

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่ 3 (1-4-4) **วิชาซีพเลือก

Modern Web Technology Development



Lecture02

HTML คืออะไร และมีโครงสร้างเป็นอย่างไร

สอนโดย อ.ธนิต เกตุแก้ว

หลักสูตรวิศวกรรมซอฟต์แวร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่ 3(1-4-4) **วิชาซีพเลือก

Modern Web Technology Development

ศึกษาและฝึกปฏิบัติการเกี่ยวกับ

- การประยุกต์ใช้ CSS ใน การออกแบบเว็บไซต์
- การพัฒนาเว็บไซต์ที่เชื่อมต่อกับฐานข้อมูล และการแปลงข้อมูล
- การโปรแกรมผู้ใช้เครื่องลูกข่าย
- การโปรแกรมผู้ใช้เครื่องแม่ข่าย เว็บเซอร์วิส และ
- การ ประยุกต์ใช้เฟรมเวิร์คในการจัดทำโปรแกรมบนเว็บที่มีประสิทธิภาพ และ
- วิธีการในการรักษาความปลอดภัยบนเว็บไซต์

จุดมุ่งหมายของรายวิชา: การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

- เข้าใจพื้นฐานการออกแบบเว็บไซต์

นักศึกษาสามารถประยุกต์ใช้ CSS เพื่อออกแบบและจัดรูปแบบเว็บไซต์ให้สวยงามและตอบสนองต่ออุปกรณ์ต่าง ๆ ได้อย่างมีประสิทธิภาพ

- พัฒนาความสามารถในการเชื่อมต่อเว็บไซต์กับฐานข้อมูล

นักศึกษาสามารถพัฒนาเว็บไซต์ที่เชื่อมตอกับฐานข้อมูลเพื่อจัดการและแสดงข้อมูลผ่านฟังก์ชันของเว็บไซต์

- เพิ่มทักษะในการโปรแกรมฝั่งเครื่องลูกข่าย (Frontend)

นักศึกษาสามารถใช้ JavaScript ในการโปรแกรมฝั่งลูกข่าย เพื่อสร้างการโต้ตอบและประสบการณ์ที่ดีให้แก่ผู้ใช้

- พัฒนาทักษะการโปรแกรมฝั่งเครื่องแม่ข่าย (Backend)

นักศึกษาสามารถพัฒนาเซิร์ฟเวอร์ด้วย Node.js หรือ Express.js เพื่อจัดการการส่งและรับข้อมูลจากเว็บไซต์

จุดมุ่งหมายของรายวิชา: การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

- สร้างและใช้เว็บเซอร์วิสที่มีประสิทธิภาพ

นักศึกษา สามารถออกแบบและพัฒนาเว็บเซอร์วิส (Web Services) โดยใช้ RESTful API เพื่อรับและส่งข้อมูลระหว่างระบบได้อย่างมีประสิทธิภาพ

- ประยุกต์ใช้เฟรมเวิร์คในการพัฒนาเว็บแอปพลิเคชัน

นักศึกษา สามารถใช้เฟรมเวิร์คสมัยใหม่ เช่น React.js หรือ Vue.js ในการพัฒนาเว็บแอปพลิเคชันที่มีประสิทธิภาพและความยืดหยุ่นสูง

- มีความเข้าใจในด้านการรักษาความปลอดภัยเว็บไซต์

นักศึกษา สามารถระบุและป้องกันการโจมตีทางเว็บไซต์ เช่น SQL Injection, Cross-Site Scripting (XSS), และจัดการการยืนยันตัวตนเพื่อป้องข้อมูลบนเว็บไซต์

ลีปดาห์ที่ 1-2 : บทนำสู่การพัฒนาเว็บ

หัวข้อเรียนรู้ :

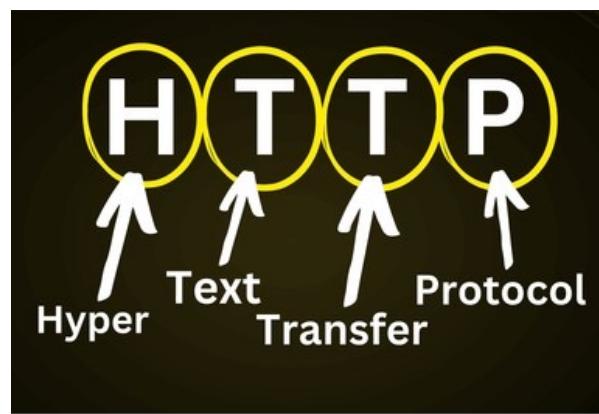
1. พื้นฐานโครงสร้างเว็บไซต์ (HTML)
2. การทำงานของเว็บเบราว์เซอร์และเซิร์ฟเวอร์ (HTTP/HTTPS)
3. การแนะนำเครื่องมือพัฒนาเว็บ (VSCode, Git)

กิจกรรม : การติดตั้งเครื่องมือพัฒนาเว็บ

งานที่ต้องทำ : ติดตั้งและตั้งค่า VSCode และ GitHub

http://

HTTP (HyperText Transfer Protocol)

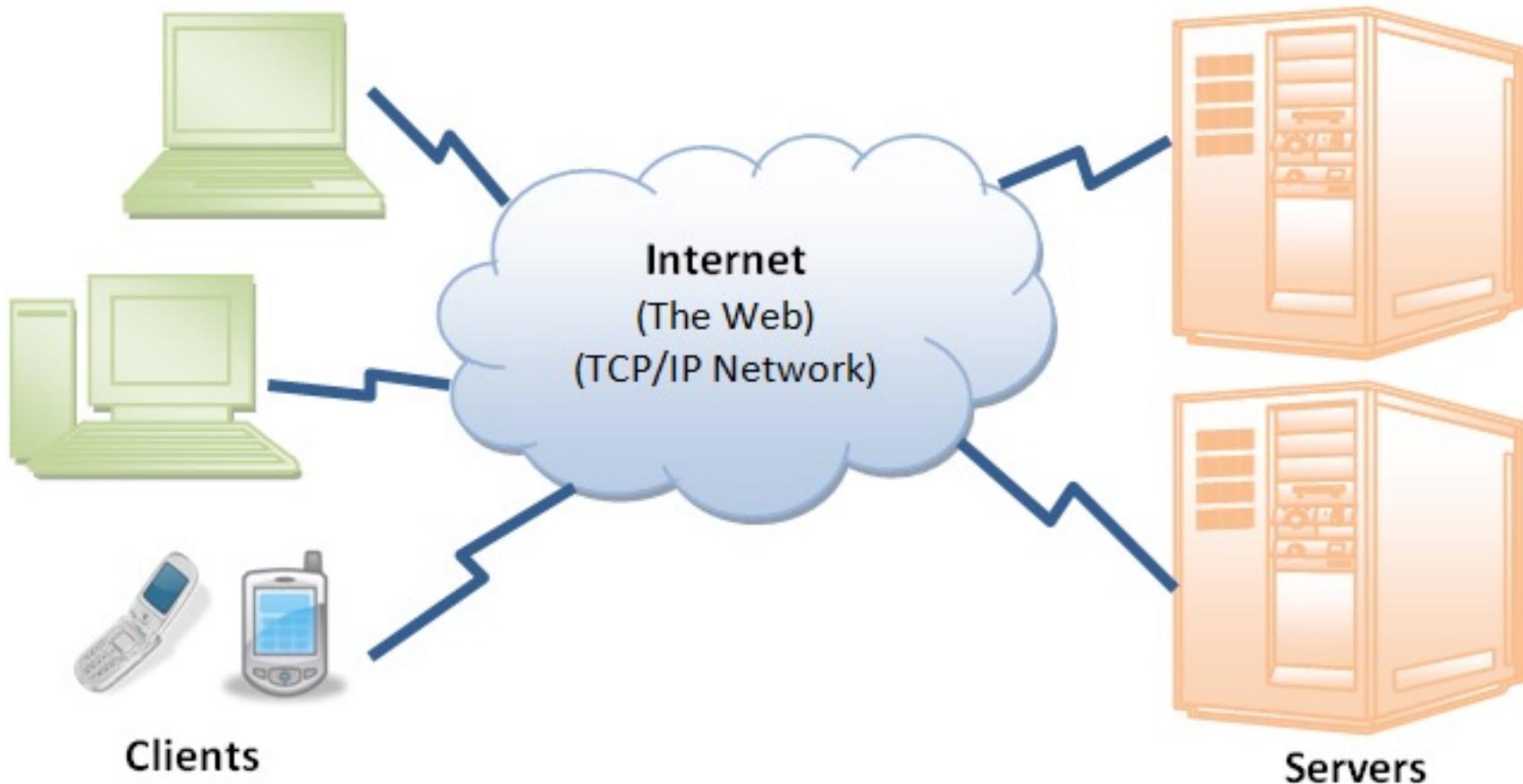


shutterstock.com - 2404109391

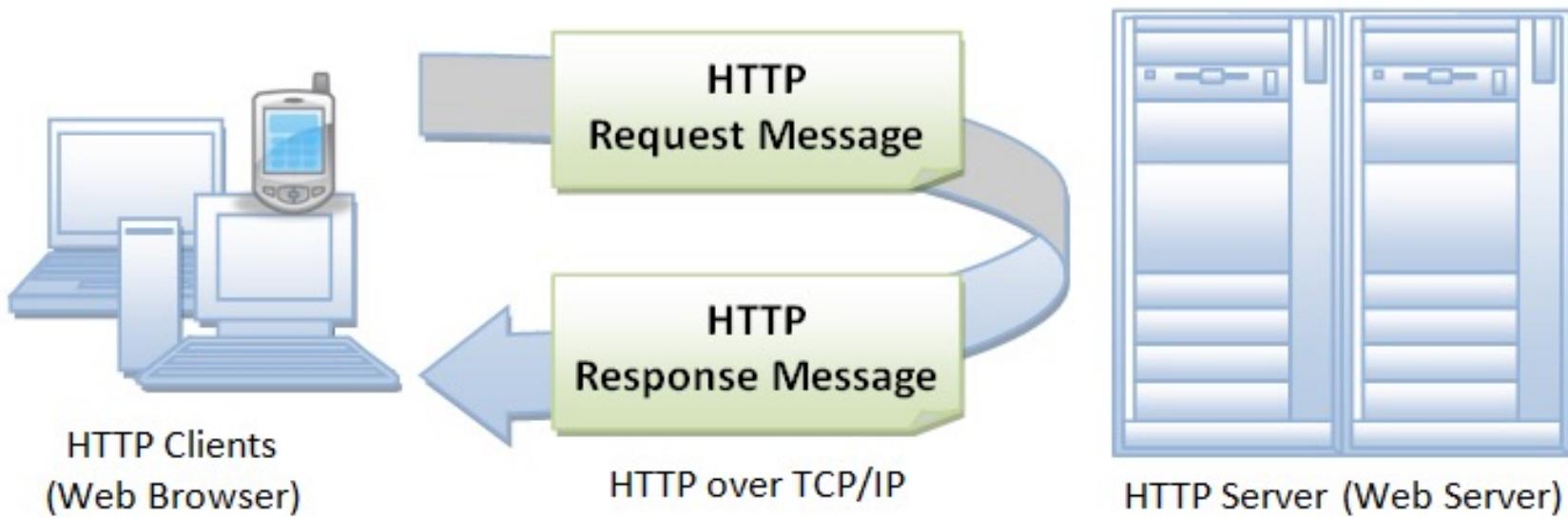
HTTP (HyperText Transfer Protocol)

The WEB

Internet (or The Web) is a massive distributed client/server information system as depicted in the following diagram.



HyperText Transfer Protocol (HTTP)



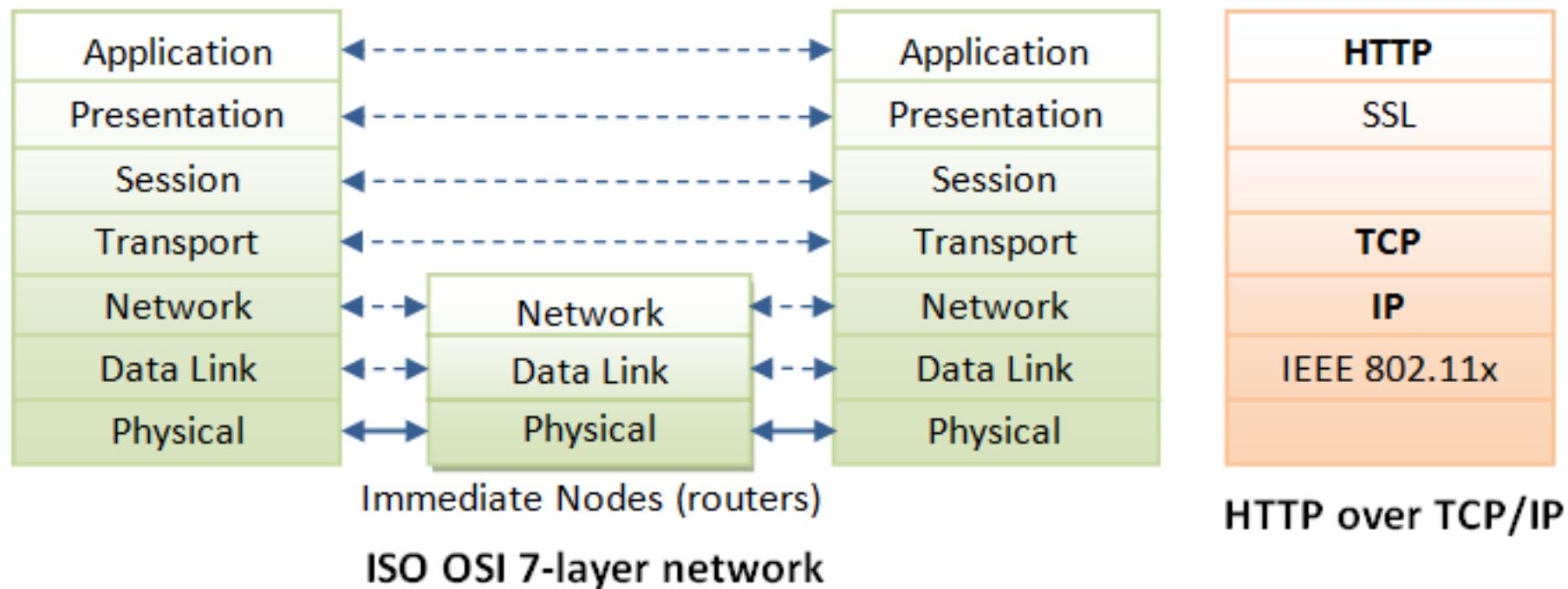
Hypertext Transfer Protocol (HTTP)

- เป็น protocol ระดับ Application Layer
- เป็น protocol หลักสำหรับการสื่อสารผ่านเว็บ
 - HTTP 1.1 เป็น version ที่นิยมใช้มากที่สุดในปัจจุบัน*
 - โดยมี HTTP/2 (เริ่มใช้ 2015) ที่ใช้งานในเว็บขนาดใหญ่
- มี method ที่สำคัญ เช่น
 - GET สำหรับร้องขอข้อมูลจาก Server
 - POST สำหรับส่งข้อมูลไปยัง Server เพื่อสร้างหรืออัปเดต resource

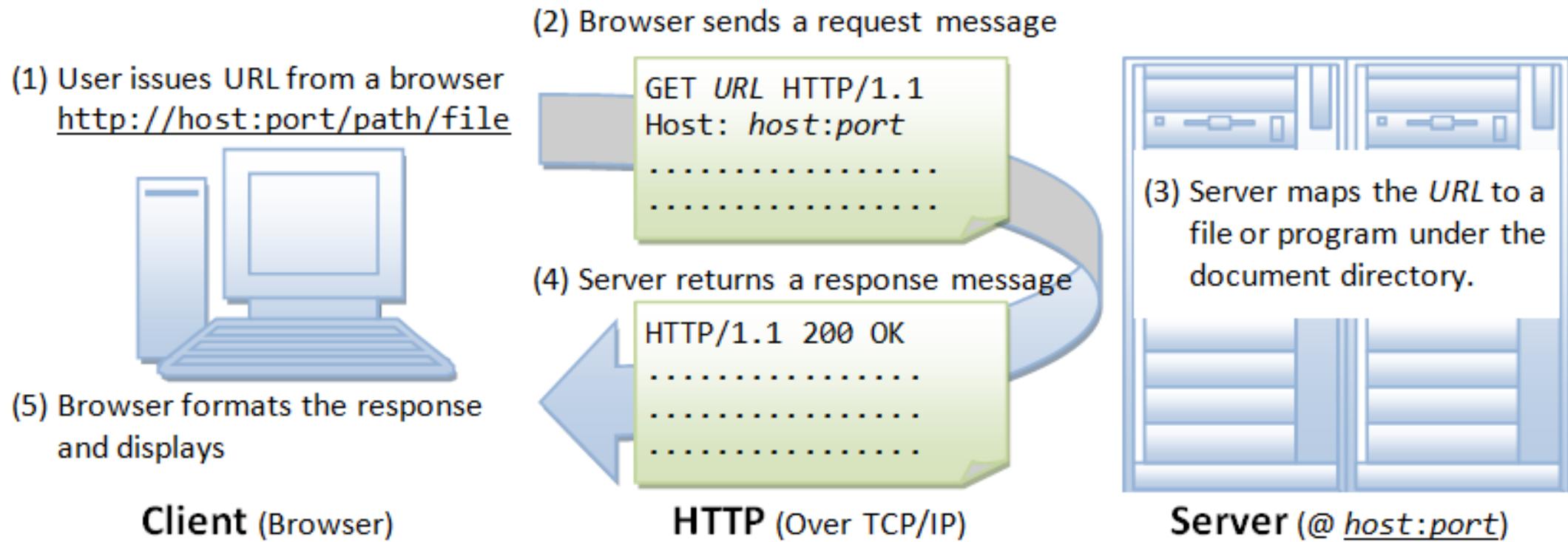
*Source: <https://w3techs.com/technologies/details/ce-hete?retrieved> April 2023

HTTP over TCP/IP

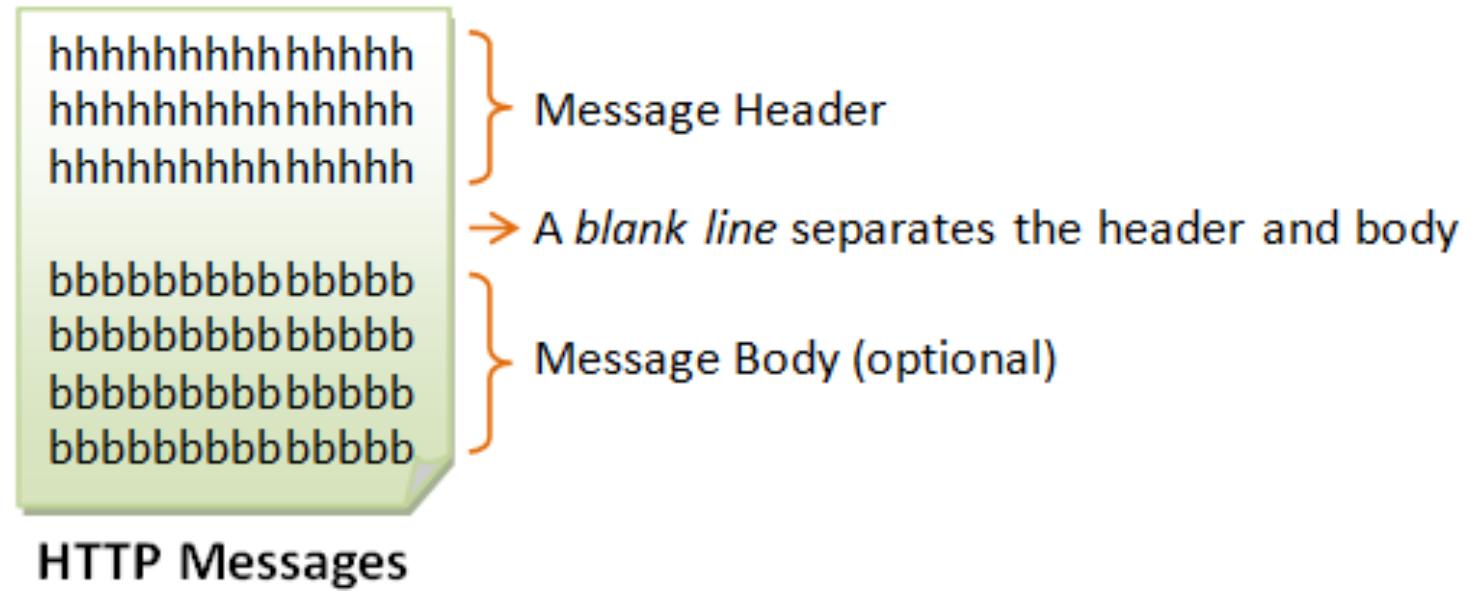
HTTP is a client-server application-level protocol. It typically runs over a TCP/IP connection, as illustrated. (HTTP needs not run on TCP/IP. It only presumes a reliable transport. Any transport protocols that provide such guarantees can be used.)



Browser

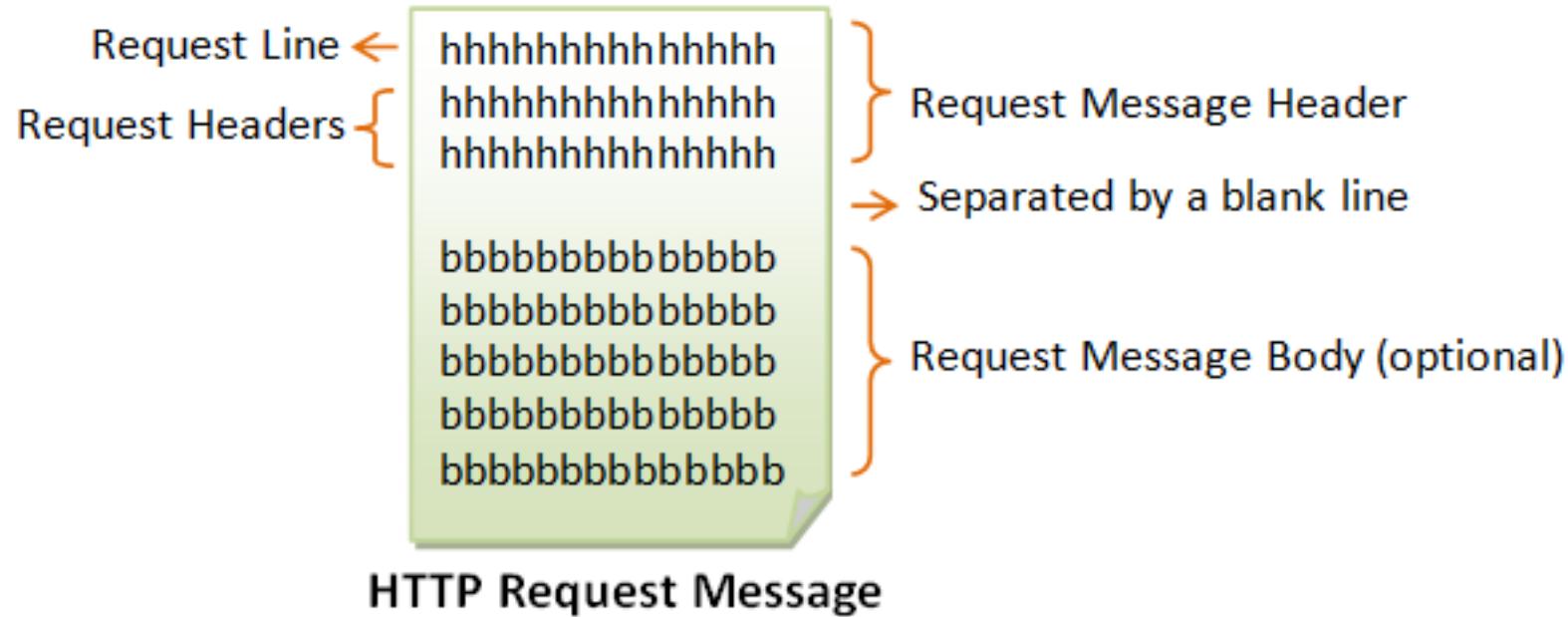


HTTP Request and Response Messages



HTTP Request Message

The format of an HTTP request message is as follow:



Example

The following shows a sample HTTP request message:

The diagram illustrates the structure of an HTTP request message. It is enclosed in a light green rounded rectangle. On the left, the request message is shown with its components labeled on the right:

- Request Line:** `GET /doc/test.html HTTP/1.1`
- Request Headers:** A group of headers:
 - `Host: www.test101.com`
 - `Accept: image/gif, image/jpeg, */*`
 - `Accept-Language: en-us`
 - `Accept-Encoding: gzip, deflate`
 - `User-Agent: Mozilla/4.0`
 - `Content-Length: 35`
- A blank line separates header & body**
- Request Message Body:** `bookId=12345&author=Tan+Ah+Teck`

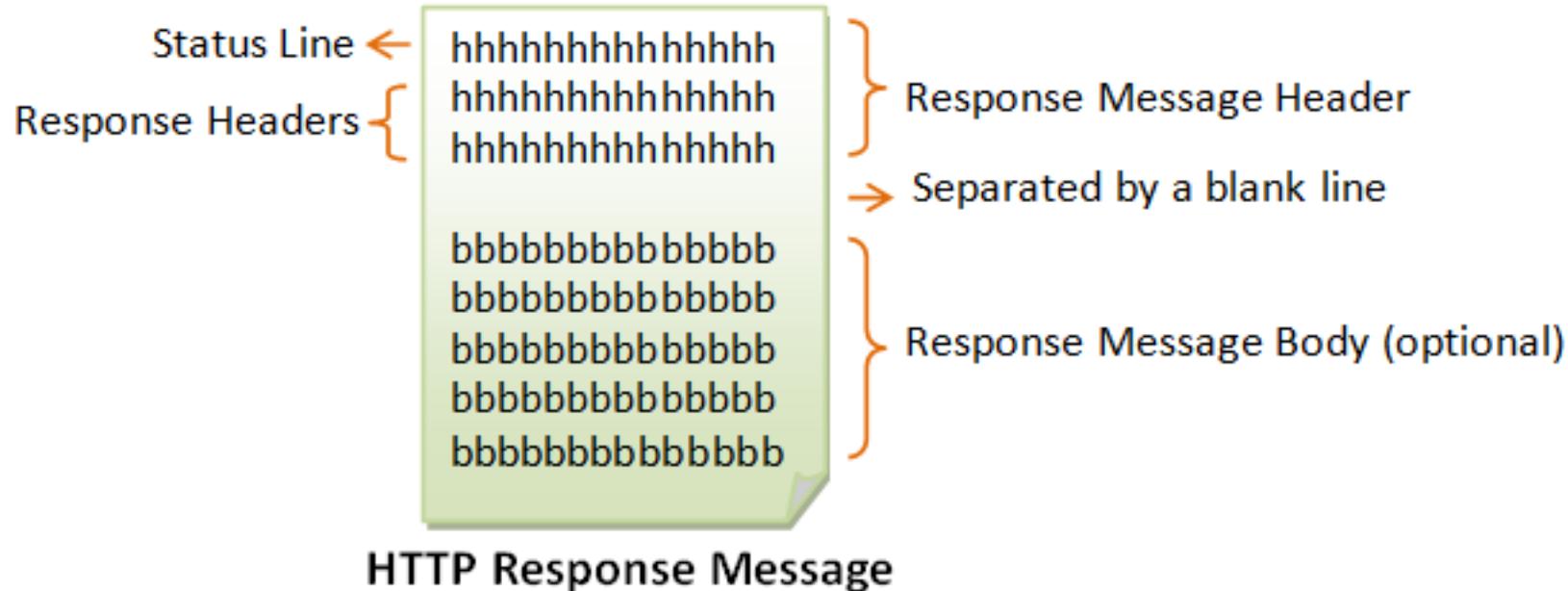
Annotations with orange arrows and curly braces point from the labels to their corresponding parts in the request message. A horizontal orange line separates the headers from the body.

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

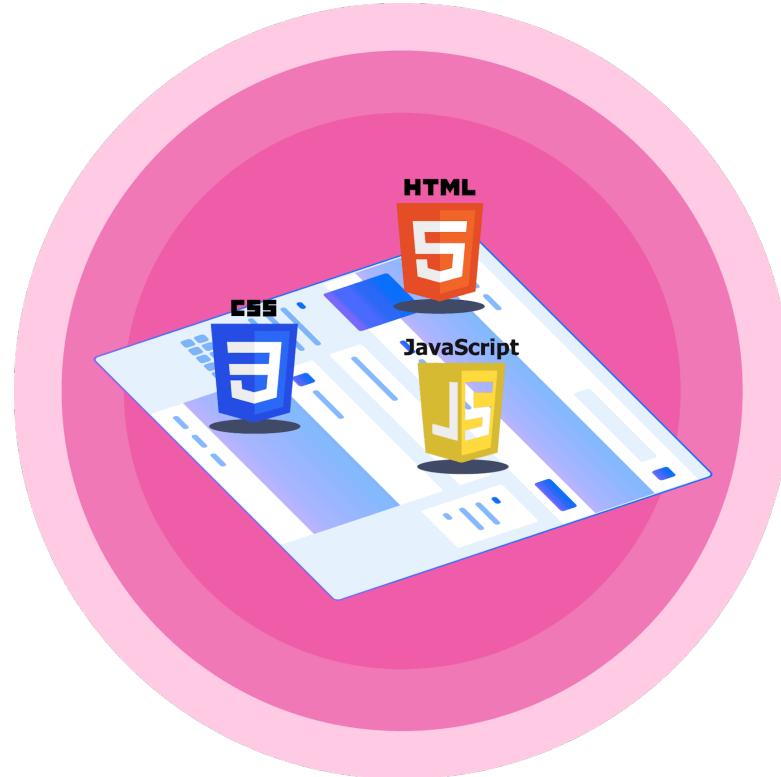
HTTP Response Message

The format of the HTTP response message is as follows:



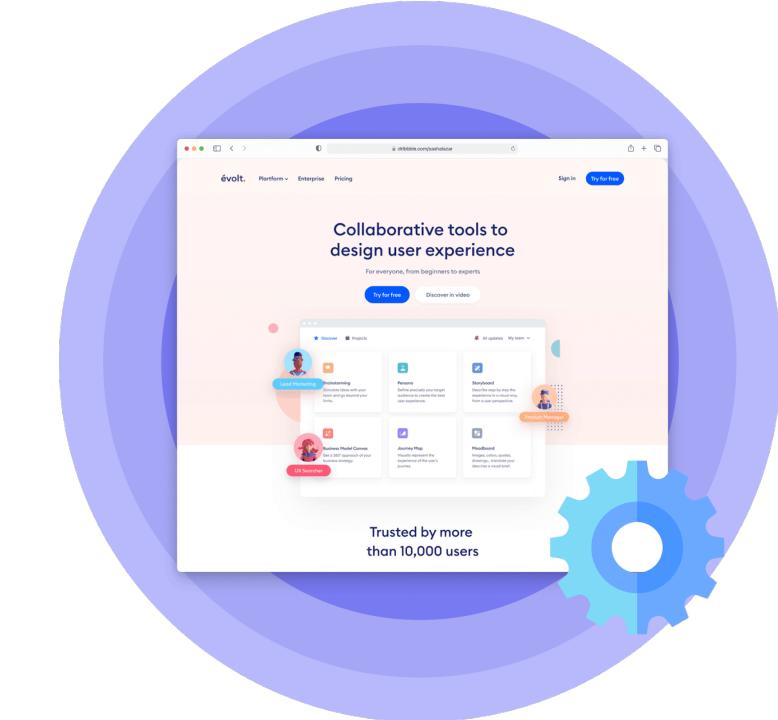


HTML คืออะไร? สอนเขียน HTML สำหรับมือใหม่แบบละเอียด



01. HTML คือ?

HTML คืออะไร และมีโครงสร้างเป็นอย่างไร
วันนี้เราจะมาทำความคิดกัน



02. Website ทำงานยังไง

กว่าจะมาเป็นหน้า Website ที่เราเห็นกันต้องผ่านขั้นตอนอะไรมา
บ้าง



03. Chrome dev tool

การใช้ Chrome inspection ให้มีประสิทธิภาพสูงสุดในการพัฒนา Website



04. VS code

Editor ที่มาแรงที่สุดในปัจจุบัน กับ Extension ที่จะทำให้การ Dev ง่ายขึ้น

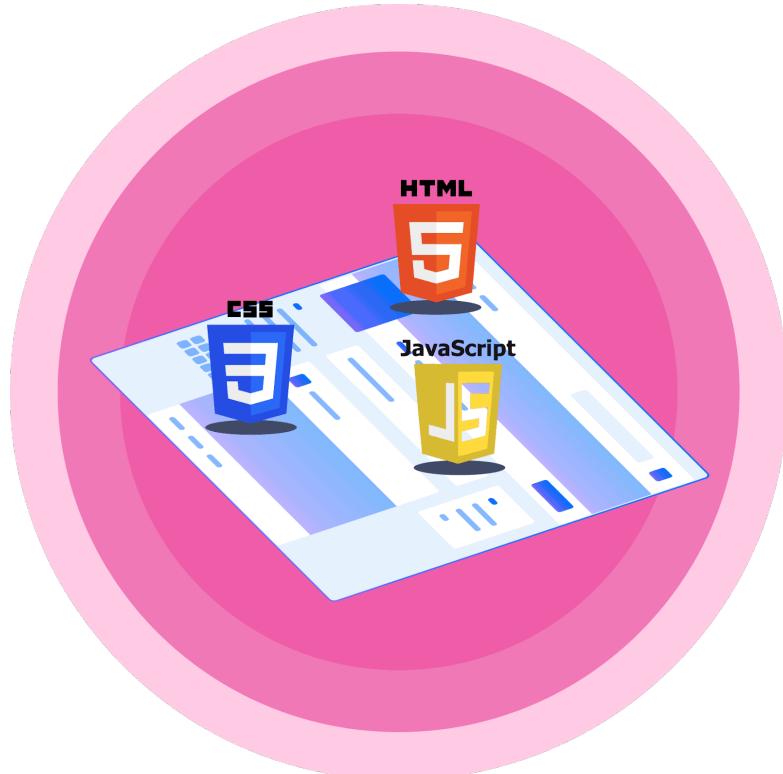


05. พื้นฐาน HTML

เริ่มเขียน HTML สร้าง Text, Image, link รวมทั้งการตกแต่งด้วย Style attribute

01. HTML คือ?

HTML คืออะไร และมีโครงสร้างเป็นอย่างไร วันนี้เราจะมาหาคำตอบกัน



ในบทแรกจะพูดถึงองค์ประกอบ HTML element และโครงสร้าง HTML document ว่ามีส่วนประกอบอะไรบ้าง รวมทั้ง พื้นฐาน HTML ที่ควรรู้

และ CSS selector พื้นฐานที่จำเป็นในการตกแต่ง HTML document ให้สวยงาม

HTML คืออะไร ?

HTML คือ ภาษาควบคุมพิวเตอร์ ที่มีโครงสร้างของการเขียนโดยอาศัยตัวกำกับหรือที่เรียกว่า Tag ใช้ควบคุม การแสดงผลหน้าจอเว็บไซต์ เช่น รูปภาพ ขอความเป็นตน [CSS](#) ใช้ในการตกแต่ง ตัว HTML ให้สวยงาม และ ตอบสนองต่อ Device หลายรูปแบบ [JavaScript](#) จะใช้ในการสร้างหรือพัฒนาระบบการทำงานให้ซับซ้อนหรือมีการ Interaction มากขึ้น

HTML



CSS



JS



ถ้าเปรียบเทียบ Website กับอวัยวะของมนุษย์
HTML คือ กระดูกส่วนต่างๆ
[CSS](#) คือ หุน ผิวพรรณ หนาตา
[JavaScript](#) คือ ระบบต่างๆในร่างกาย

HTML element ประกอบไปด้วยอะไรบ้าง



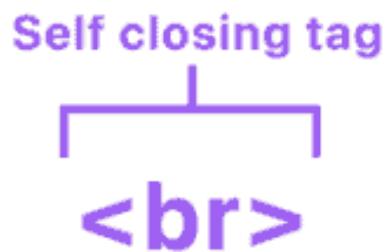
จากรูปจะเห็น HTML จะเริ่มด้วย Start tag และจบที่ End tag โดยมีเนื้อหาหรือข้อมูล (Content) คุ้นเคยอยู่โดยปกติในแต่ละ Tag จะประกอบไปด้วยแค่ Element name ที่จะบอกว่า Element ตัวนี้ทำหน้าที่อะไร ตัวอย่างเช่น: <H1> ย่อมาจาก Heading ทำหน้าที่เป็นหัวข้อเนื้อหา

แต่ในกรณี ที่เราต้องการเพิ่มเติมลักษณะพิเศษให้กับ HTML element เราจะใช้ Attribute ซึ่งจะอยู่ใน Start tag เช่น

HTML self-closing element



Self closing element



Example

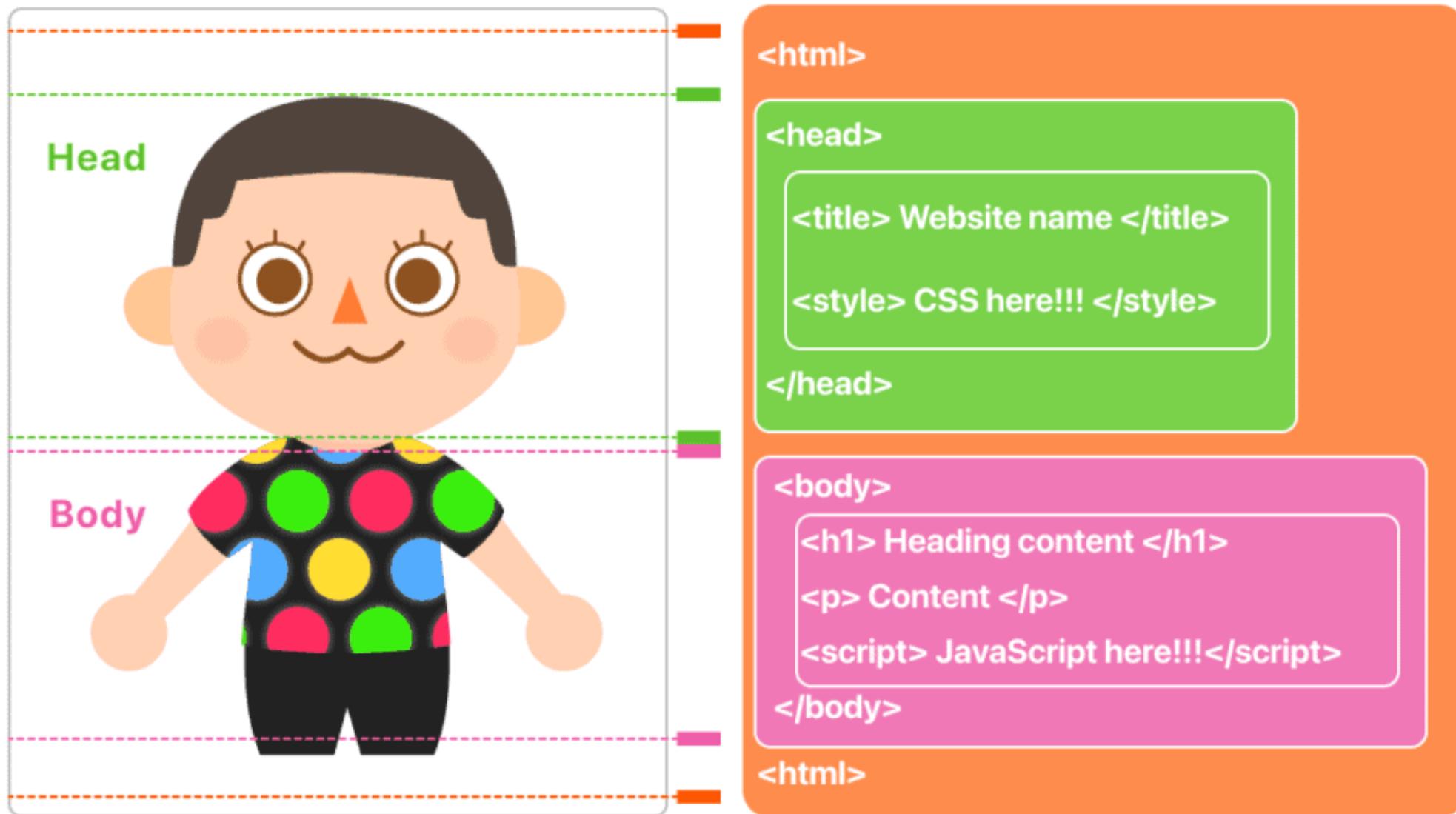
- 1.

- 2. <embed>
- 3. <input>
- 4. <hr>
- 5.
- 6. <link>
- 7. <meta>
- 8. <param>
- 9. <source>

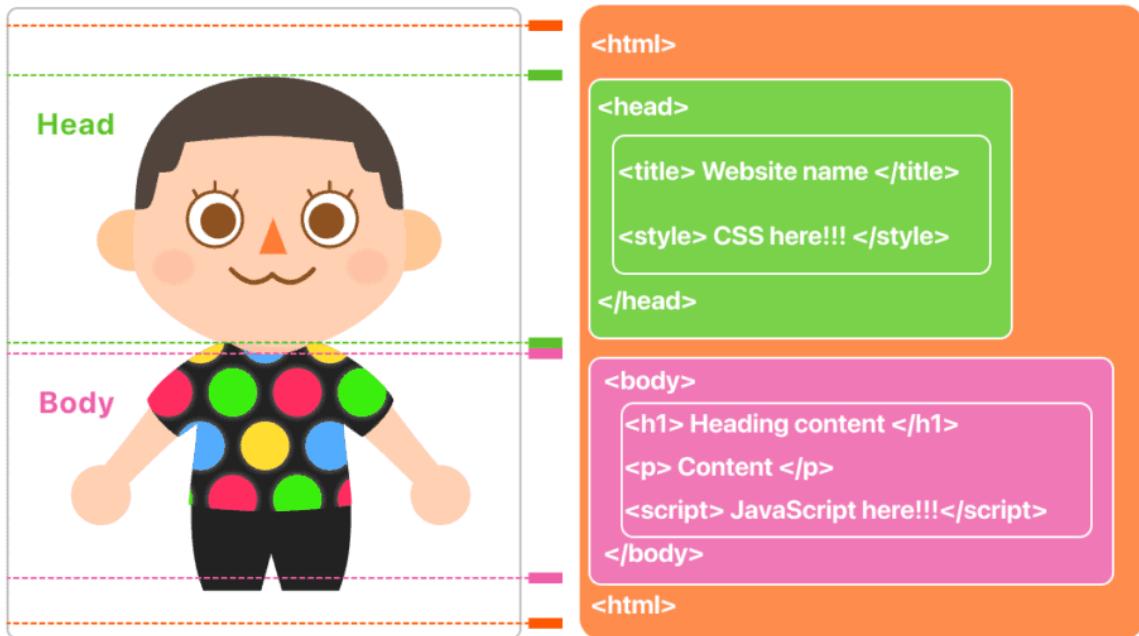
Self-Closing element คือ HTML Element ที่ไม่ต้องมี Close tag เป็นองค์ประกอบ



HTML Structure



HTML Structure



จากรูปจะเห็นได้ว่า โครงสร้าง HTML มีรูปแบบเป็นลำดับชั้น หรือที่เรียกว่า Hierarchy

โดยวงนอกสุดจะเริ่มจากการที่มี <html> ครุ่ม 2 tag อื่นอยู่คือ

1. <head> หน้าที่เก็บข้อมูลที่สำคัญและแสดงชื่อ Website (Title)

โดยจะประกอบไปด้วย tag จำนวนมาก ตัวอย่างเช่น <meta>, <title>, <viewport> รวมทั้งเป็นส่วนที่ใช้เขียน CSS ใน <style>

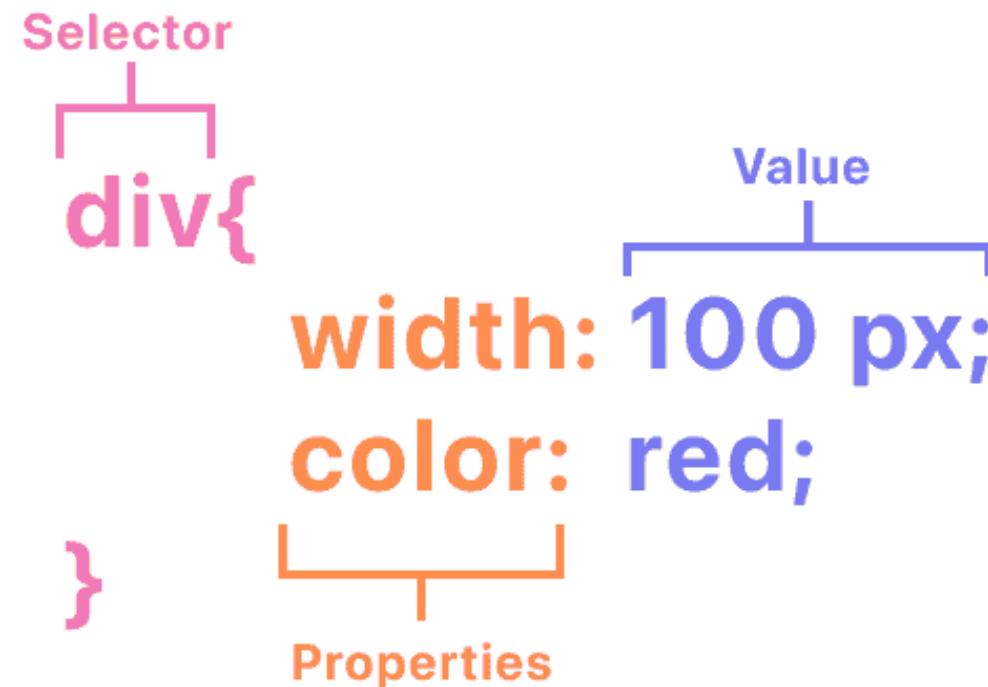
2. <body> หน้าที่แสดงข้อมูลเนื้อหาให้แก่ผู้ใช้ Website โดยจะประกอบไปด้วย tag จำนวนมาก ตัวอย่างเช่น <h1>, <p>, <input> รวมทั้งเป็นส่วนที่ใช้เขียน JavaScript ใน <script>

CSS selector คือ

สำหรับการเริ่มเขียน HTML ควรจะรู้ [CSS](#) เล็กน้อยมาก่อน เพื่อให้สามารถใช้งานบาง Attribute ได้ เราจะเขียน [Selector](#) ใน `<style>` เช่นอน



Selector structure



CSS Selector คือ การเลือก HTML element เฉพาะเจาะจงเพื่อการตกลงด้วย **CSS** โดยจะประกอบ 3 องคประกอบหลักคือ

1. **Selector** = ใช้กำหนด HTML element ที่จะทำการตกลง

2. **Properties** = ส่วนของการตกลง ตัวอย่าง width คือ กำหนดความกว้างของ Element

3. **Value** = ค่าที่เรากำหนดขึ้นมา ตัวอย่าง width : 100px คือ Element นี้จะมีความกว้าง 100 pixel

Selector มีหลายประเภทมาก วันนี้แอดจะมาแนะนำ 5 simple selector ที่นิยมใช้กัน

Simple Selector

	CSS	HTML
Element	p{...}	<p>...<p/>
Class	.fst{...}	<p class="fst">...<p/>
ID	#fst{...}	<p id="fst">...<p/>
Universal	*{...}	All element
Group	h1, p{...}	<h1>...<h1/> <p>...<p/>

CSS selector

Simple Selector

	CSS	HTML
--	-----	------

Element	p{...}	<p>...<p/>
----------------	--------	------------

Class	.fst{...}	<p class="fst">...<p/>
--------------	-----------	------------------------

ID	#fst{...}	<p id="fst">...<p/>
-----------	-----------	---------------------

Universal	*{...}	All element
------------------	--------	-------------

Group	h1, p{...}	<h1>...<h1/> <p>...<p/>
--------------	------------	-------------------------

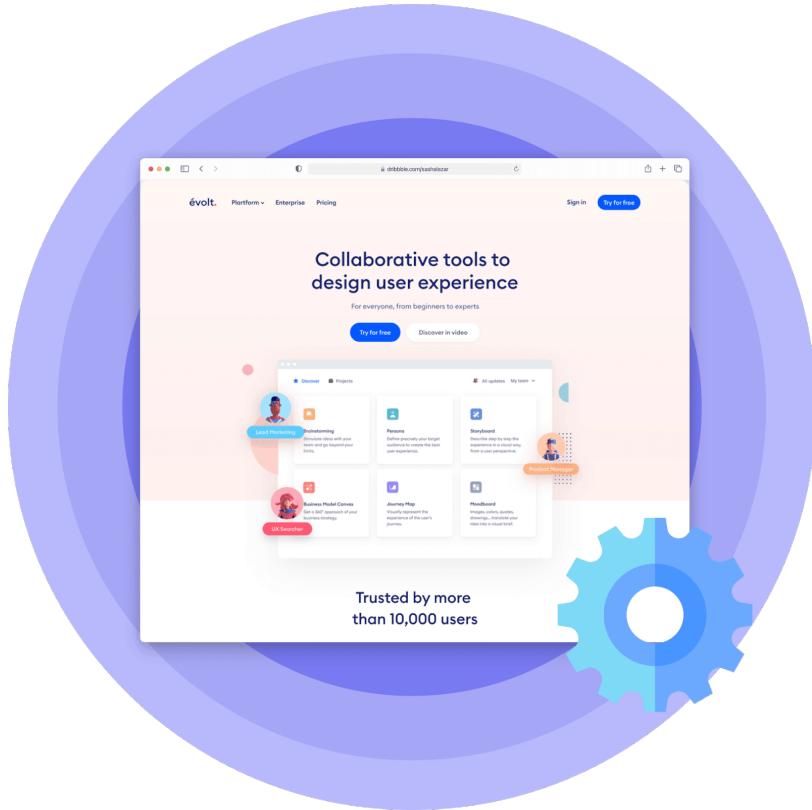
1. Element ใช้เลือก HTML element ตัวอย่างเช่น p{} เลือก <p>

2. Class ใช้เลือก element ที่ attribute ชนิด class โดยเขียน .Class

3. ID ใช้เลือก element ที่ attribute ชนิด id โดยเขียน #ID

4. Universal ใช้เลือกทุก element เราจะเขียนด้วยตัวดอกจันทร์(*)

5. Group ใช้เลือกหลาย element พร้อมกัน เช่น <h1> กับ <p> จะเขียน h1,p



02. Website ทำงานยังไง กว่าจะมาเป็นหน้า Website ที่เราเห็นกันต้องผ่าน ขั้นตอนอะไรมาบ้าง

ในบทนี้จะพูดถึงการทำงานของ Website ว่า กว่าจะมีหน้าแต่ละหน้าให้เราได้ใช้กัน ต้องผ่านอะไรมาบ้าง

Client และ Server คืออะไร ส่งข้อมูลอะไรให้กัน วันนี้เราจะมาทำความตื้อกันครับ

URL คืออะไร

ก่อนที่จะเข้าใจการทำงาน เราต้องมารู้จัก URL(Uniform Resource Location) กันซะก่อน URL คือที่อยู่ website บนโลกอินเทอร์เน็ต เปรียบเทียบได้กับ ที่อยู่บ้านของเรานั่นเอง



โดยจะประกอบไปด้วย

1. Protocol คือ ภาษาที่ใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์
2. Sub domain เป็นลำดับการตั้งชื่อ ซึ่งรองลงมากจาก Domain name
3. Domain name คือชื่อหลักของ website ใชระบุ IP address
4. Path เป็นเหมือนเส้นทางไปยังข้อมูลต่างๆ เช่น รูปภาพ, วิดีโอ

คำศัพท์ที่ควรรู้

1.TCP/IP คือ Protocol ที่นำไปใช้ในการส่งข้อมูลบนโลกอินเทอร์เน็ต

2.DNS คือ ระบบเก็บข้อมูล domain name เมื่อเพื่อนๆ คนหน้า website ตัว Browser จะเข้าไปหา web address ที่ DNS

3.Client คือ เครื่องมือที่ใช้ในการเชื่อมต่ออินเทอร์เน็ต เช่น Firefox, Chrome [browser](#)

4.Server คือ ระบบคอมพิวเตอร์ที่เก็บข้อมูลหน้า website มีหน้าที่รอรับ Respond และส่งข้อมูล web ไปที่ Client

5.HTTP คือ Protocol ที่ Website ใช้ในการคุยกัน

HTTP ต่างกับ HTTPS อย่างไร

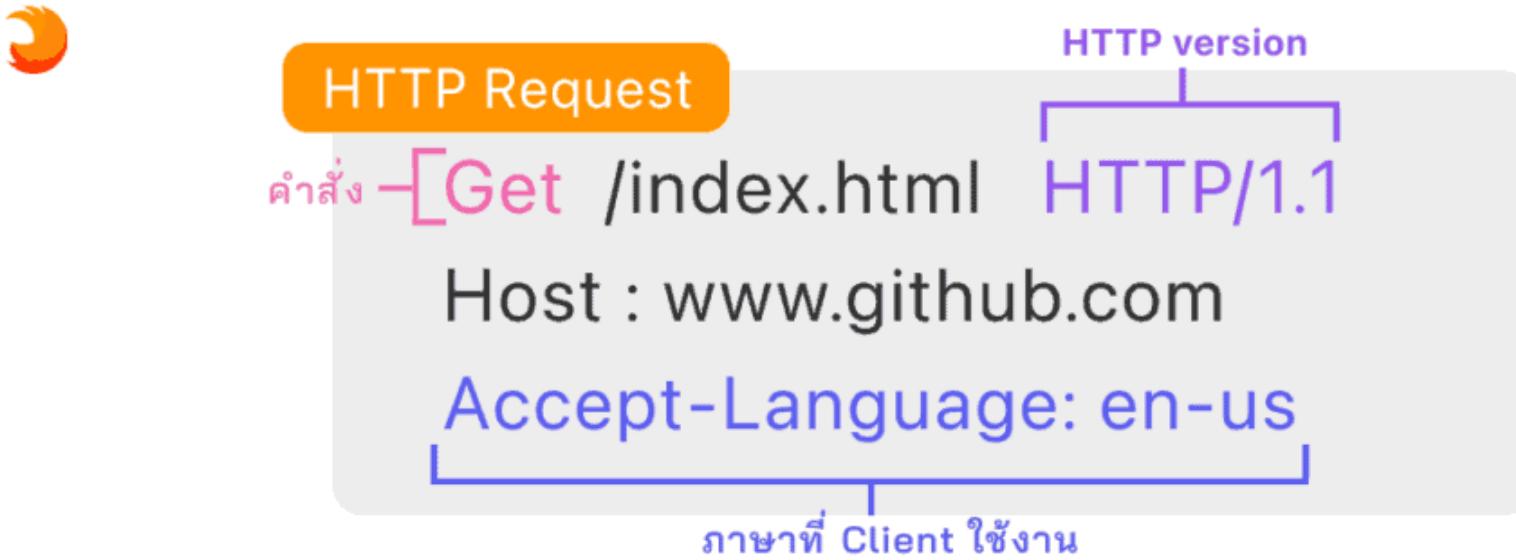
HTTP และ HTTPS ต่างเป็น Protocol ที่มีหน้าที่ในการส่งข้อมูลบน website แต่สิ่งทำให้ HTTPS แตกต่างจาก HTTP คือการมีในส่วน Encryption ทำการส่งข้อมูลมีความปลอดภัยที่มากกว่า HTTPS ใช้ในการส่งข้อมูลจำพวก Password, Credit card เป็นต้น



การทำงานของ Website

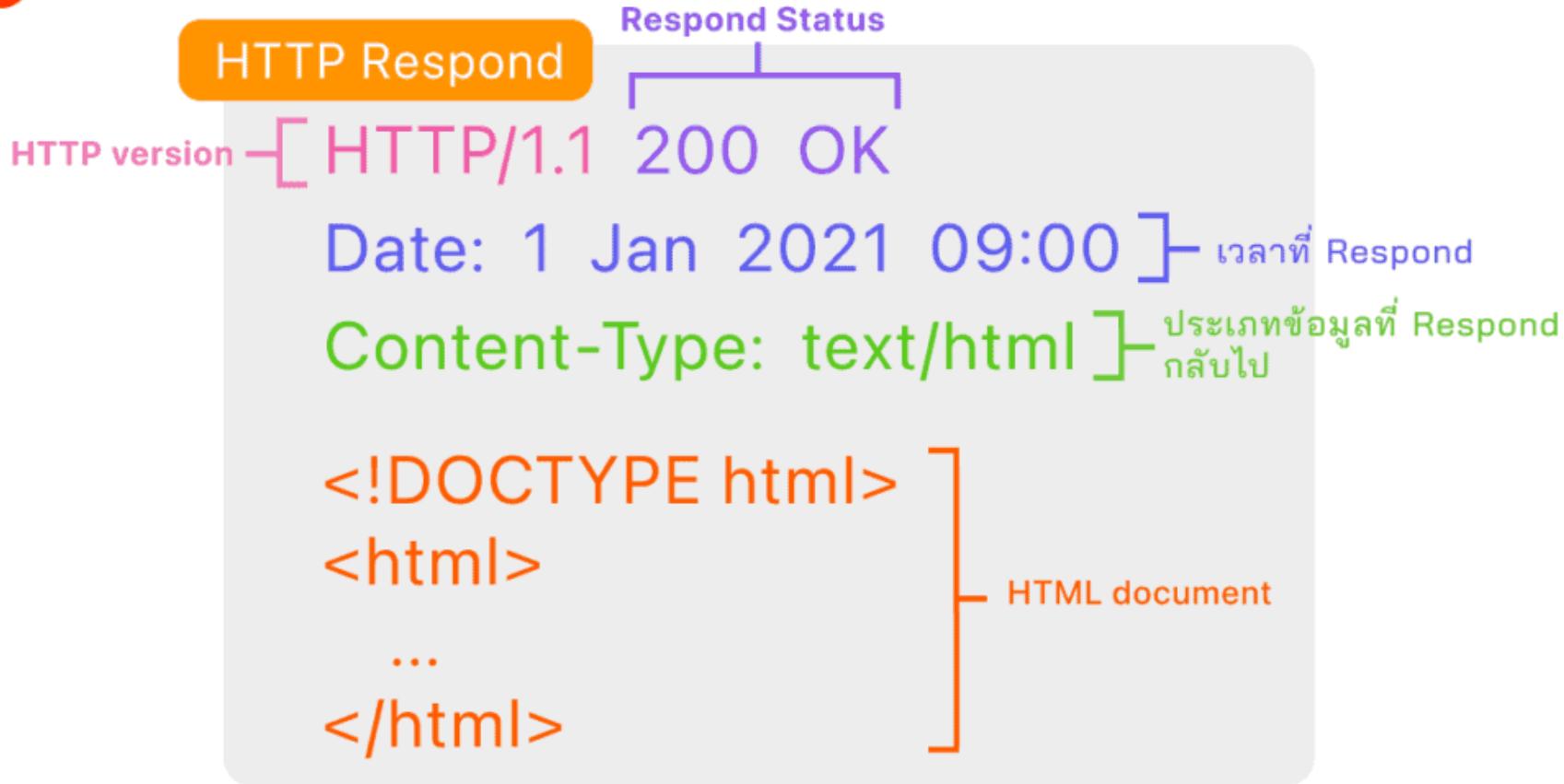
มีขั้นตอนดังนี้ :

1. การทำงานของ Website จะเริ่มจากที่เราค้นหา URL ใน Browser ก่อน
2. หลังจากนั้นตัว Browser หรือ Client จะทำการส่ง HTTP Request ไปที่ Server เพื่อรับขอข้อมูลหน้า Website



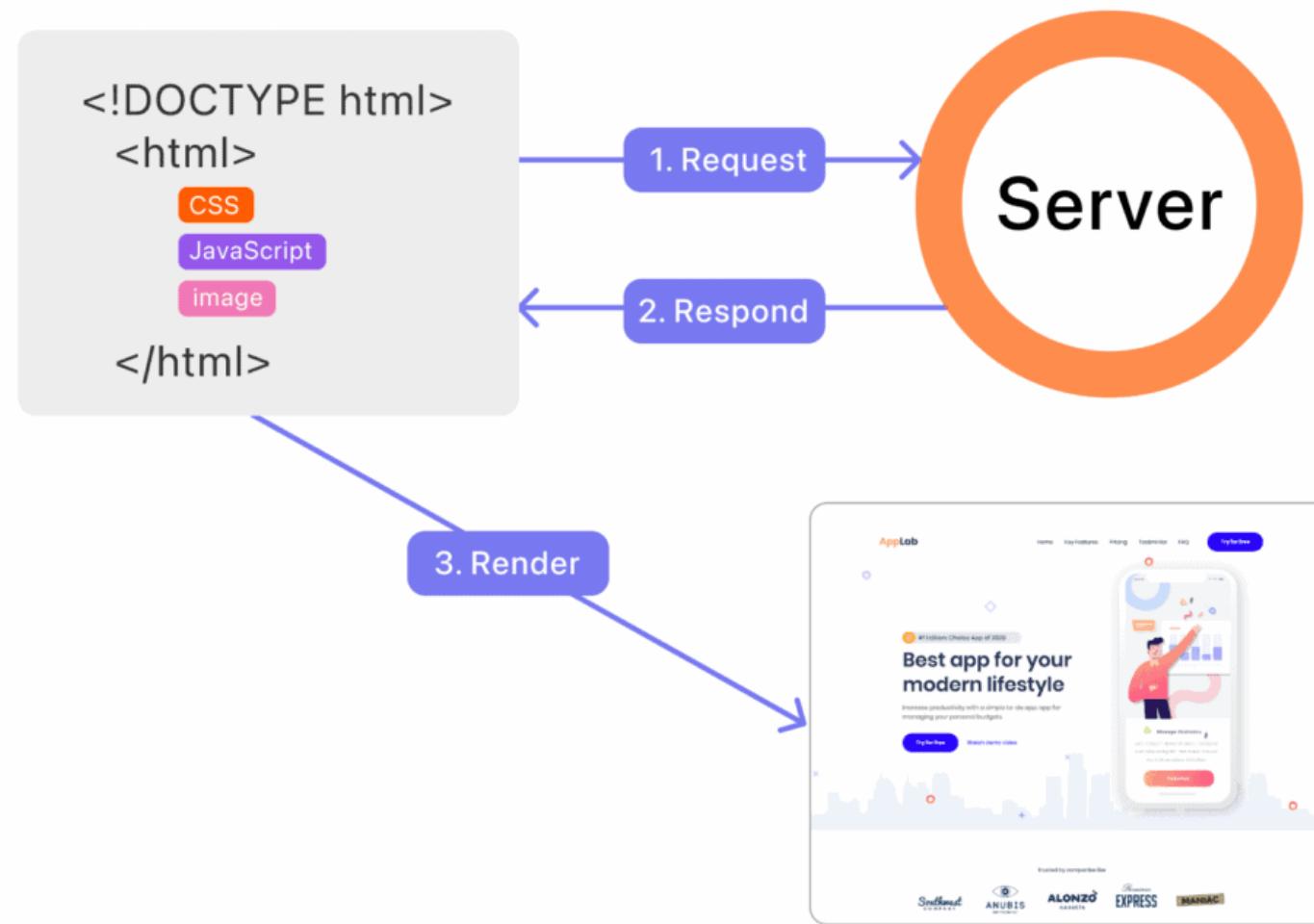
3. Server ได้รับ Request และทำการค้นหา Website

- ถ้ามี Website ตัว Server จะทำการเตรียมส่ง HTTP Respond กลับไปยัง Client
- ถ้าไม่มี Website ที่รองขอ จะส่ง HTTP 404 error page ไปยัง Client แทน อย่างที่เราเห็นกันเป็นประจำเวลาไม่เจอหน้า Web



4, Client ได้รับ Respond และทำการตรวจสอบ HTML document ตั้งแต่ตัวหัวจรดเท้า เพื่อหา file อื่นๆ เช่น CSS file, รูปภาพ, วิดีโอ และทำการส่ง HTML request ในแต่ละ file พร้อมกันไปยัง Server อีกทีซึ่งเรียกว่า DOM ทำให้องค์ประกอบของ Website จะปรากฏขึ้นบน Browser เร็วมาก

5. เมื่อ Client มี File เพียงพอแล้ว ก็จะแสดงผล Website บน Browser ทันที



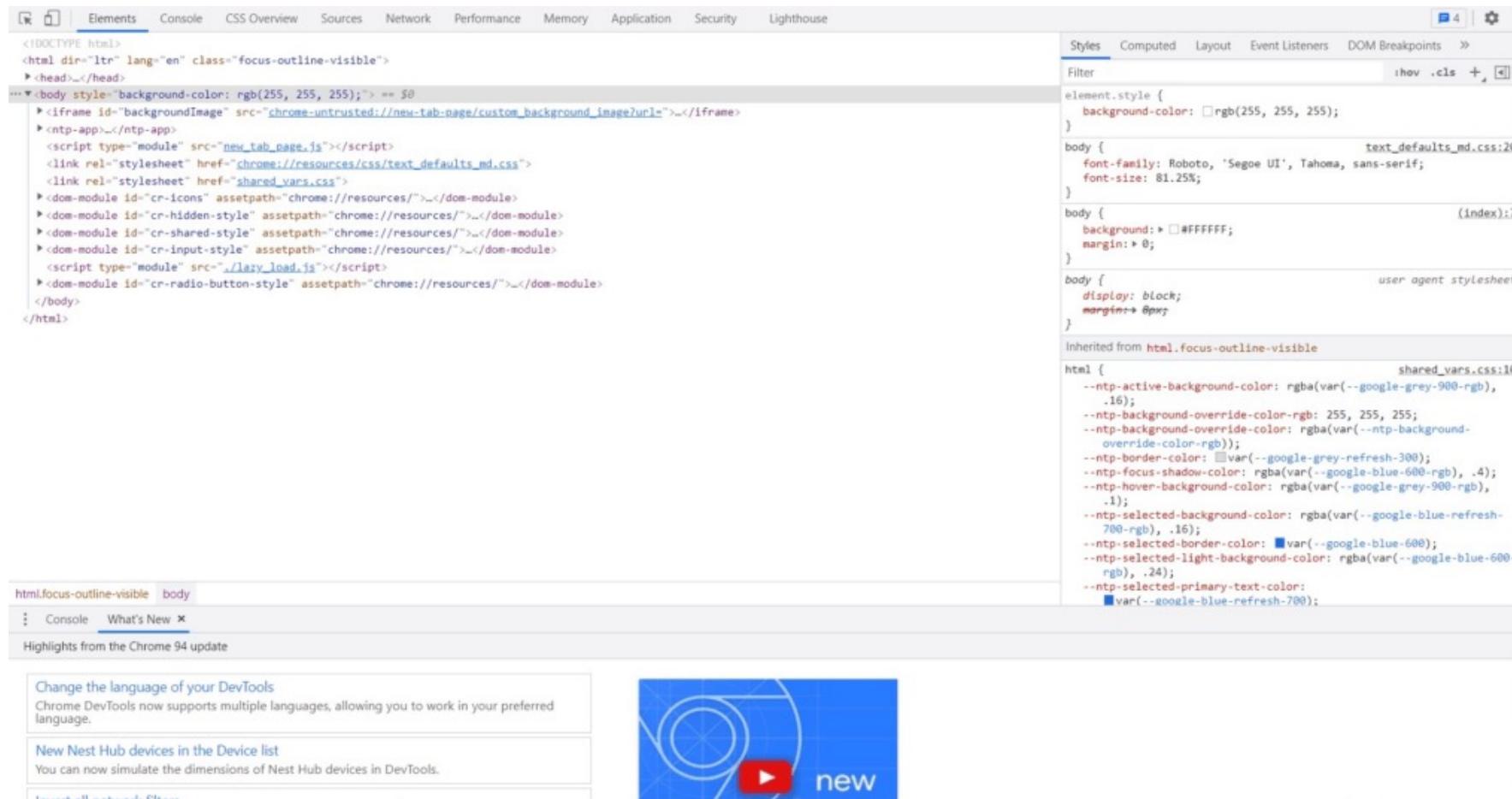


03. Chrome dev tool

การใช้ Chrome inspection ให้มีประสิทธิภาพ
สูงสุดในการพัฒนา Website

Chrome dev tool គីវោនា

Chrqme dev tool គីវោនាមួយទៀតដែលមានប្រព័ន្ធដែលបានរចនាបានសម្រាប់ការអនុវត្តន៍ការងារនៃ Website ឬ Browser ដូចមាន Mozilla dev tool និង Chrome dev tool ដែលមានប្រព័ន្ធជាមួយគ្នា។



วิธีเปิด Chrome dev tool

การเปิด Chrome dev tool มี 4 วิธี

1. คลิกขวาที่หน้า website -> คลิก Inspect

2. กด F12(Windows)

3. กด ที่ icon จุด 3 จุด ทางด้านขวาบนสุดของ Browser -> เลือก More tools -> คลิก Developer tools

4. กด Command + Option + I (MacOS)

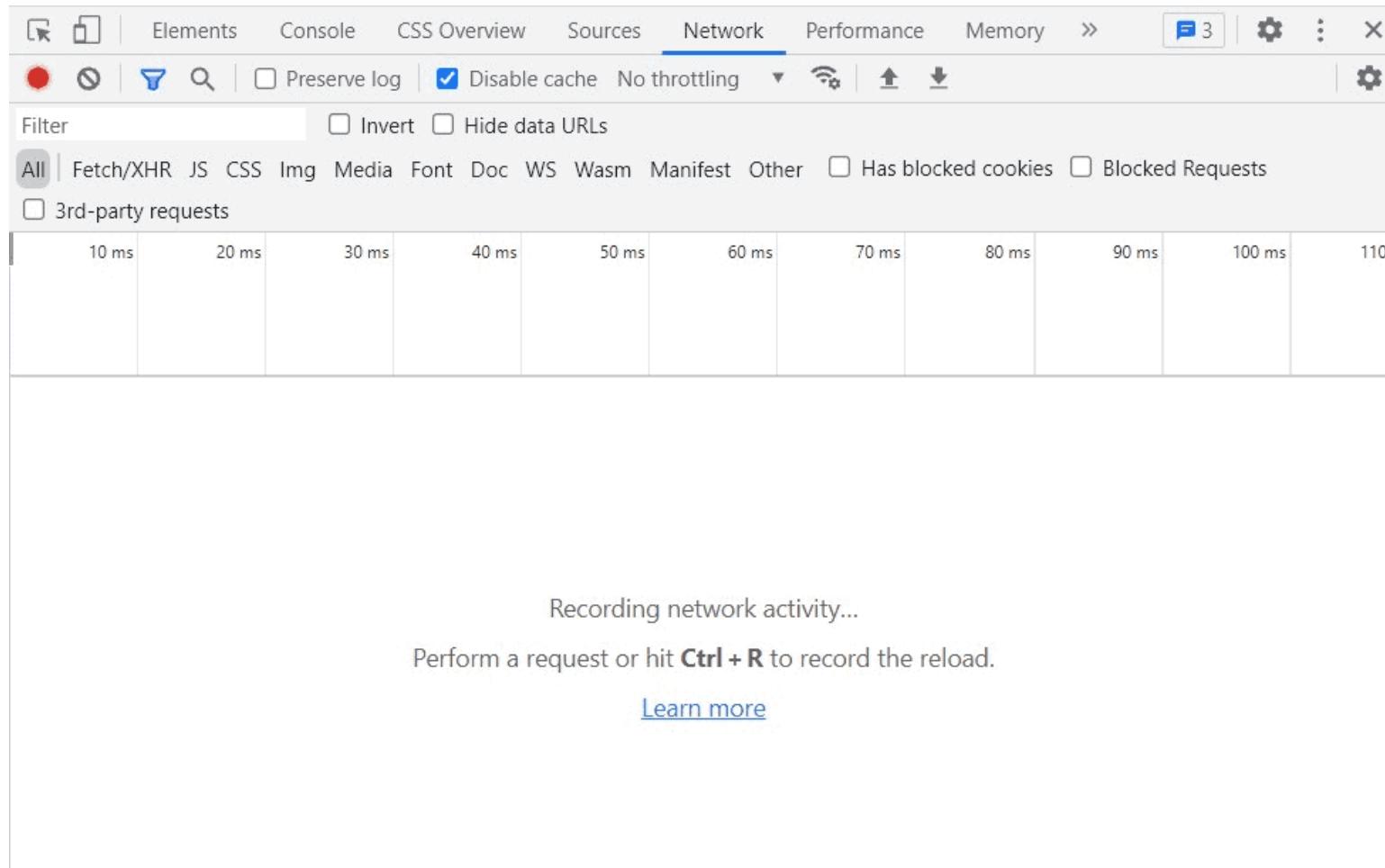
เพื่อนๆ สามารถปรับตำแหน่งของ Chrome dev tool ได้ด้วยการคลิกปุ่ม 3 จุด -> เลือก Dock side ขนาดต่างๆ

การใช้งาน Chrome dev tool

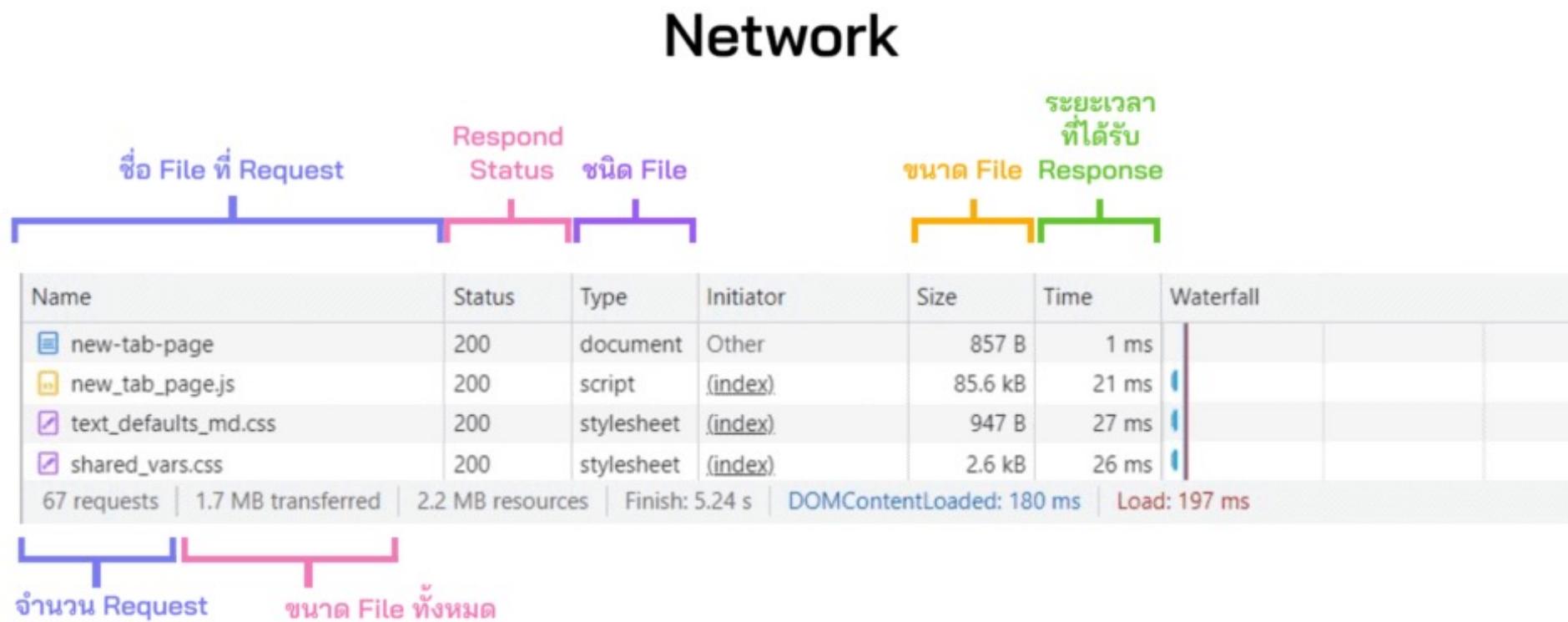
1. Network

Tab แรกที่จะพูดถึงกันคือ Network

นักพัฒนาจะใช้ Tab นี้ในการตรวจสอบ HTTP request และ Respond ที่เราได้เรียนในบทที่แล้ว



ถ้ายังไม่มีค่าว่าอะไรปรากฏขึ้นมา เพื่อนๆสามารถกด Refresh หน้า Website
แล้วตารางของลางจะปรากฏขึ้นใน Network ขึ้นมา ซึ่งจะมีข้อมูลที่แสดงผลออกตามนี้



2. Element

นักพัฒนาใช้เพื่อตรวจสอบ HTML document รวมทั้งทดลองปรับแก้ไข code ในส่วนต่างๆ

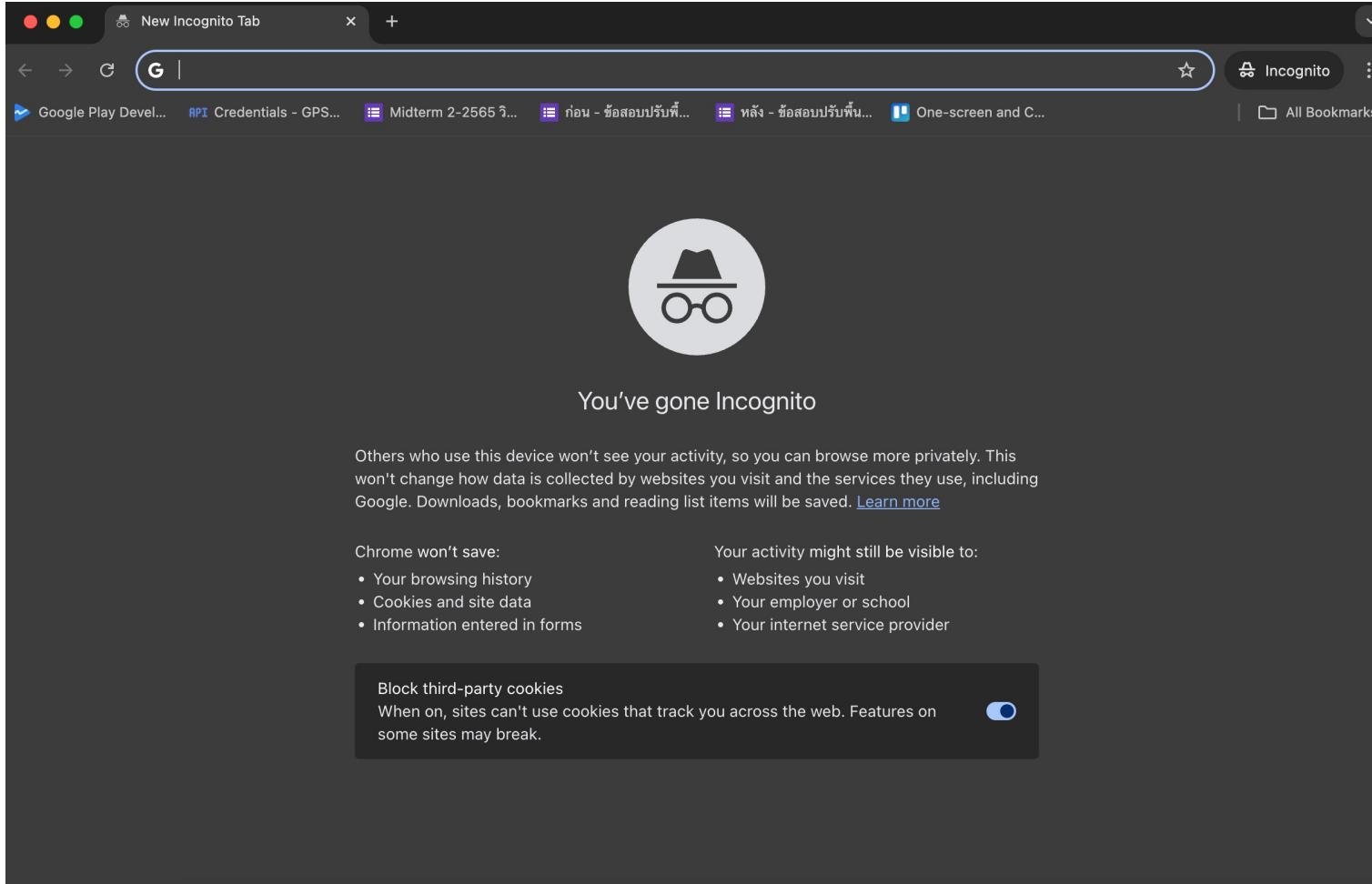
The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. The left pane displays the DOM tree of the current page, starting with the <html> element. The right pane shows the CSS styles applied to the elements, organized by source: 'text_defaults_md.css:20', '(index):7', and 'user agent stylesheet'. The 'text_defaults_md.css:20' section contains rules for the body element, including background-color: #FFFFFF and font-family: Roboto, 'Segoe UI', Tahoma, sans-serif. The '(index):7' section contains a rule for the body element with background: #FFFFFF and margin: 0px. The 'user agent stylesheet' section contains a rule for the body element with display: block and margin: 8px. The bottom status bar indicates the element being inspected is 'html.focus-outline-visible'.

```
<!DOCTYPE html>
<html dir="ltr" lang="en" class="focus-outline-visible">
  <head>...</head>
  ...<body style="background-color: rgb(255, 255, 255);"> == $0
    ><iframe id="backgroundImage" src="chrome-untrusted://new-tab-page/custom_background_image?url">...
    </iframe>
    ><ntp-app>...</ntp-app>
      <script type="module" src="new_tab_page.js">
      </script>
      <link rel="stylesheet" href="chrome://resources/css/text_defaults_md.css">
      <link rel="stylesheet" href="shared_vars.css">
    ><dom-module id="cr-icons" assetpath="chrome://resources/">...</dom-module>
    ><dom-module id="cr-hidden-style" assetpath="chrome://resources/">...</dom-module>
    ><dom-module id="cr-shared-style" assetpath="chrome://resources/">...</dom-module>
    ><dom-module id="cr-input-style" assetpath="chrome://resources/">...</dom-module>
      <script type="module" src="./lazy_load.js">
      </script>
    ><dom-module id="cr-radio-button-style" assetpath="chrome://resources/">...</dom-module>
  </body>
</html>
```

html.focus-outline-visible | body

3. โหมดไม่ระบุตัวตน(Incognito Mode)

Incognito Mode หรือ โหมดไม่ระบุตัวตน ชี้อีกแบบอย่างว่าเป็นโหมดไม่ระบุตัวตน ซึ่งนั่นก็หมายความว่าเป็น การท่องเว็บโดยไม่ระบุตัวตนผู้ใช้งานนั้นเอง เมื่อคุณเปิดไปใช้งานในโหมดนี้ หน้าต่างที่เราเปิดจะไม่เก็บประวัติการของ เว็บ คุกกี้ ข้อมูลเว็บไซต์ ข้อมูลที่เรากรอกลงบนแบบฟอร์มผ่านทางหน้าเว็บ หรือข้อมูลที่เราทำการ Login เข้าสู่ ระบบตามหน้าเว็บไซต์ต่างๆ จะไม่ถูกเก็บ



The screenshot shows a Google Chrome browser window with an open DevTools panel. The main content area displays the 'You've gone Incognito' page, which includes a circular icon of a person wearing a hat and glasses, the title 'You've gone Incognito', a message about private browsing, and a 'Learn more' link. Below this, there are two sections: 'Chrome won't save the following information:' and 'Your activity might still be visible to:', each with a bulleted list. A 'Block third-party cookies' toggle switch is also present. The DevTools panel is open to the 'Elements' tab, showing the DOM structure of the page. The DOM tree includes nodes like <html>, <body>, <div class="content">, <div id="icon">, <h1>, <p id="subtitle">, and . The 'Styles' tab is selected in the DevTools sidebar, showing styles for elements like 'div.content' and 'body'. The 'Elements' tab also displays the CSS for these elements.

```
<!DOCTYPE html>
<html dir="ltr" hascustombackground="false" lang="en" class="md">
  <head>...
  <body>
    <div class="content">
      <div class="icon" role="presentation" alt="Incognito mode icon"></div>
      <h1>You've gone Incognito</h1>
      <p id="subtitle">...
        <span>...</span> == $0
        <a class="learn-more-button" href="https://support.google.com/chrome/?p=incognito">Learn more</a>
      </p>
      <div id="bulletpoints-wrapper">...</div> (flex)
      <div id="cookie-controls">...</div> (flex)
        <a class="learn-more-button" href="https://support.google.com/chrome/?p=incognito">Learn more</a>
      </div>
      <script type="module">...</script>
      <!-- Lazy-load cr_elements to avoid performance penalty introduced by loading Polymer -->
      <script type="module" src="chrome://resources/cr_elements/cr_toggle/cr_toggle.m.js" async>
      </script>
      <script type="module" src="chrome://resources/cr_elements/policy/cr_tooltip_icon.m.js" async>
    </div>
  </body>
</html>
```

04. VS code

Editor ที่มาแรงที่สุดในปัจจุบัน กับ Extension ที่จะทำให้การ Dev ง่ายขึ้น



Visual studio code คือ [Visual studio code](#) คือ โปรแกรมที่ใช้สำหรับเขียน Code ถูกพัฒนาขึ้น จาก บริษัท Microsoft ซึ่งมี Feature ที่จะอำนวยความสะดวกแก่ Programmer อย่างเรา ทำให้ในที่สุดกลายเป็น Editor ตัวนึงที่ได้รับความนิยมสูงในปัจจุบัน

ตัว VS code รองรับทั้ง 3 ระบบได้แก่

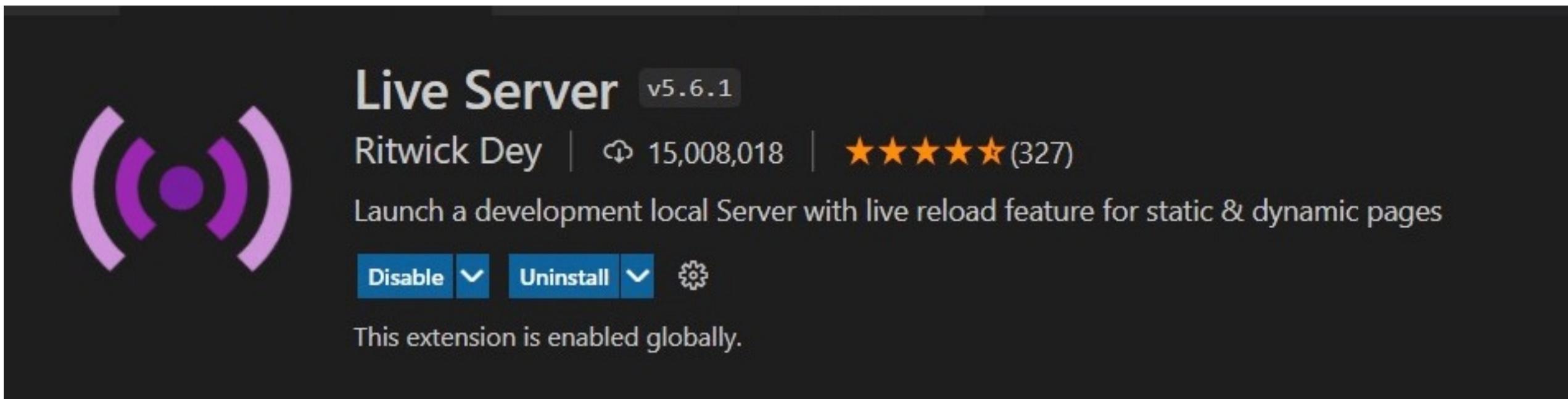
- Windows
- Linux
- MacOS

VS code extension

[Extension](#) คือส่วนเสริมของ [Visual studio code](#) ที่จะช่วยให้การใช้งานของพากเราง่ายขึ้น และ productive เราสามารถคนหาและติดตั้ง [Extension](#) ได้ในส่วน Navbar ของ VS code หรือกด (Ctrl+Shift+X)

ในการเริ่มพัฒนา พื้นฐาน HTML จำเป็นต้องใช้ 3 extension นี้มาก

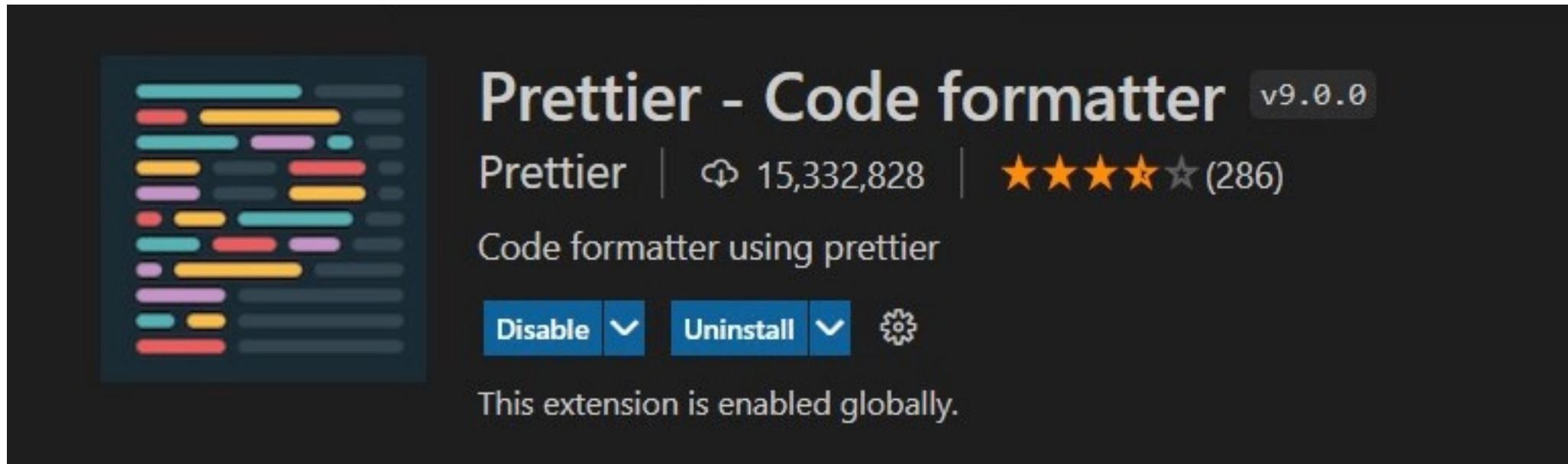
1. [Live Server](#)



เป็น [Extension](#) ที่จะช่วยให้เราสามารถ launch ตัว local [Server](#) ได้อย่างง่ายดาย หรือให้พูดง่ายๆ คือสามารถสร้าง Web,page ได้ใน server ของตัวเอง เราสามารถอัพเดต code ที่เปลี่ยนไปได้แบบ real-time เพียงกด save หรือ Ctrl + s ใน VS code

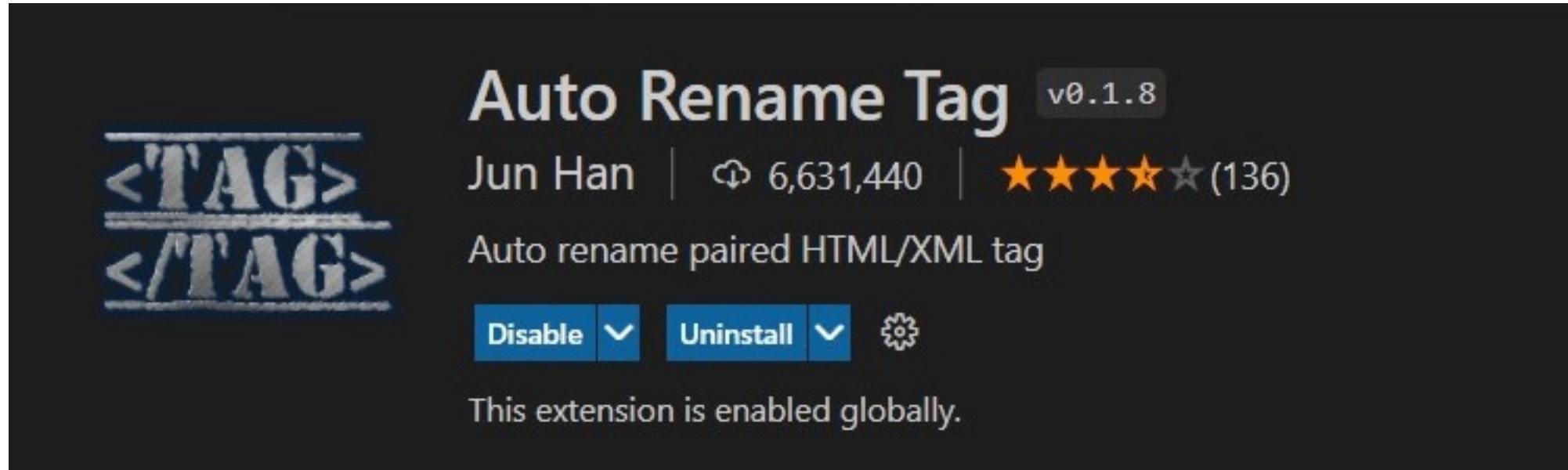
2. [Prettier](#)

เป็น [Extension](#)ช่วยจัดรูปแบบ Code ให้อ่านง่ายขึ้น



3. Auto Rename Tag

Extension ที่ทำให้เราสามารถแก้ไข Start tag และ End tag ใน HTML element ได้แบบอัตโนมัติ



index.html - HTML - Visual Studio Code

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
    <div>
    </div>
  </body>
</html>
```

Ln 6, Col 30 Spaces: 4 UTF-8 CRLF HTML ⚡ Port : 5500 ✅ Prettier ⌂ ⌂



05. พื้นฐาน HTML

เริ่มเขียน HTML สร้าง Text, Image, link รวมทั้งการ
ตกแต่งด้วย Style attribute

HTML Element ที่ควรรู้

1. Heading

Heading tag ใช้ในการแสดงหัวข้อเนื้อหาต่างๆ มีอยู่ด้วยกัน 6 ระดับ ตั้งแต Heading 1 ไป Heading 6 ซึ่งจะเรียงไปตามขนาดและความสำคัญ จะเขียนอยู่ในรูป <hx>...</hx>

The screenshot shows a CodePen interface with two main sections: 'HTML' and 'Result'. The 'HTML' section contains the following code:

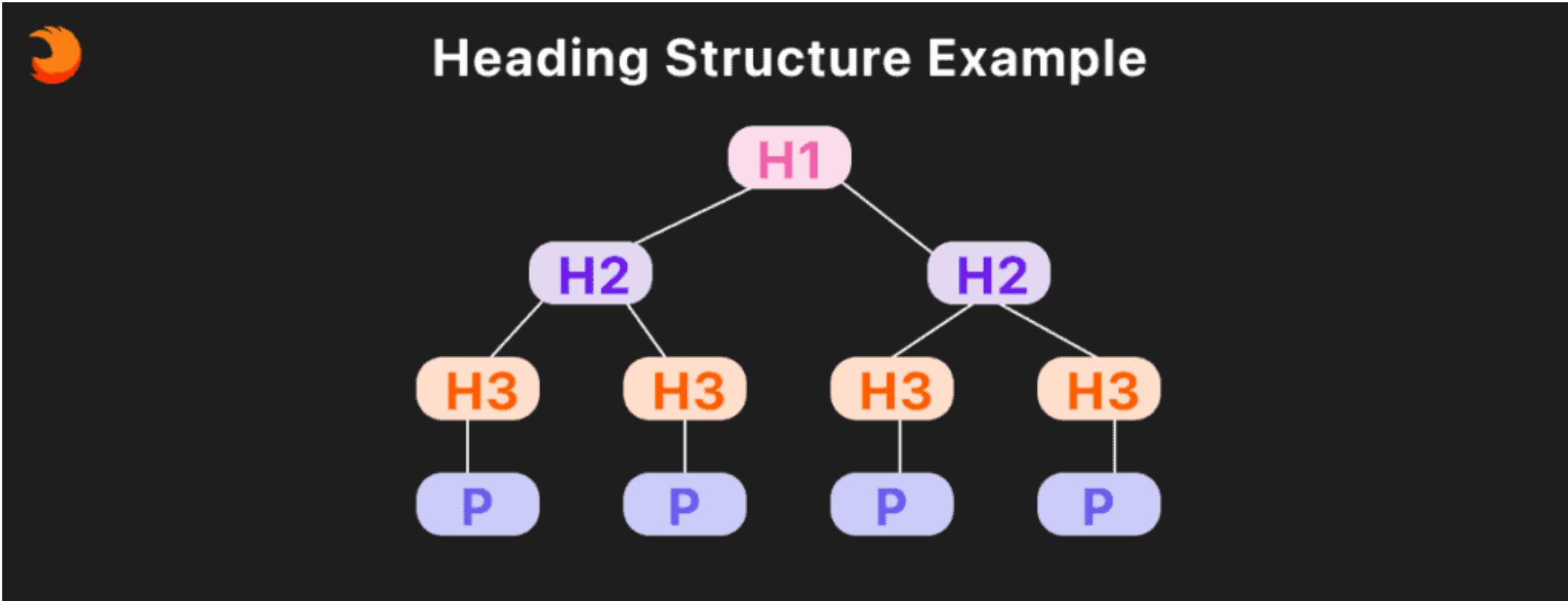
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
<body>
```

The 'Result' section displays the rendered output with four levels of headings:

- Heading 1**
- Heading 2**
- Heading 3**
- Heading 4**

At the bottom of the interface, there are buttons for 'Resources', '1x', '0.5x', '0.25x', 'Rerun', and 'EDIT ON CODEPEN'.

เนื้อหาใน Website ส่วนใหญ่จะมี Heading 1 แค่อันเดียว
และมีการวางโครงสร้างหัวข้อต่างๆ แบบลำดับขั้นตอนไม่ เรียงกันไป



2. Paragraph

Paragraph tag ใช้ในการแสดงเนื้อหาต่างๆ
จะเขียนอยู่ในรูป <p>...</p>

The screenshot shows a CodePen interface. On the left, under the 'HTML' tab, is the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
<body>
```

On the right, under the 'Result' tab, the word "Content" is displayed in a large, bold, black font.

3. Formatting

Formatting tags จะใช้ในการตกแต่งข้อความต่างๆ ซึ่งมีหลายรูปแบบให้เลือกใช้กัน

The screenshot shows a browser-based code editor interface. On the left, there's an 'HTML' tab with the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
```

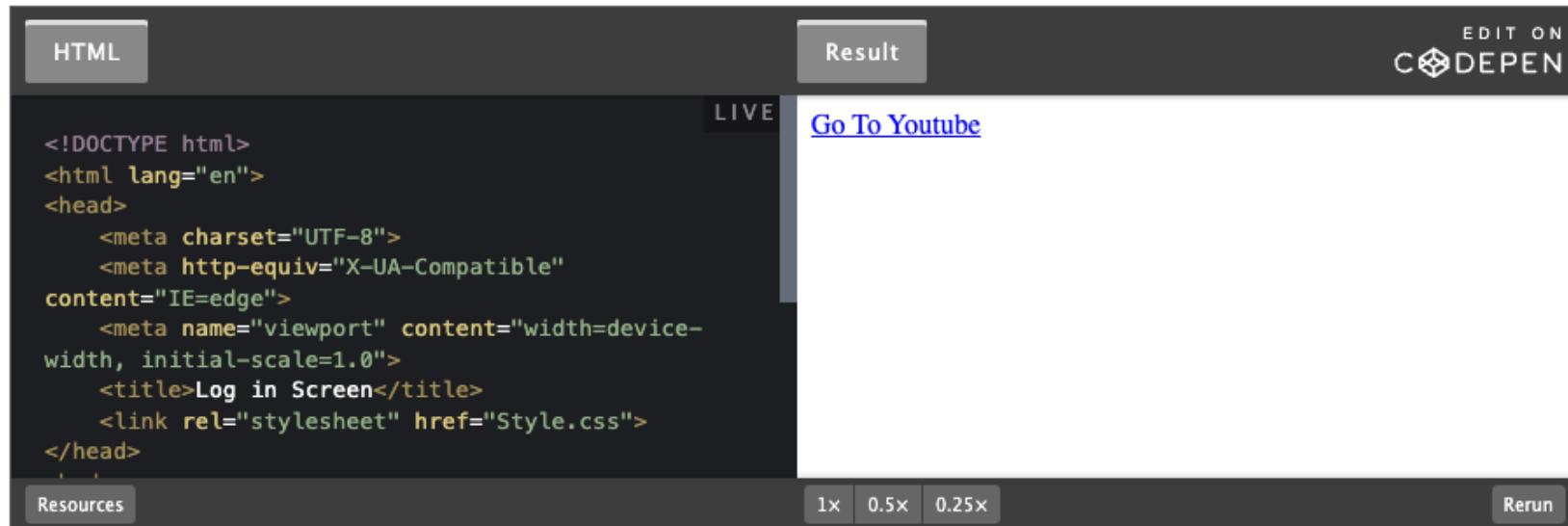
On the right, there's a 'Result' tab showing the output of the code. The word 'Designil' is repeated multiple times in different styles. The first four instances are in bold black font, the fifth is in yellow, the sixth is underlined, and the last two are in regular black font. There's also a small 'LIVE' button above the result area.

At the bottom, there are buttons for 'Resources', '1x', '0.5x', '0.25x', and 'Rerun'. In the top right corner, it says 'EDIT ON CODEPEN' with a small icon.

4. Hyperlink

Hyperlink tag ใช้ในการสร้าง link ให้กับองค์ประกอบต่างๆใน HTML document จะเขียนอยู่ในรูป <a>... และใช้คู่กับ href attribute เสมอในการ link ไปที่ website

รวมทั้งยังสามารถ Link ไปยัง HTML document อื่นๆ หรือ component ต่างๆใน HTML document ได้อีกด้วย



The screenshot shows the CodePen interface with the following components:

- HTML Tab:** Contains the source code of an HTML document.
- Result Tab:** Shows the output of the code, which includes a blue underlined link labeled "Go To Youtube".
- LIVE:** A button indicating the code is live.
- CODEPEN:** The CodePen logo with the text "EDIT ON CODEPEN".
- Resources:** A button for managing resources.
- Scaling Buttons:** Buttons for scaling the result view at 1x, 0.5x, and 0.25x.
- Rerun:** A button to rerun the code.

The HTML code in the editor is:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
```

โดยเราทำการแก้ไขส่วน href จาก Website link เป็นชื่อไฟล์ HTML document หรือถ้าเป็นไฟลภาพที่อยู่ในโฟลเดอร์ ก็ต้องใส่ชื่อโฟลเดอร์ตามด้วย ” / “



Hyperlink tag

```
<a href="img/1.jpg"
```

```
<a href="Main.html">
```

สำหรับใครที่อยากรู้ว่า link ไปเปิดใน Tab browser ถัดไปเราสามารถใส่ target=" _blank" ใน tag ได้นะ

5. Image

ตรงตามชื่อ Image tag ที่ใช้แสดงผลรูปภาพ

ด้วยความที่เป็น Self-closing tag เลยเขียนอยู่ในรูป

และใช้ src attribute ในกิรุลีอกรปภามาแสดง และใช้ alt ในการบอกกับผู้ใช้ว่าไฟล์รูปภาพนี้คืออะไรในกรณีที่ไม่สามารถดูรูปภาพนั้นได้ (เน็ตชา)

alt attribute ถือเป็นสวนสำคัญของ image tag ที่ขาดไม่ได เพื่อแสดงให้ Web page เห็น

HTML

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Log in Screen</title>
<link rel="stylesheet" href="Style.css">
</head>
<body>
  
</body>
</html>
```

Result

LIVE



I design and develop experiences that make people's lives simple.

1x 0.5x 0.25x Rerun

6. Comment

Comment คือการอธิบาย Code ในกรณีที่มีคนอ่าน code ของเราหรือในกรณีที่เรากลับมาอ่าน แล้วจำไม่ได้ว่า code บurrทั้นใช่ทำอะไร ซึ่งตัว Compiler จะไม่นำ Comment ไปแสดงผล

Comment ถือเป็นสิ่งที่สำคัญมากในการทำงานพัฒนา Website รวมกัน เพื่อให้มีเข้าใจ Code ที่เราเขียนได้ โดยจะอยู่ในรูป <!--...-->

The screenshot shows a live coding interface. On the left, under the 'HTML' tab, is the following code:

```
LIVE
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Log in Screen</title>
<link rel="stylesheet" href="Style.css">
</head>
<body>
    <!--This is a Comment-->
    <h1>Heading</h1>
</body>
</html>
```

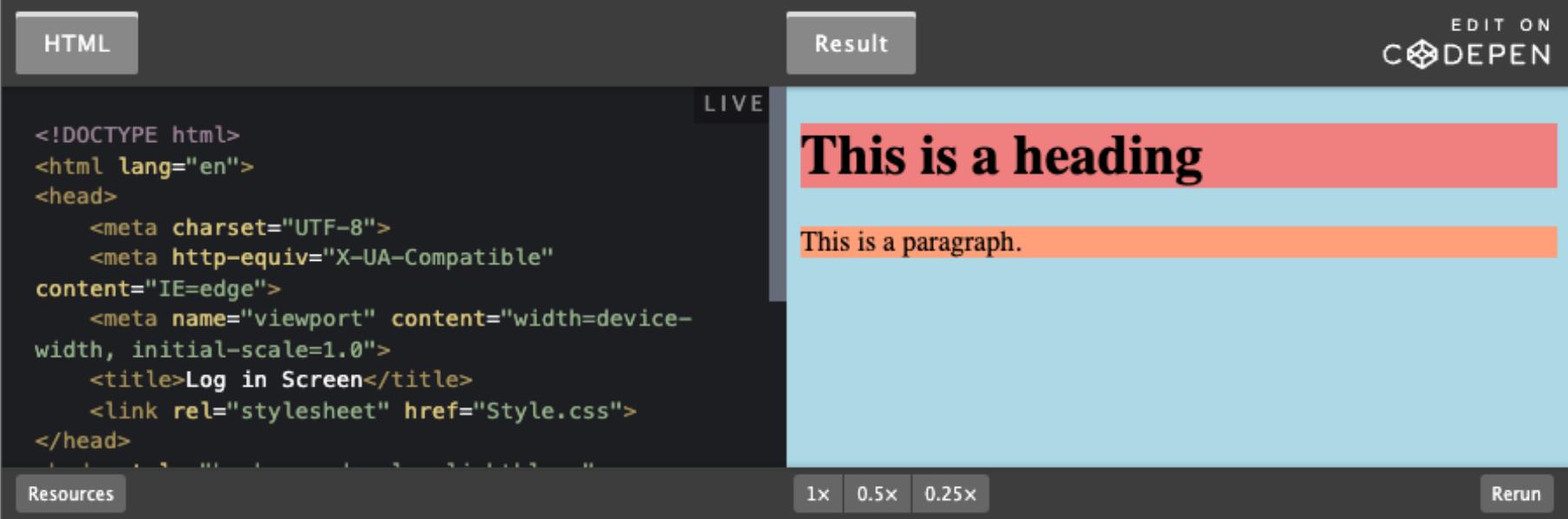
On the right, under the 'Result' tab, the output is displayed as a large bold heading:

Heading

HTML Style ที่ควรรู้
ส่วนนี้จะพูดถึงพื้นฐานการเขียน [CSS](#) เพื่อนำไปใช้ในการตกแต่ง HTML document ใน <style> นะ

1. Background color

เราใช้ background-color property ในการกำหนดสีพื้นหลัง HTML element



The screenshot shows a CodePen interface with the following components:

- HTML** tab: Contains the HTML code for a basic page structure.
- Result** tab: Shows the rendered output of the HTML.
- LIVE** button: Located between the tabs.
- EDIT ON CODEPEN** button: In the top right corner.
- Resources** button: At the bottom left.
- 1x 0.5x 0.25x** buttons: At the bottom center.
- Rerun** button: At the bottom right.

The rendered output shows two elements with different background colors:

- A heading element (**This is a heading**) with a red background.
- A paragraph element (**This is a paragraph.**) with an orange background.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
```

2. Text color

เราใช้ color property ในการกำหนดสีตัวอักษร

The screenshot shows a live code editor interface. On the left, the 'HTML' tab is active, displaying the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
```

On the right, the 'Result' tab is active, showing the rendered output. The heading is colored red, and the paragraph is colored blue.

LIVE

Result

EDIT ON
CODEPEN

This is a heading

This is a paragraph.

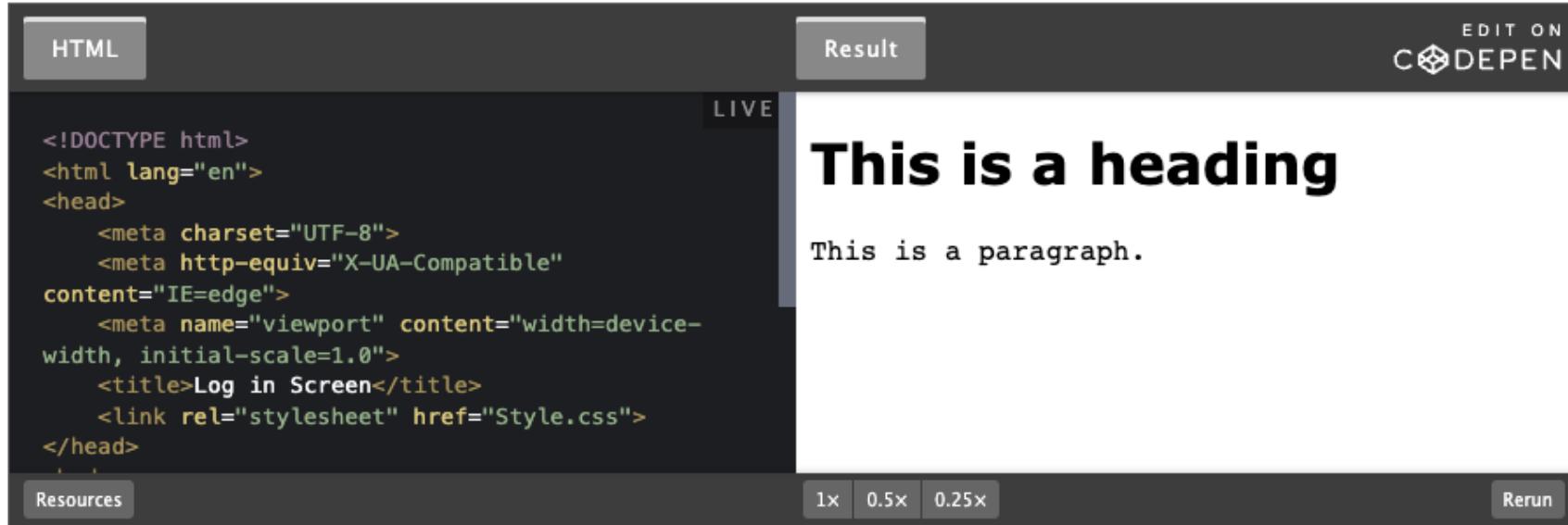
Resources

1x 0.5x 0.25x

Rerun

3. Font

เราใช้ font-family property ในการกำหนดประเภท font ที่ใช้



The screenshot shows a CodePen interface. On the left, under the "HTML" tab, is the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
```

Under the "Result" tab, the output is displayed:

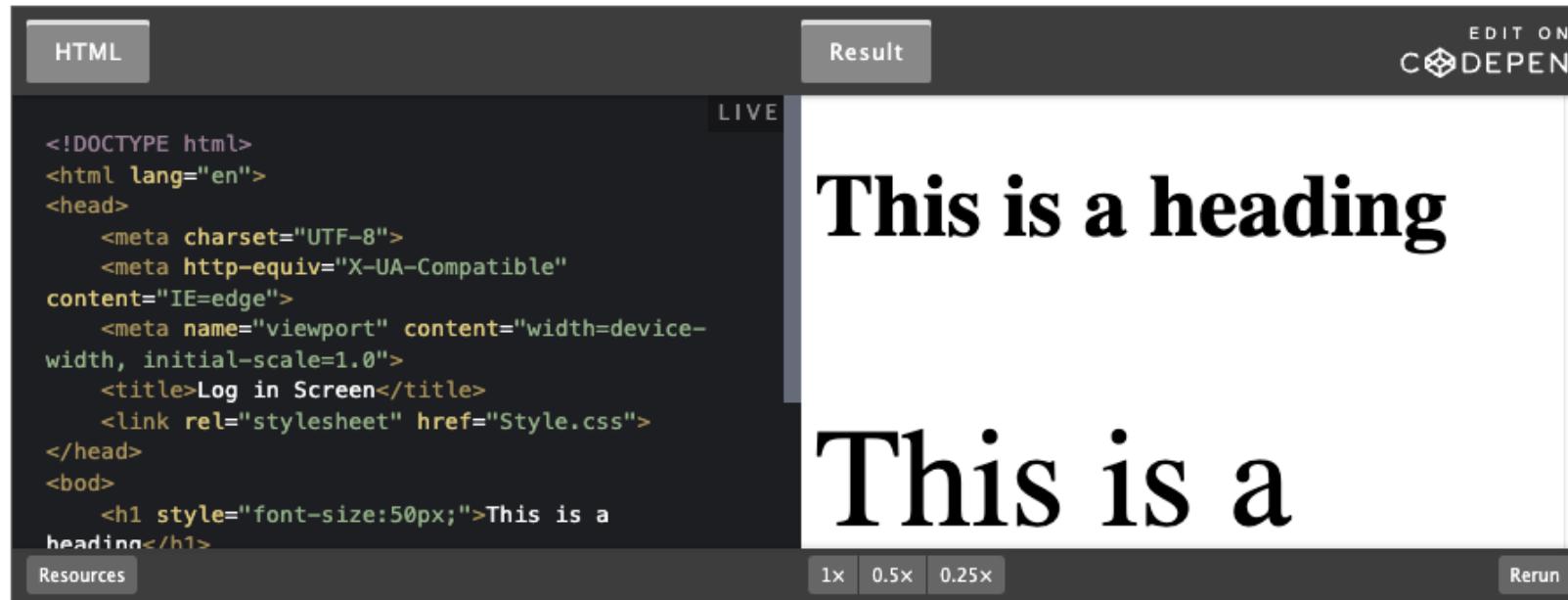
This is a heading

This is a paragraph.

At the bottom, there are buttons for "Resources", "1x", "0.5x", "0.25x", and "Rerun".

4. Text size

เราใช้ font-size property ในการกำหนดขนาดตัวอักษร



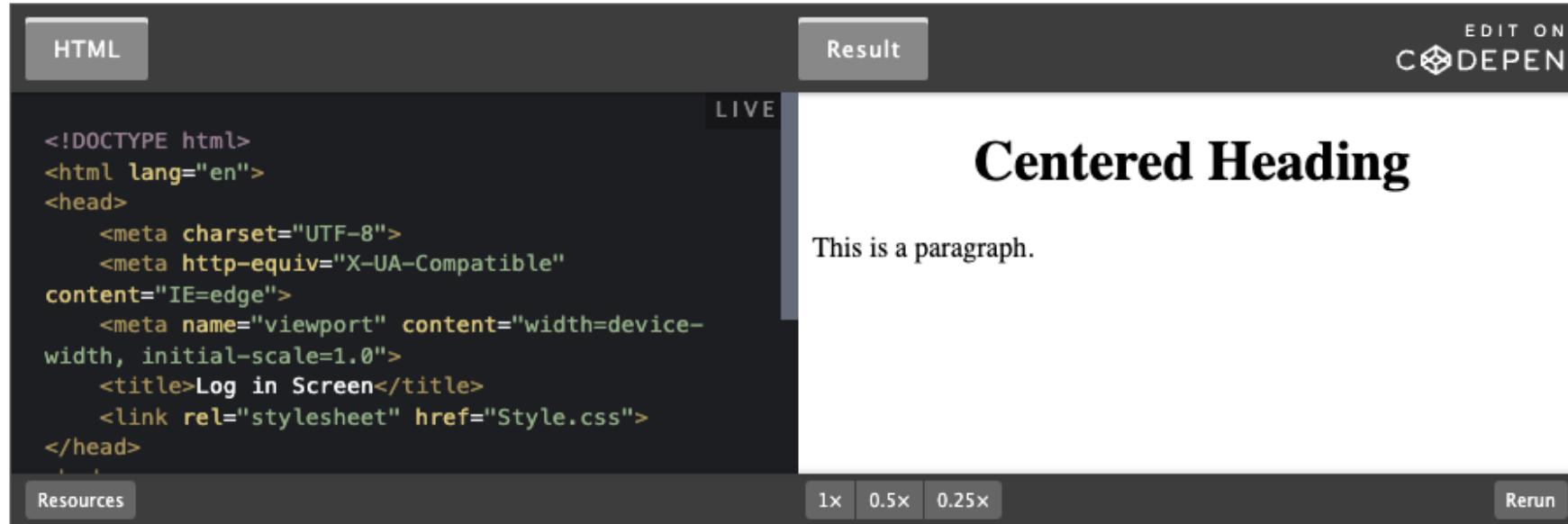
The screenshot shows a CodePen interface with the following components:

- HTML Tab:** Contains the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
<body>
  <h1 style="font-size:50px;">This is a heading</h1>
</body>
```
- Result Tab:** Shows the output of the code, displaying the text "This is a heading" in a large, bold, black font.
- LIVE Tab:** Located above the Result tab.
- EDIT ON CODEPEN:** A button in the top right corner.
- Controls:** At the bottom, there are buttons for "1x", "0.5x", "0.25x", "Rerun", and a "Resources" link.

5. Text alignment

เรายังใช้ text-align property ในการกำหนดตำแหน่งตัวอักษร บน HTML document



The screenshot shows a CodePen interface with the following components:

- HTML Tab:** Contains the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Log in Screen</title>
  <link rel="stylesheet" href="Style.css">
</head>
<body>
  <h1>Centered Heading</h1>
  <p>This is a paragraph.</p>
</body>
```
- LIVE Tab:** Shows the rendered result of the HTML code.
- Result Area:** Displays the rendered output:

Centered Heading

This is a paragraph.
- CodePen Logo:** EDIT ON CODEPEN
- Controls:** Resources, 1x, 0.5x, 0.25x, Rerun

Let's practice by your own