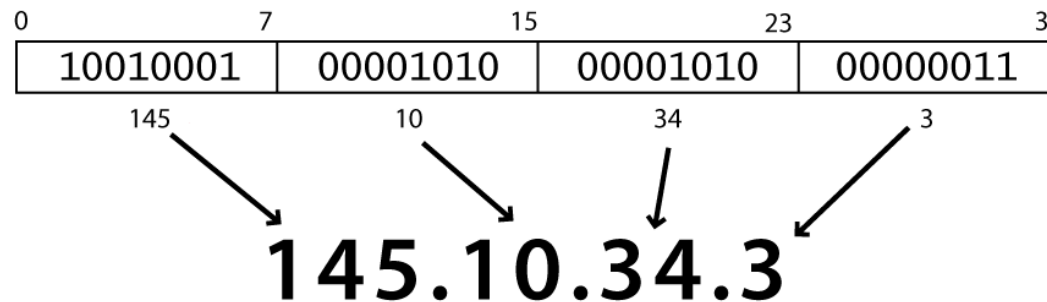
---

- Internet Engineering Task Force (IETF): internet protocol standards
- Internet Corporation for Assigned Names and Numbers (ICANN): จัดการ top-level domain names
- World Wide Web Consortium (W3C): web standards



## Internet Protocol (IPv4 )

- เป็น protocol ที่ใช้สำหรับส่งข้อมูลระหว่างคอมพิวเตอร์
- each device has a 32-bit IP address written as four 8-bit numbers (0-255)



- find out your internet IP address: [whatismyip.com](http://whatismyip.com)
- find out your local IP address:
  - in a terminal, type: ipconfig (Windows) or ifconfig (Mac/Linux)

# Domain Name System (DNS)

- ชื่อของ web site ที่ใช้เรียกแทน ip address
- ในแต่ละชื่อโดเมนเนมจะประกอบด้วยชุดตัวอักษรตั้งแต่สองกลุ่มขึ้นไปที่คั่นด้วยจุด (dot) ซึ่งมีการเรียกแต่ละส่วนของชุดตัวอักษร ในแต่ละชื่อเต็มของโดเมนหนึ่งๆ เป็นลำดับๆไปดังนี้
- ส่วนขวาสุด: เรียกว่าโดเมนระดับบนสุด (Top-Level Domain) ซึ่งเป็นองค์ประกอบสำคัญที่จะปรากฏในทุก ๆ ชื่อโดเมน เช่น www.kmutt.ac.th มี top-level domain คือ th
  - Example: www.kmutt.ac.th → 202.44.11.241
  - www.google.com -> 142.251.46.174



# Uniform Resource Locator (URL)

---

- ที่อยู่หรือสิ่งที่ระบุตำแหน่งที่อยู่ของ เอกสารข้อมูลของ web site นั้น
- a basic URL: http://www.ddddd.com/info/stepp/index.html  


protocol      host      path
- แสดงผลของ document บน web browser ของเรา

# Hypertext Transport Protocol (HTTP)

---

- เป็นโพรโทคอลหลักที่ใช้ในการแลกเปลี่ยนข้อมูล (HTML) กันระหว่าง Web Server และ Web Client (Browser)
- ใช้ URL (Uniform Resource Locator) ในการเข้าถึงเว็บไซต์ (Web Site) ซึ่งจะขึ้นต้นด้วย http:// ตามด้วยชื่อของเว็บไซต์
- ทำงานที่พอร์ต(port) 80 (มาตรฐาน)
- ส่งข้อมูลเป็นแบบ Clear text คือ ไม่มีการเข้ารหัสข้อมูลในระหว่างการส่ง (None-Encryption) จึงสามารถถูกดักจับได้ และอ่านข้อมูลนั้นรู้เรื่อง

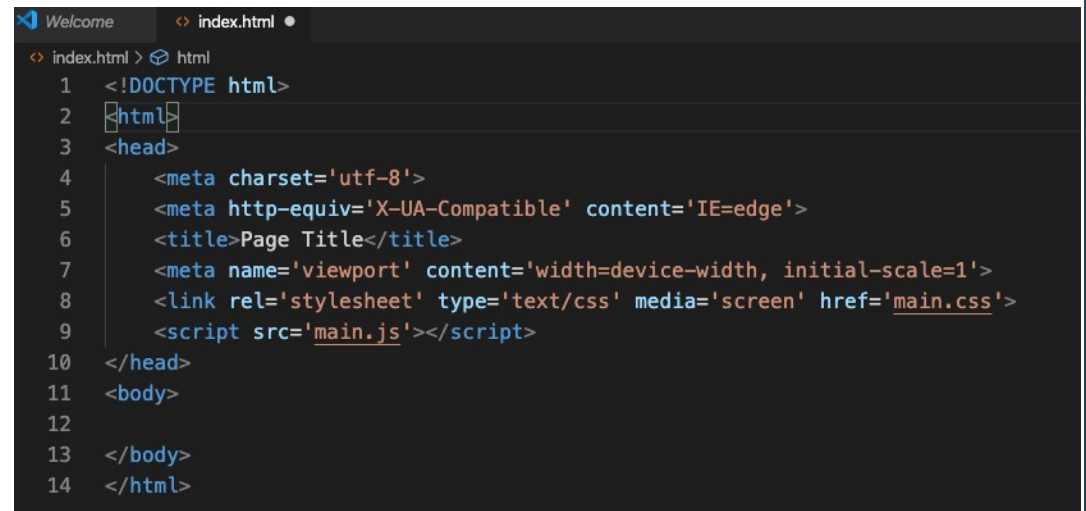
# Hypertext Markup Language (HTML)

---

- ใช้เพื่อแสดง content and structure of information บน a web page
- ประกอบไปด้วย opening and closing tags
- แต่ละ tags เราเรียกว่า **element**
  - syntax: **<element>** content **</element>**
  - example: **<p>**This is a paragraph**</p>**
- Newer version called HTML5

# Structure of an HTML page

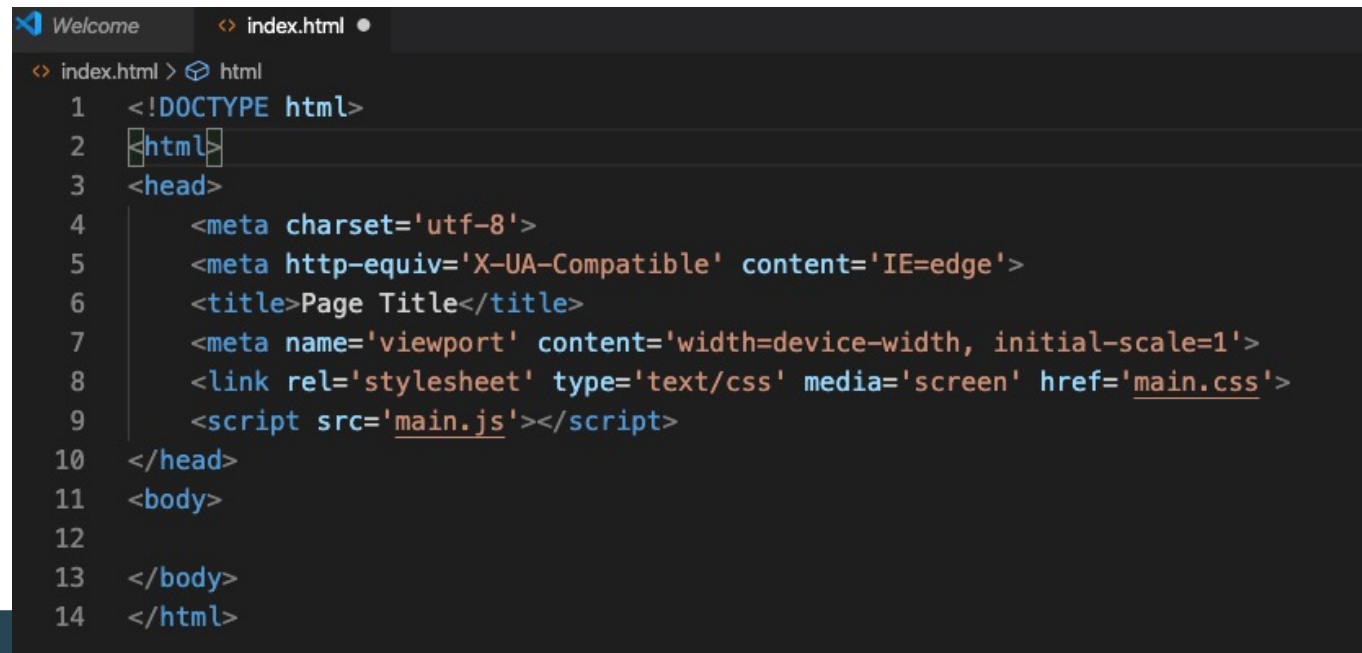
- header ทำหน้าที่อธิบายถึง web page นั้น และส่วน body สำหรับเนื้อหาของ website นั้น
- HTML page file จะมีนามสกุล เป็น .html หรือ .htm
- DOCTYPE tag ทำหน้าที่บอก web browser ว่า เราจะ รันแบบ HTML5, the latest/greatest version of the language นะ



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset='utf-8'>
5   <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6   <title>Page Title</title>
7   <meta name='viewport' content='width=device-width, initial-scale=1'>
8   <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
9   <script src='main.js'></script>
10 </head>
11 <body>
12
13 </body>
14 </html>
```

## Page title: <title>

- placed within the head of the page
- displayed in the web browser's title bar and when bookmarking the page



```
index.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset='utf-8'>
5      <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6      <title>Page Title</title>
7      <meta name='viewport' content='width=device-width, initial-scale=1'>
8      <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
9      <script src='main.js'></script>
10 </head>
11 <body>
12
13 </body>
14 </html>
```

## Paragraph: <p>

- placed within  
the body of the page
- more paragraph  
examples
- ตัวอย่าง

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset='utf-8'>
5   <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6   <title>Page Title</title>
7   <meta name='viewport' content='width=device-width, initial-scale=1'>
8   <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
9   <script src='main.js'></script>
10 </head>
11 <body>
12
13   <p>The Way Get Start Is To Quit Talking And Begin Doing.| -Walt Disney</p>
14
15 </body>
16 </html>
```

The Way Get Start Is To Quit Talking And Begin Doing.| -Walt Disney

Headings: <h1>, <h2>, <h3>, <h4>, <h5> , <h6>

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset='utf-8'>
5   <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6   <title>Page Title</title>
7   <meta name='viewport' content='width=device-width, initial-scale=1'>
8   <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
9   <script src='main.js'></script>
10 </head>
11 <body>
12   <h1>Kejkaew Thanasuan</h1>
13   <h2>เก็จนแก้ว ฅเนศวร</h2>
14   <h5>Computer Science</h5>
15   <p>The Way Get Start Is To Quit Talking And Begin Doing. | -Walt Disney</p>
16 </body>
17 </html>
```

- More heading examples

# Kejkaew Thanasuan

เก็จแก้ว ธารนคร

Computer Science

The Way Get Start Is To Quit Talking And Begin Doing. | -Walt Disney



## <article> and <section>

- <article> is a standalone piece of content (eg, entire blog post, including title, author, etc) (block)
- <section> is a piece of content that doesn't make sense on it's own (a chapter, paragraph, etc) (block)

```
<article> <h2>Thanasuan's resume:</h2>
<section> <h3>Objective:</h3> ... </section>
<section> <h3>Experience:</h3> ... </section>
<section> <h3>References:</h3> ... </section>
</article>
```

### **Thanasuan's resume:**

**Objective:**

...

**Experience:**

...

**References:**

...

## Horizontal rule: <hr>

- a horizontal line to visually separate sections of a page (block)

```
<p>First paragraph</p> <hr /> <p>Second paragraph</p>  
<hr> <p>Third paragraph</p> First paragraph
```

First paragraph

---

Second paragraph

---

Third paragraph

First paragraph

## Links: <a>

- links, or "anchors", to other pages (inline)
- uses the href attribute to specify the destination URL
  - can be absolute (to another web site) or relative (to another page on this site)
- anchors are inline elements; must be placed in a block element such as p or h1

```
<p>  
Search for it on  
<a href="http://www.google.com/">Google</a>!  
</p>
```

Search for it on [Google](http://www.google.com/)

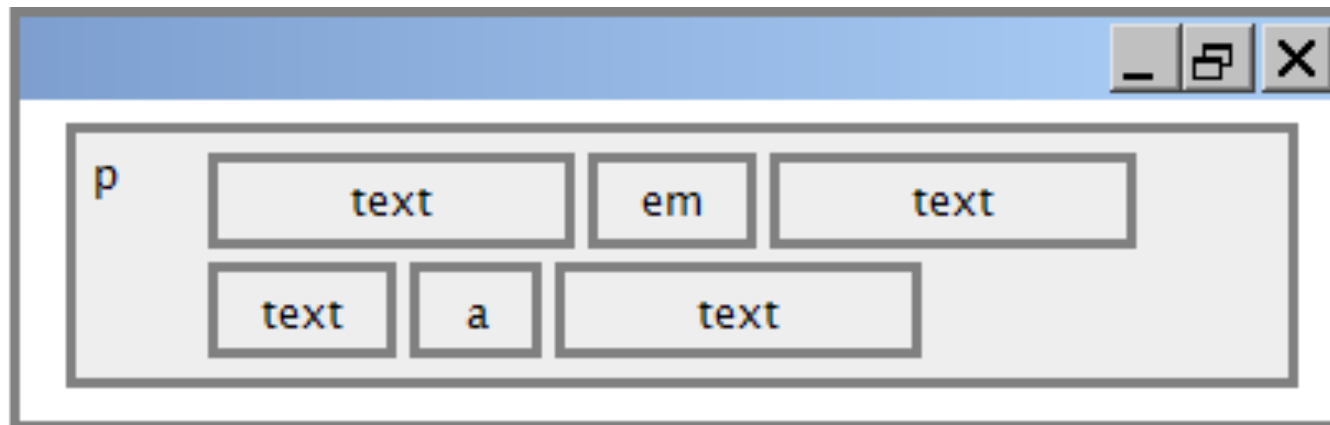
## More about HTML tags

---

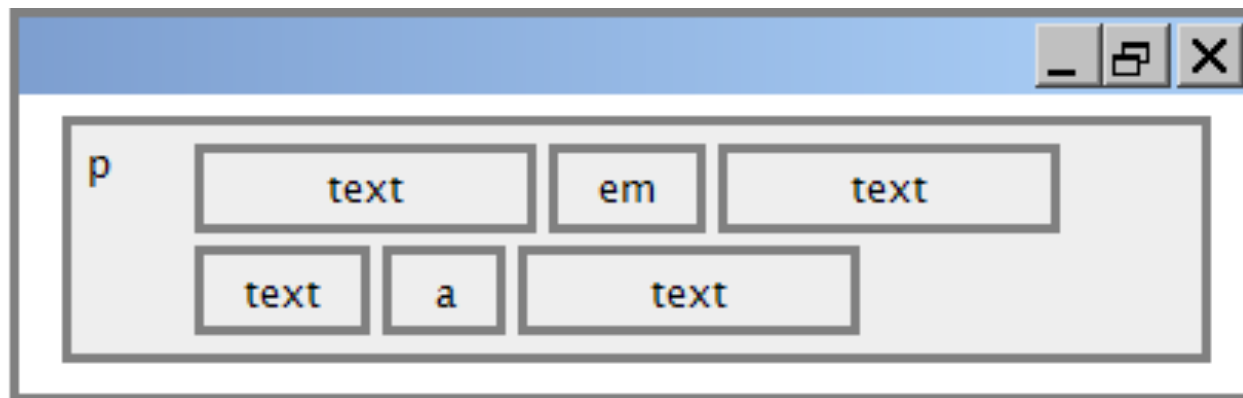
- some tags can contain additional information called **attributes**
  - syntax: `<element attribute="value" attribute="value"> content </element>`
  - example: `<a href="page2.html">Next page</a>`
- some tags don't contain content; can be opened and closed in one tag
  - syntax: `<element attribute="value" attribute="value" />`
  - example: `<hr />`, `<hr />`, `<hr>`, `<hr >`
  - example: ``

## Block and inline elements

- block elements จะกำหนดความกว้าง (width) ของ element โดยความกว้างของหน้าจอหรือของเพจนั้น
- examples: paragraphs, lists, table cells, div



- inline elements จะถูกกำหนดความกว้าง (width) โดยเนื้อหาของ content  
examples: span, bold text, link
- The browser allows many inline elements to appear on the same line  
must be nested inside a block element



## ข้อยกเว้น

---

สำหรับ Block/inline:

- Block บางประเภท เช่น `<body>`, `<form>` ไม่สามารถมี block เดิมข้างในได้
- `<p>` สามารถใส่ได้แค่ inline elements และ plain text เท่านั้น
- Block บางประเภทสามารถใส่ได้ทั้ง block และ inline เช่น `<div>`, `<li>`
- `<ul>` สามารถมีได้แค่ `<li>` เท่านั้น
- บาง elements สามารถมีได้แค่ 1 element ใน document เช่น `<html>`

## Images: <img>

- การใส่ graphical image ลงใน page (แสดงผลแบบ inline-block)
- ``



the **src** attribute มาจากคำว่า source = ที่อยู่ของ image file

the **alt attribute** จะใช้สำหรับแสดงข้อความก็ต่อเมื่อรูปไม่ขึ้น (เกิด error) หรือใช้สำหรับผู้พิการทางสายตา



## Line break: <br>

- ใช้สำหรับขึ้นบรรทัดใหม่ (inline)

```
<p>The woods are lovely, dark and deep, <br /> But I  
have promises to keep,  
<br /> And miles to go before I sleep, <br /> And  
miles to go before I sleep.</p>
```

The woods are lovely, dark and deep,  
But I have promises to keep,  
And miles to go before I sleep,  
And miles to go before I sleep.

## Phrase elements : <em>, <strong>

- em: emphasized text (ตัวอักษรเอียง)
- strong: strongly emphasized text (ตัวอักษรหนา)

```
<p>  
  HTML is <em>really</em>,  
  <strong>REALLY</strong> fun!  
</p>
```

output:      HTML is *really*, **REALLY** fun!

## Nesting tags

- tags must be correctly nested
  - (เปิดและปิด tags ให้ถูกต้อง)
- ถึงแม้ว่า browser จะทำงานได้ แต่ เป็น HTML code ที่ไม่ถูกต้อง

**What is wrong in this code?**

```
<p>  
  HTML is <em>really,  
  <strong>REALLY</em> lots of</strong> fun!  
</p>
```

HTML is *really*, **REALLY** fun!

## Comments: <!-- ... -->

- คำสั่ง สำหรับ comment
- Comments ไม่สามารถซ้อนกันได้

```
<!-- My web page, by Kejkaew -->  
<p>MDT419 web programming course is <!-- NOT --> a  
lot of fun!</p>
```

## Unordered list: <ul>, <li>

- ul สำหรับแสดงผล bulleted list of items ที่ไม่เป็นลำดับ (block)
- li แทน แต่ละ list (block)

```
<ul>  
  <li>No shoes</li>  
  <li>No shirt</li>  
  <li>No problem!</li>  
</ul>
```

- No shoes
- No shirt
- No problem!

## More about unordered lists

---

```
<ul>
  <li>Simpsons:
    <ul>
      <li>Homer</li>
      <li>Marge</li>
    </ul>
  </li>
  <li>Family Guy:
    <ul>
      <li>Peter</li>
      <li>Lois</li>
    </ul>
  </li>
</ul>
```

- Simpsons:
  - Homer
  - Marge
- Family Guy:
  - Peter
  - Lois

## Ordered list: <ol>

- ol สำหรับแสดงผล a numbered list of items (block) เป็นลำดับ

```
<p>RIAA business model:</p>
<ol>
  <li>Sue customers</li>
  <li>???</li>
  <li>Profit!</li>
</ol>
```

RIAA business model:

1. Sue customers
2. ???
3. Profit!

## Web page metadata: <meta>

- แท็ก <meta> ใน HTML เป็น metadata ของหน้าเว็บ หรือข้อมูลที่บอกเกี่ยวกับหน้าเว็บนั้น ๆ แต่ ไม่แสดงผลโดยตรงบนหน้าเว็บ ให้ผู้ใช้เห็น โดยใช้เพื่อสื่อสารกับเบราว์เซอร์, เครื่องมือค้นหา (Search Engines), และ Social Media
- placed in the head section of your HTML page



# Meta Attributes

Attribute	Value	Description
<u>charset</u>	<i>character_set</i>	Specifies the character encoding for the HTML document
<u>content</u>	<i>text</i>	Specifies the value associated with the http-equiv or name attribute
<u>http-equiv</u>	content-security-policy content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
<u>name</u>	application-name author description generator keywords viewport	Specifies a name for the metadata

# Example of meta tag

---

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

## Example of meta tag

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



- You can read more about the viewport in our [Responsive Web Design - The Viewport Tutorial](#).

## Favorites icon ("favicon")

---

- `<link href="filename" type="MIME type" rel="shortcut icon" />`

ตัวอย่าง

- `<link href="google.png" type="image/png" rel="shortcut icon" />`
- the link tag, placed in the head section, attaches another file to the page in this case, an icon to be placed in the browser title bar and bookmarks



google - Google Search



## Week 1: Classroom game

---

- ตอบคำถามใน LEB2 ภายในวันอังคารหน้า

# Assignment 1

---

- สร้าง Portfolio web page ของนักศึกษาแต่ละคนสำหรับใช้เพื่อการสมัครงาน โดยใช้ HTML ง่าย ๆ และมีรายละเอียดข้อมูลดังนี้
  - มีชื่อ - นามสกุล
  - งานอดิเรก
  - ทักษะ ความสามารถ
  - ความสนใจ
  - รูป
  - มี Link ไปเว็บอื่น เช่น Facebook, Instagram ของเรา
  - หรือรายละเอียดอื่น ๆ (ข้อมูลส่วนตัว ไม่เอานะ!)
- ส่งด้วยนะคะ
- Portfolio web page นี้ เราจะใช้จนถึงเรื่อง JavaScript นะคะ เก็บไว้ดี ๆ