01. Intro to Web Programming

Ric Huang / NTUEE

(EE 3035) Web Programming

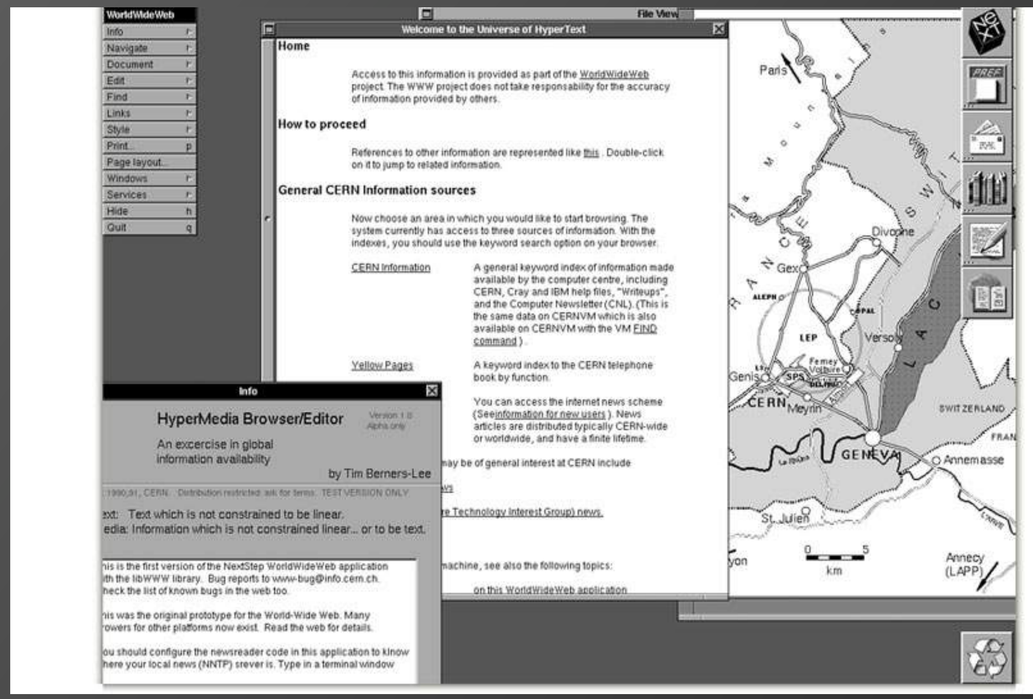
Web。Programming

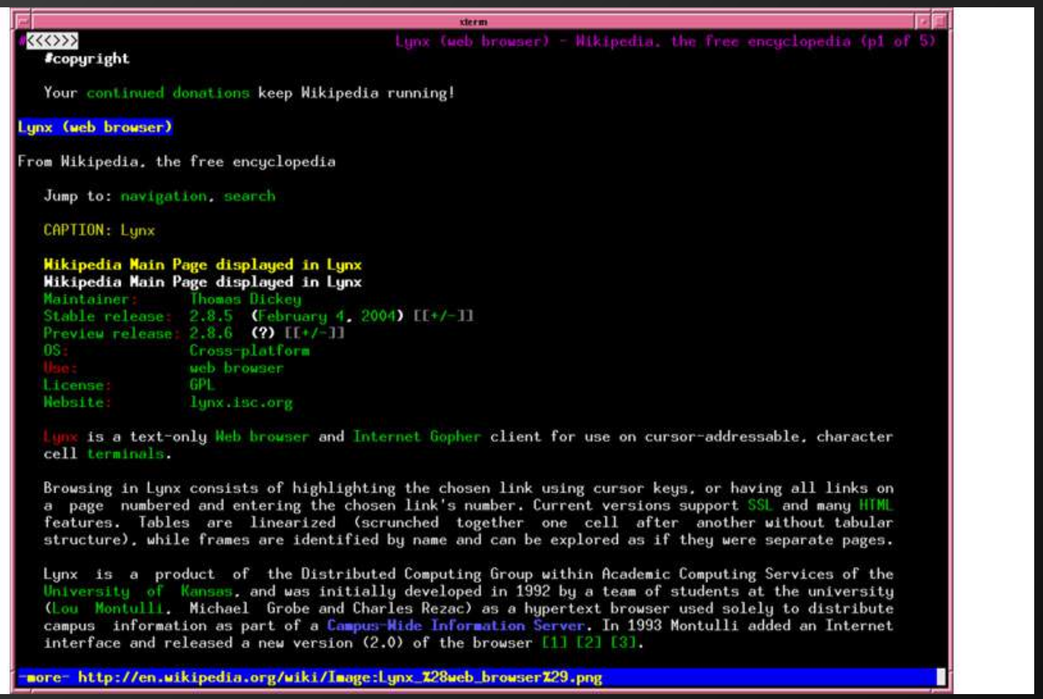
- 顧名思義，就是提供各式各樣「網路服務」的程式
- 對於使用者而言，可以透過瀏覽器(Web Browser)、APP、或是特定裝置上的軟體獲得服務
- 簡化起見，本課程將 focus 在透過瀏覽器所產生的網路服務

前端。後端

- 顧名思義，前端就是在前面利用各種介面(UI)設計、服務使用者的程式，而後端就是在後面負責各種(較為消耗資源的)運算與儲存、回傳給前端服務資料的程式
- 也許更精確的講法是：client-side (客戶端) vs. server-side (伺服器端)

前端。瀏覽器（Web Browser）



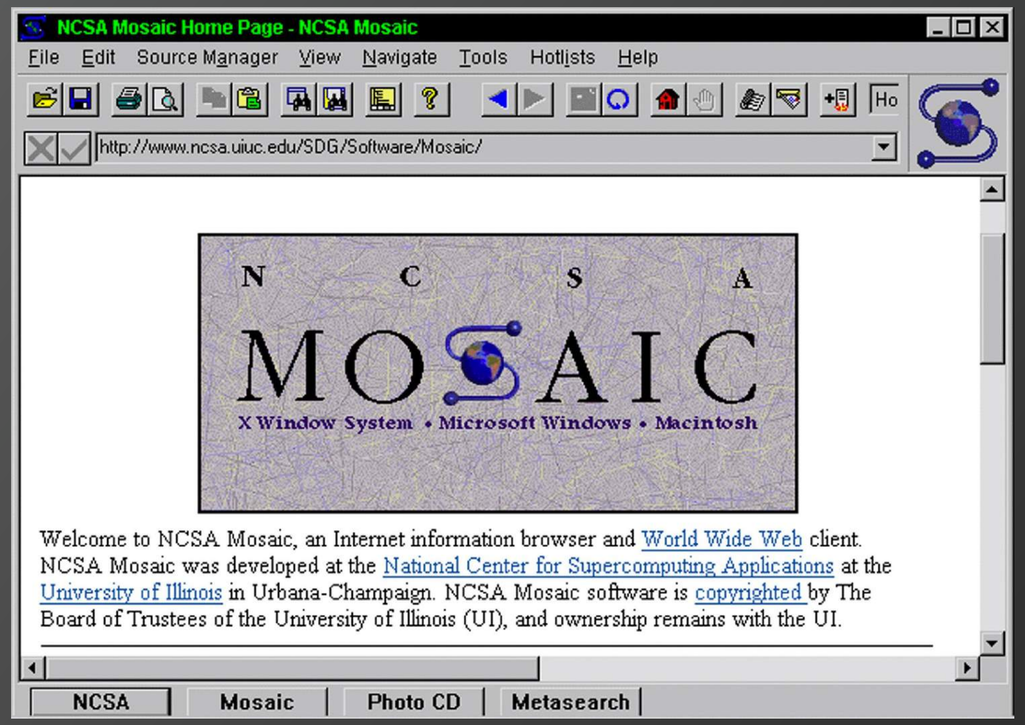


First web browser: WorldWideWeb (aka. Nexus), by Tim Berners Lee, 1990

Lynx, a text-based browser, by a team of students and staff at the university (of Kansas), 1992 (http://lynx.browser.org/)

Ref: https://www.telegraph.co.uk/technology/microsoft/11577364/Web-browsers-a-brief-history.html

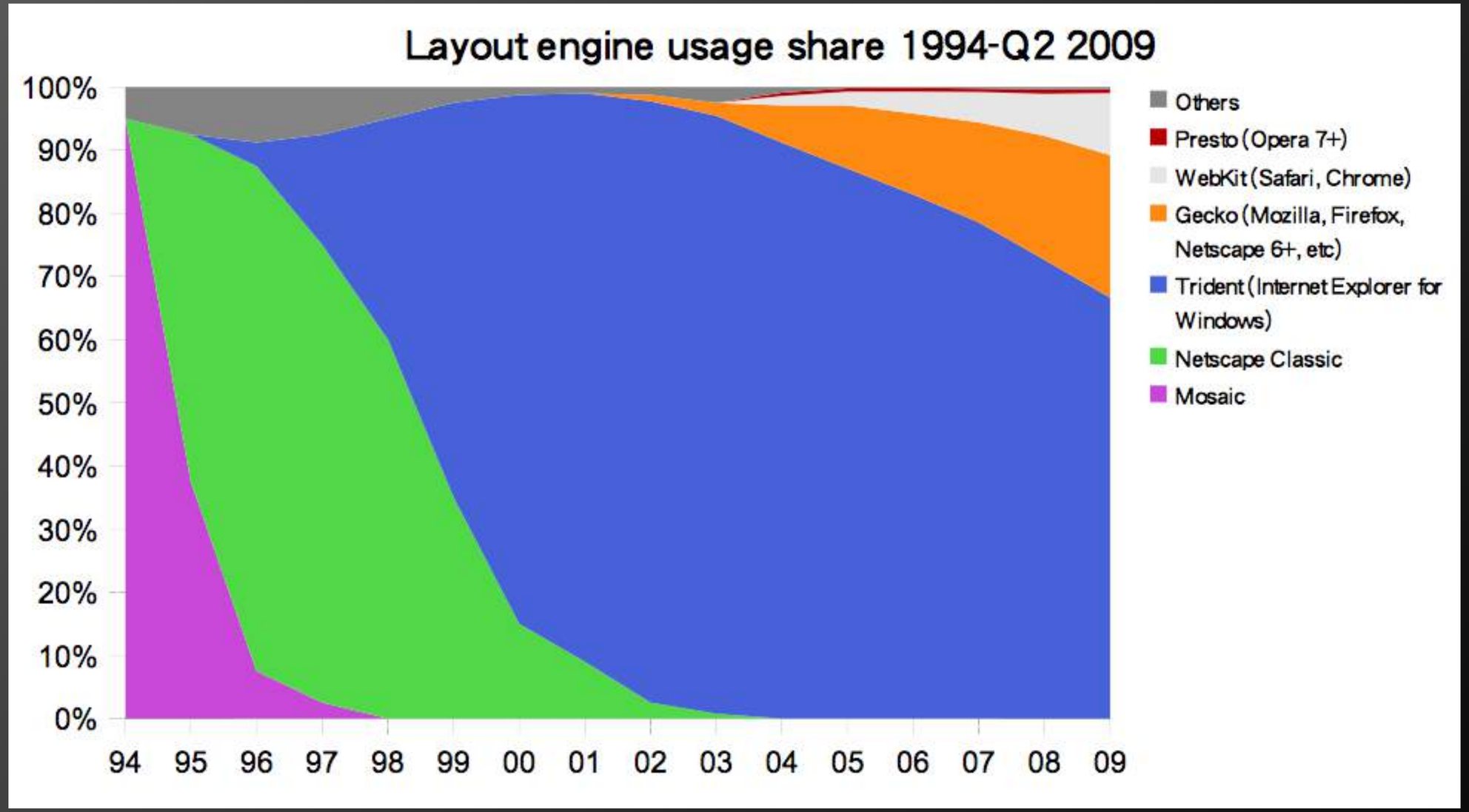
前端。瀏覽器（Web Browser）



Mosaic, often regarded as the first graphical web browser, was developed by National Center for Supercomputing Applications (NCSA) of UIUC, in late 1992 (Led by Marc Andreessen)

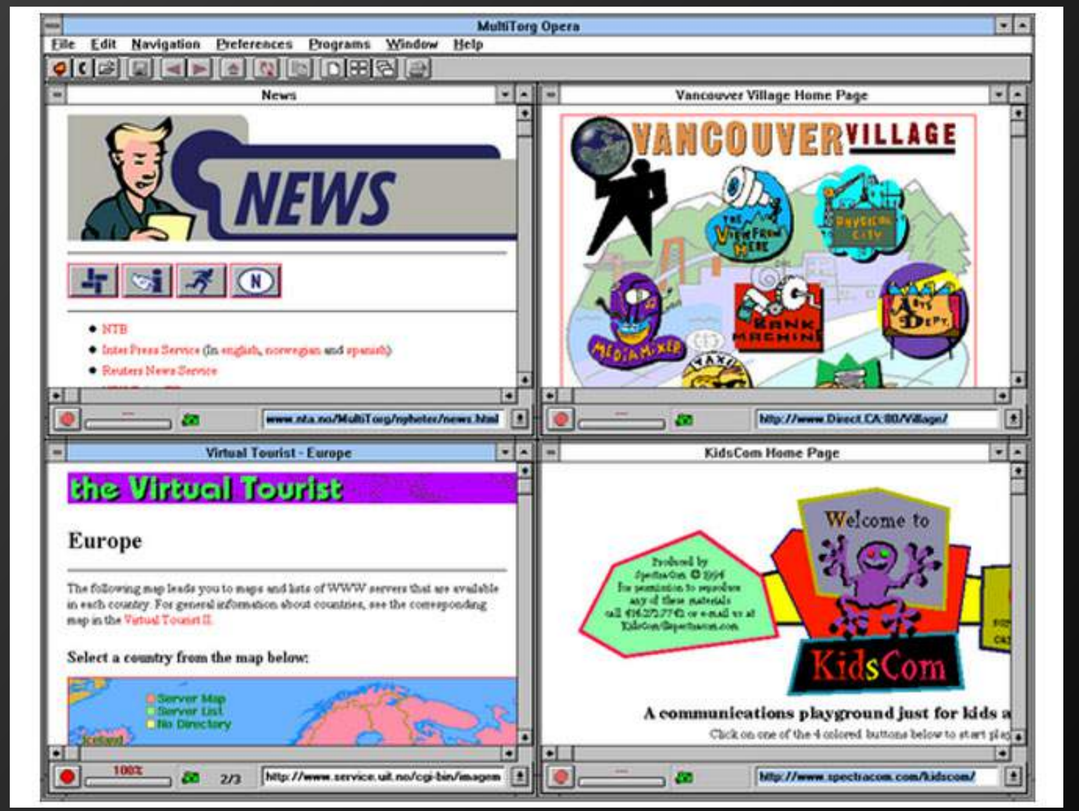
Netscape, founded by Marc Andreessen, James H. Clark, and 4 others from Mosaic, was the most successful commercial web company in 90s

前端。瀏覽器（Web Browser）



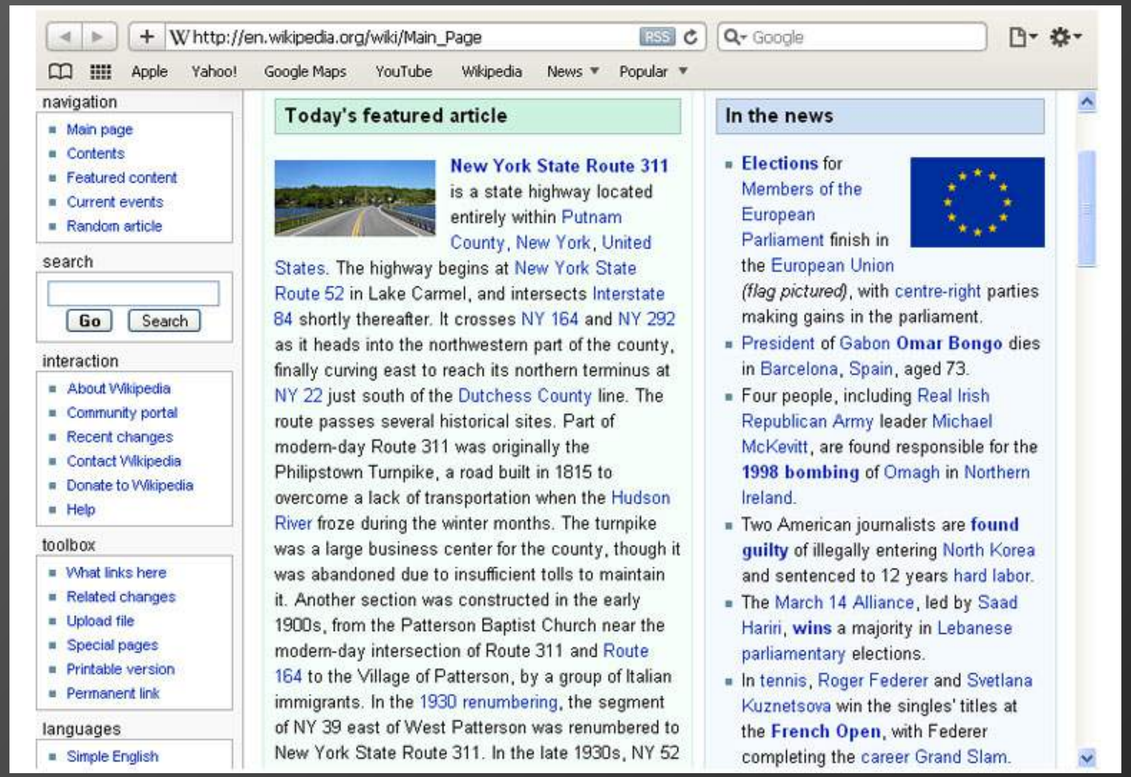
前端。瀏覽器（Web Browser）

Internet Explorer (IE), by Microsoft, in 1995

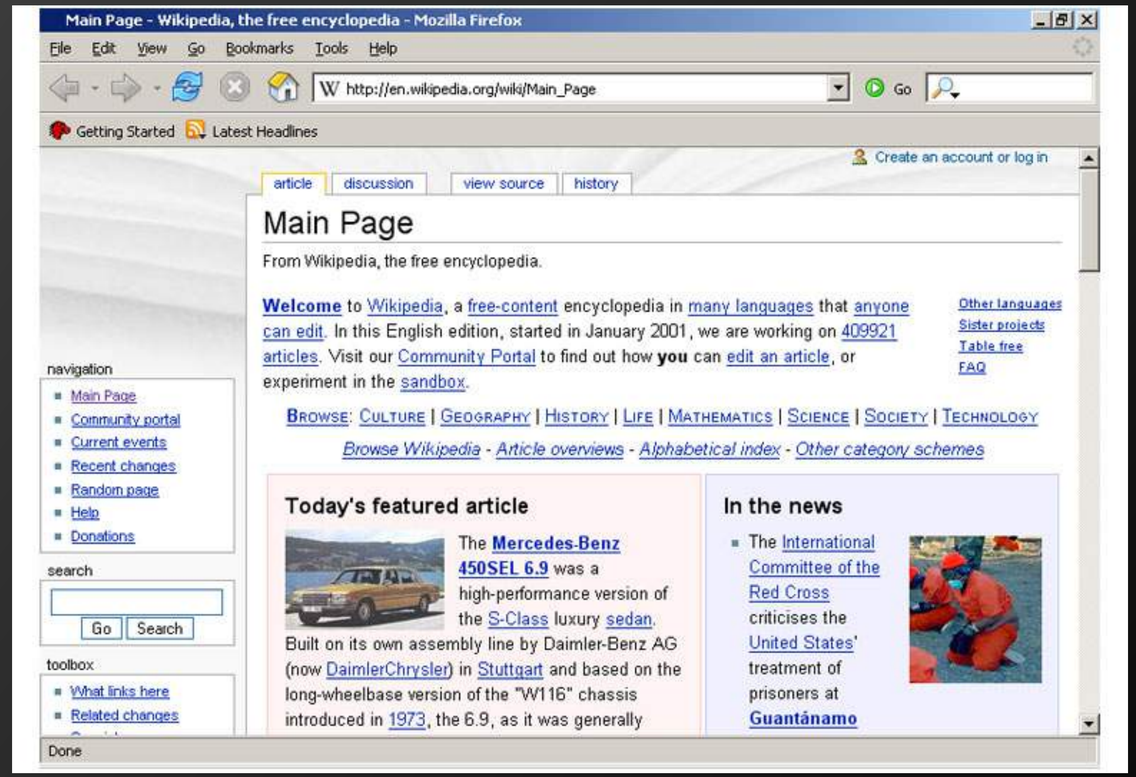


Opera, by Telenor, in 1996

前端。瀏覽器（Web Browser）



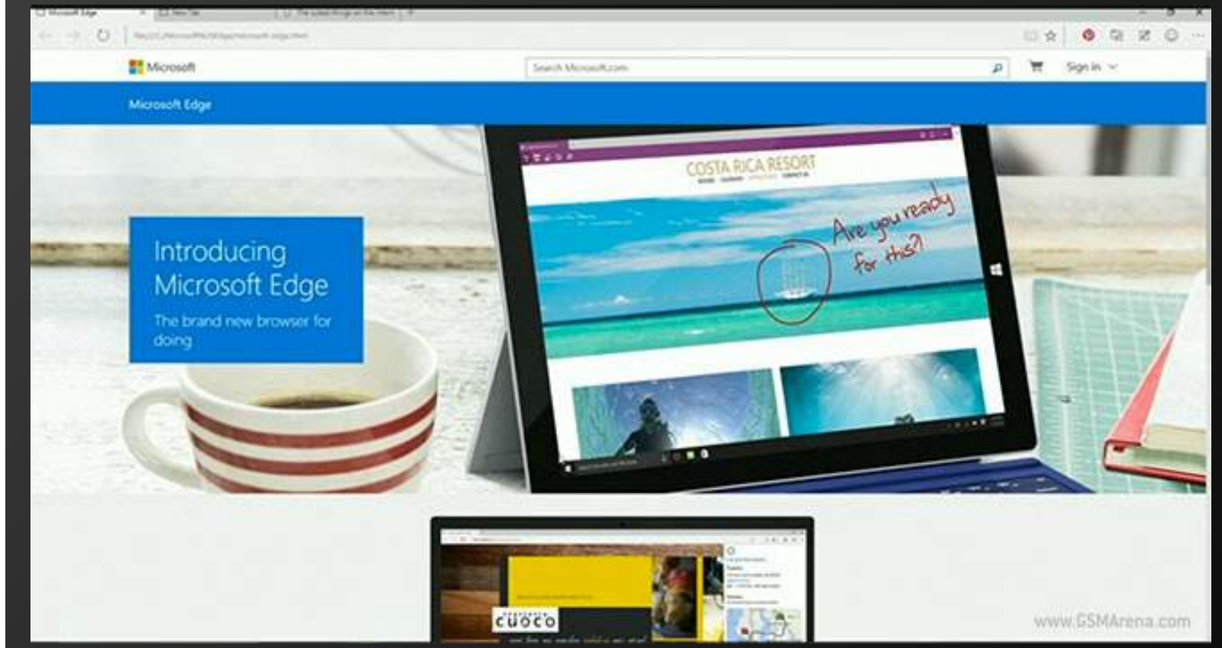
Safari, by Apple, in 2003



Firefox, by Mozilla, in 2004

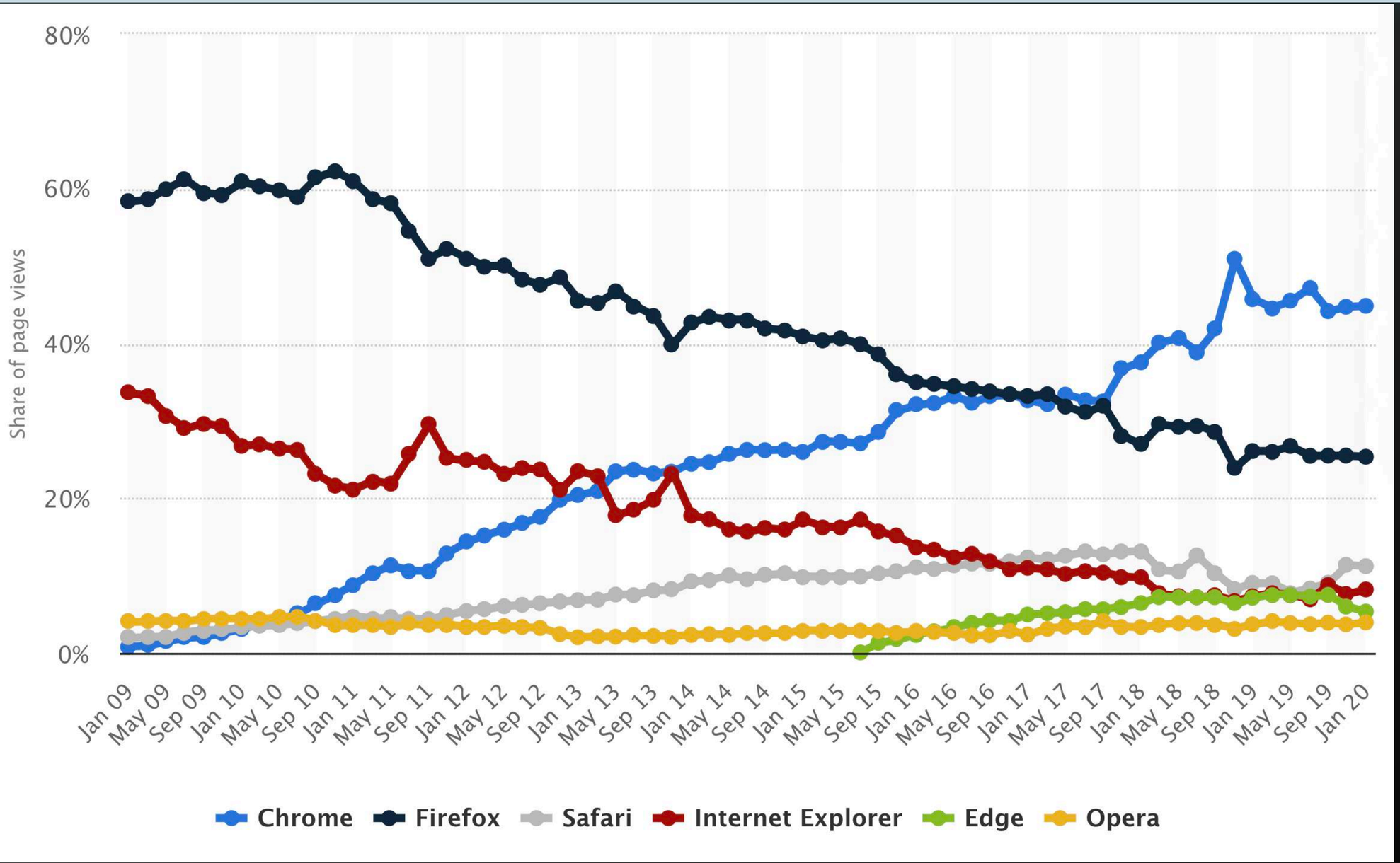
前端。瀏覽器（Web Browser）

Chrome, by Google, in 2008



Edge, by Microsoft, in 2015

前端。瀏覽器 (Web Browser)



src:
Statista

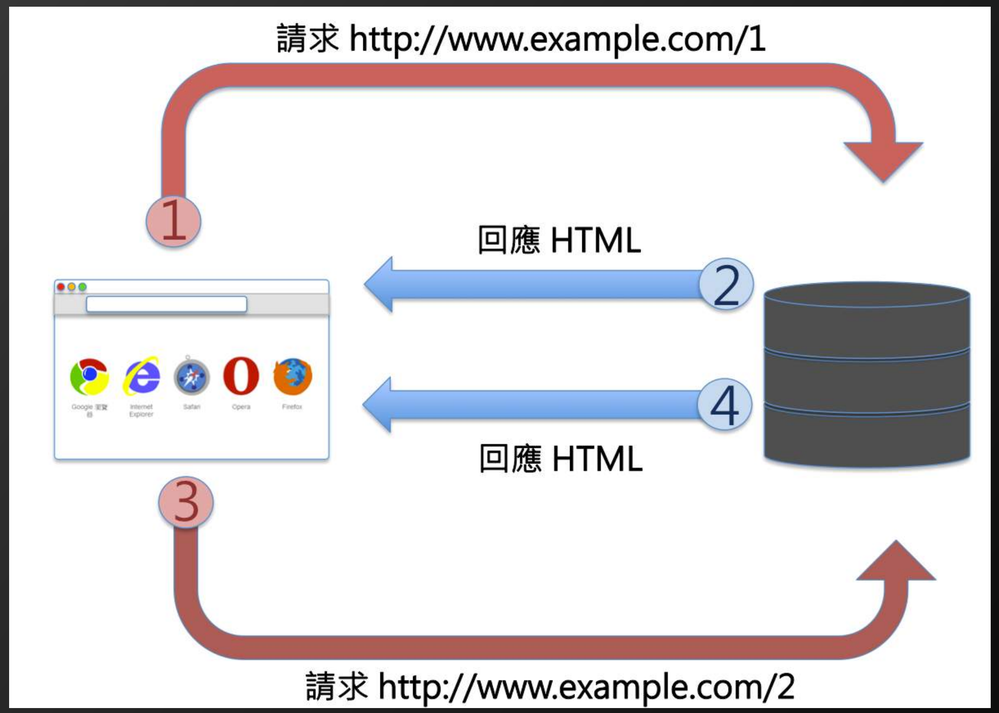
~~前端~~。網頁

這二十幾年來網頁有許多的改變：https://archive.org/

- Web 1.0 (90's ~ early 00's)
  - 文字 + 圖片 + 超連結 (HyperText Markup Language, HTML)
  - 各種小花招 (跑馬燈、計數器、奇怪的游標...)
  - 開始有搜尋功能
- Web 2.0 (early 00's ~ now)
  - wikipedia.org、各種論壇/部落格
  - 各種需要會員登入的網站
  - e-commerce
  - Social network

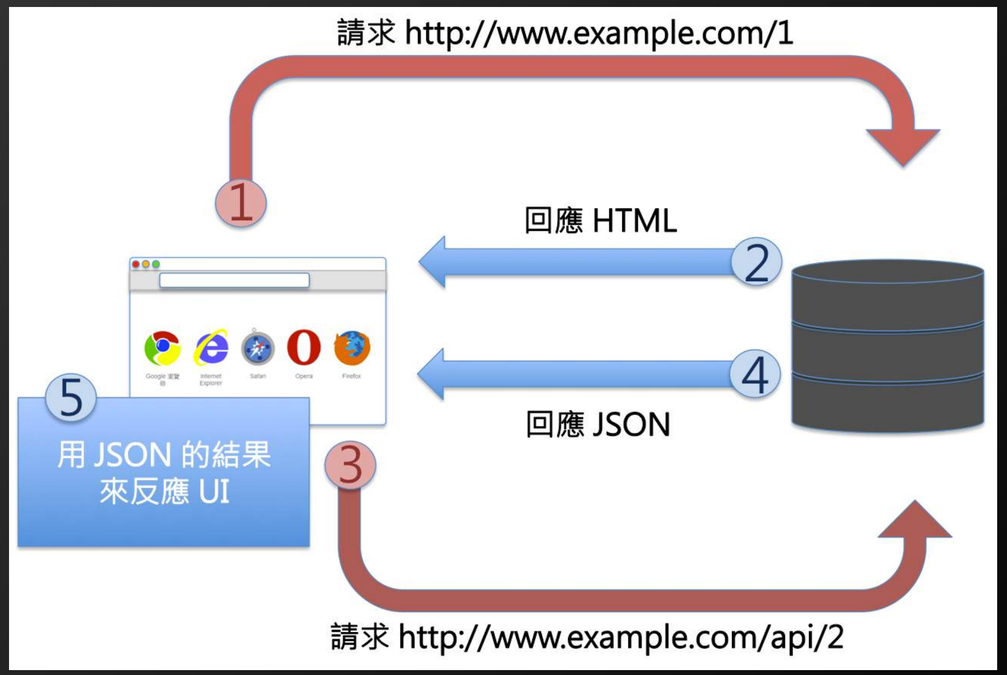
網頁。靜態網頁

- 瀏覽器打開某網址，而從該網址的 web server 回傳 HTML 檔，show 在瀏覽器上，網頁的內容不會隨著使用著的瀏覽行為而有所改變
- 如果按下超連結或是提交表單，是跳掉別的網址，網頁內容會重新下載
- 換頁時會有「白畫面



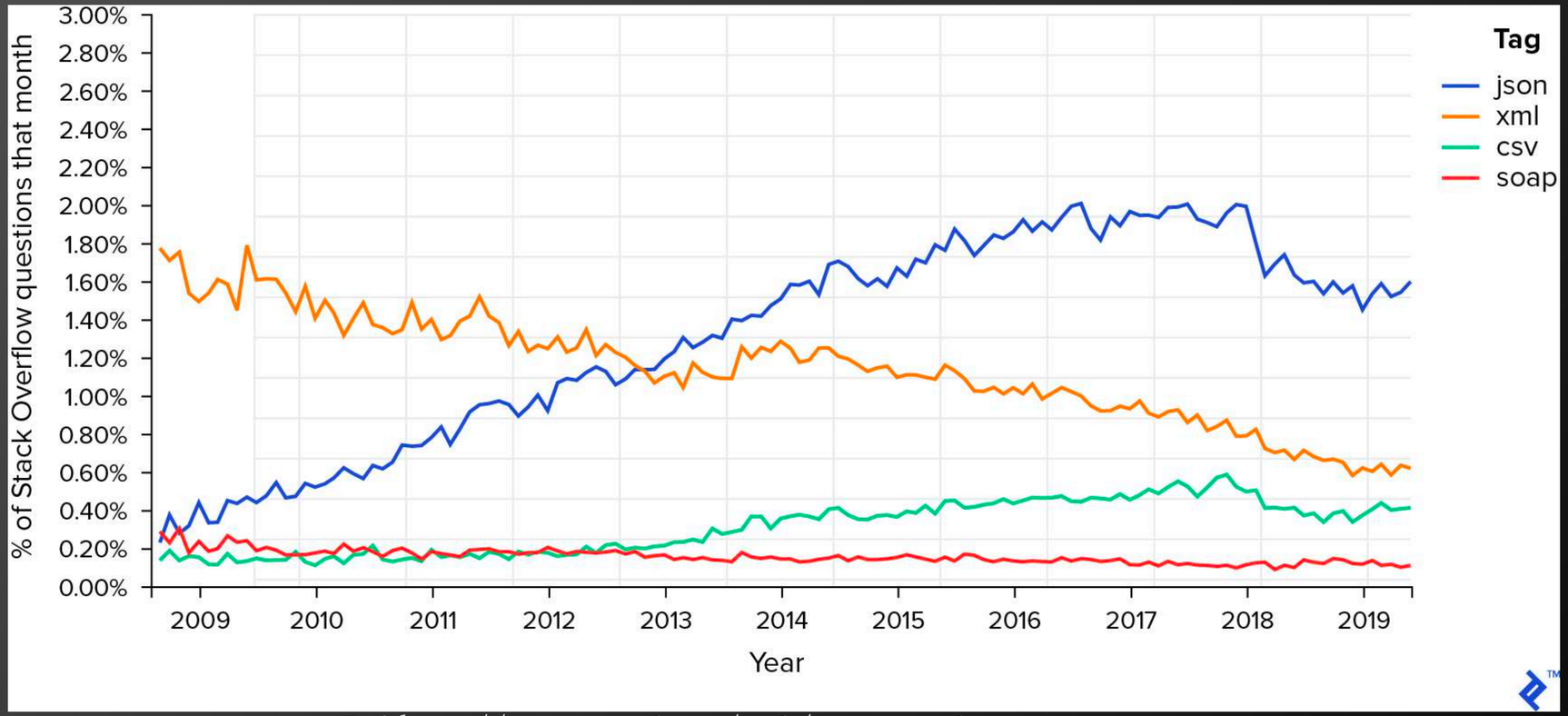
網頁。動態網頁

- 網頁內嵌會聽取(listen)某些事件(events, 如：滑鼠點擊、移動，鍵盤輸入，登入...等)的腳本語言(script language, 如：JavaScript)，當事件發生時，瀏覽器會根據腳本的描述去伺服器(i.e. web server)與資料庫抓取資料，回傳到瀏覽器，而更動「部分」的網頁內容
- 計數器：登入時人數加一
- 相簿：按下按鈕可以切換相片
- 搜尋框：輸入文字可以搜尋資料
- 會員登入：根據會員資料顯示不同內容
- 社群網路：朋友互動的即時顯示



Ajax: Asynchronous JavaScript and XML

- Ajax 透過非同步的方式跟後台拿資料，在這中間使用者還是可以進行其他操作，不會被打斷
- 雖然叫 Ajax, 但現代的網頁都用 JSON



Ref: https://www.toptal.com/web/json-vs-xml-part-1

Ajax: Asynchronous JavaScript and XML

- XML vs. JSON

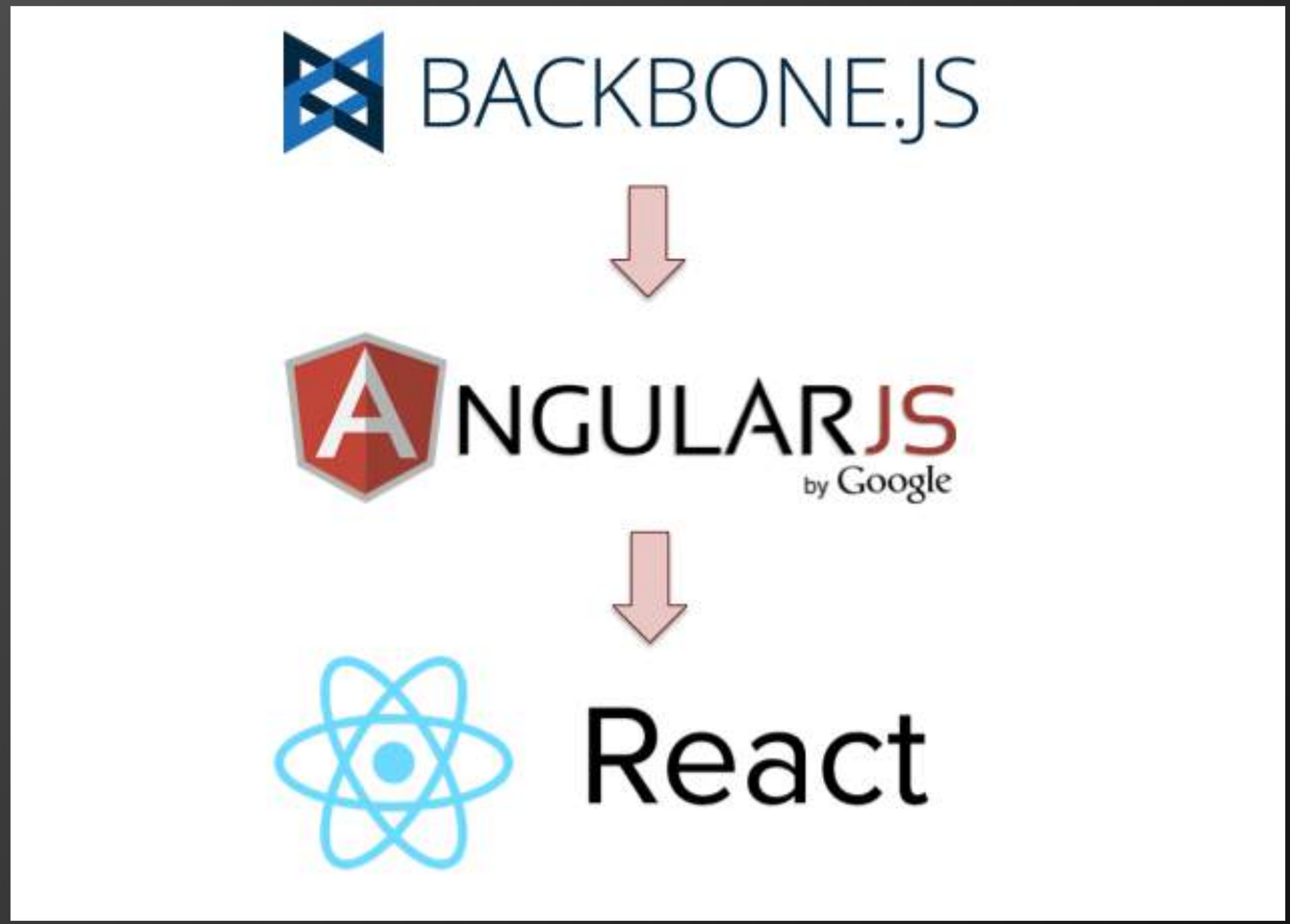
```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <department>
  - <employee>
      <name>John Doe</name>
      <job>Software Analyst</job>
      <salary>2000</salary>
    </employee>
  - <employee>
      <name>Jane Fletcher</name>
      <job>Designer</job>
      <salary>2500</salary>
    </employee>
  </department>
```

```
{
  "arguments" : { "number" : 10 },
  "url" : "http://localhost:8080/restty-tester/collection",
  "method" : "POST",
  "header" : {
    "Content-Type" : "application/json"
  },
  "body" : [
    {
      "id" : 0,
      "name" : "name 0",
      "description" : "description 0"
    },
    {
      "id" : 1,
      "name" : "name 1",
      "description" : "description 1"
    }
  ],
  "output" : "json"
}
```

- 前後端的分工逐漸明確
  - 前端：JavaScript 來負責顯示畫面 (client-side rendering)
  - 後端：as an API server, 負責透過定義好的介面來產生資料
  - 框架：使用Ajax，並定義一些 MVX (如：MVC) 的架構

前端。Client-side Rendering 的演進

- 從 2010 年開始，風向一直在變



React.js。元件式開發 + 狀態決定 UI

Component-based implementation with states —

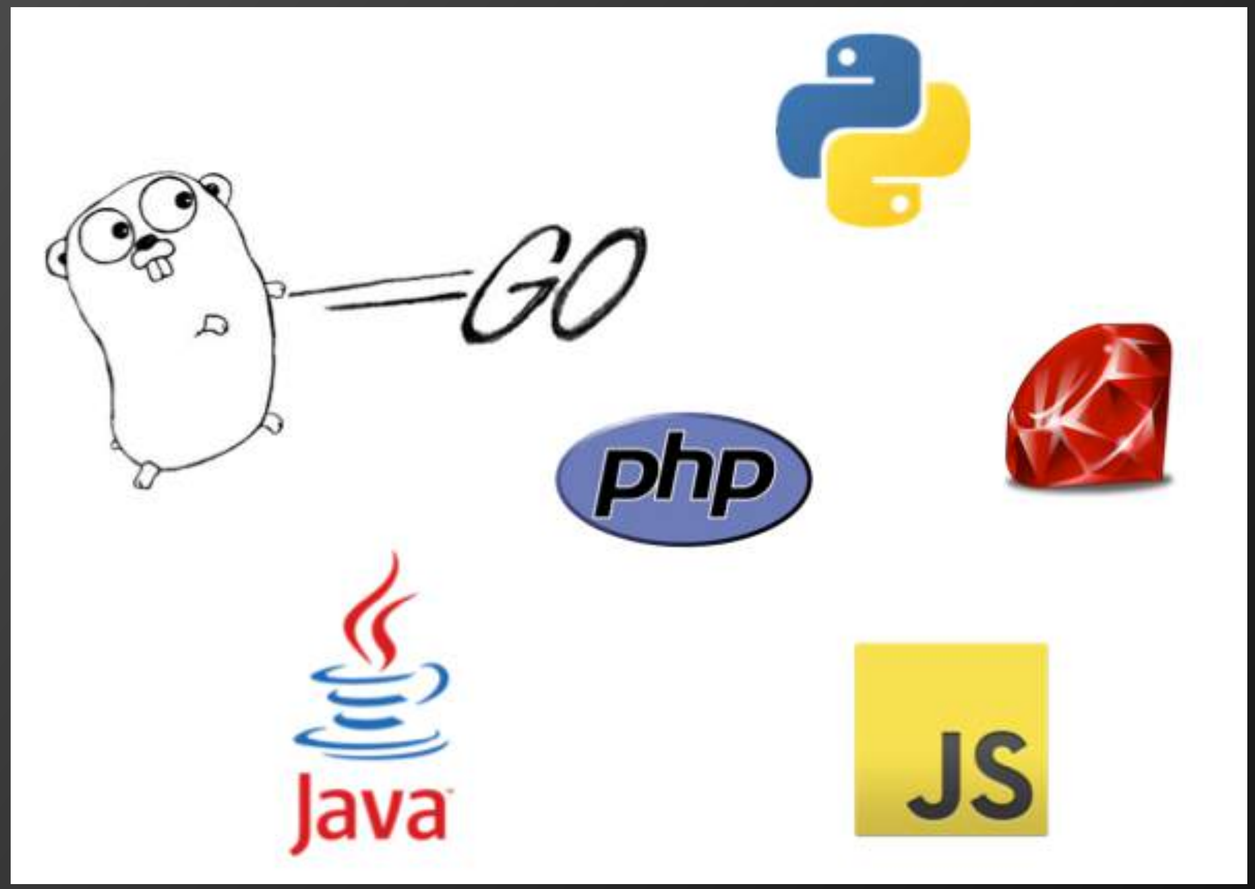
- 以前：某個 UI 呈現出來的 bug 可能要數個操作步驟來重現，debug 過程可能會讓你很崩潰
- With React.js components/states :

```
let AppUI = AppComponent(state);
```

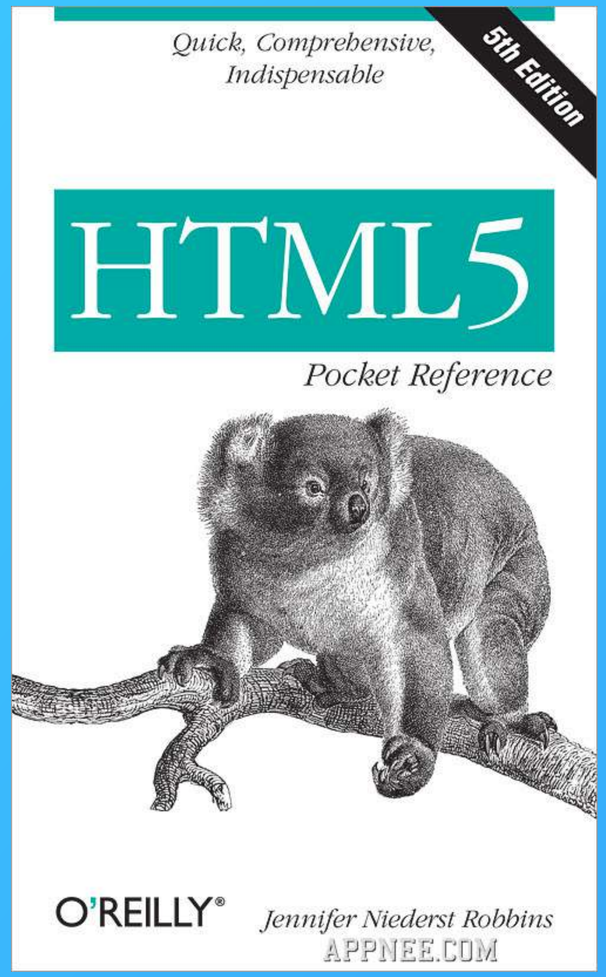
  - 讓資料儲存在單一地方 (i.e. single source of truth)，所有的改變會立即反應在 component 上面
  - 除了 debug 較為輕鬆之外，頁面更新的速度也可以變快 (more to cover later)
    // more to cover in later meetings

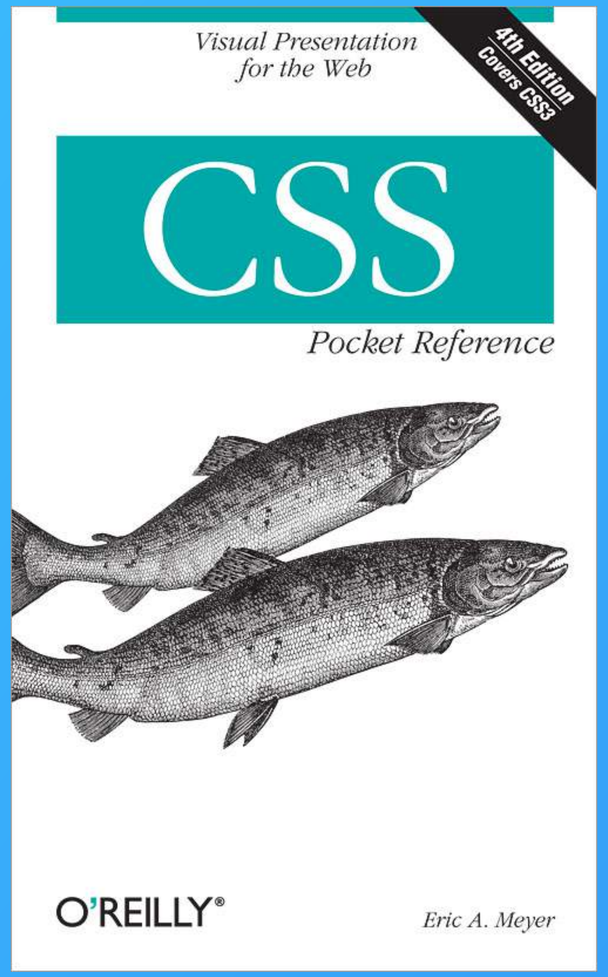
後端語言更是百花齊放～

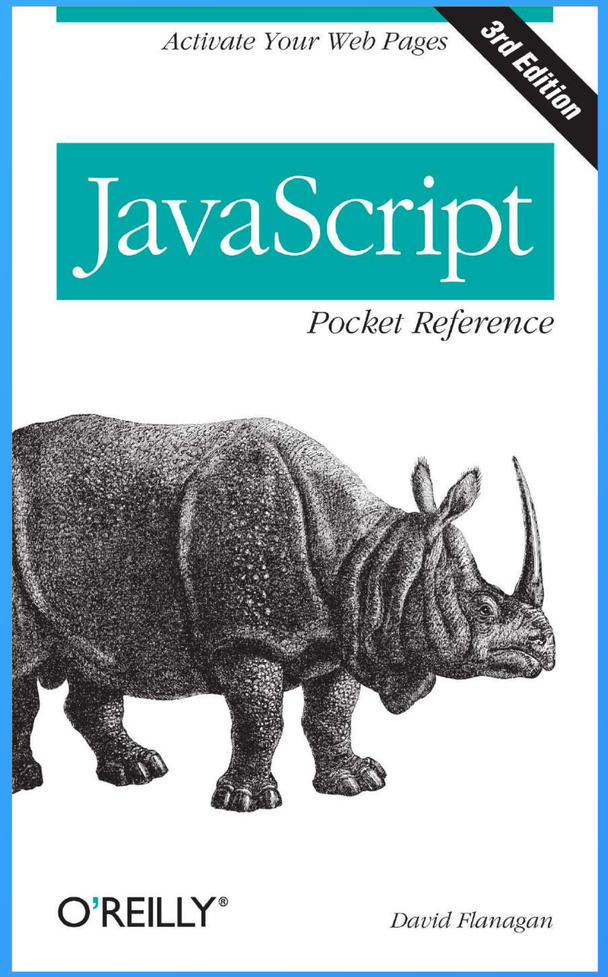
- 可以跑在伺服器 (電腦) 的程式語言都可以用



HTML/CSS/~~JavaScript~~ Basics



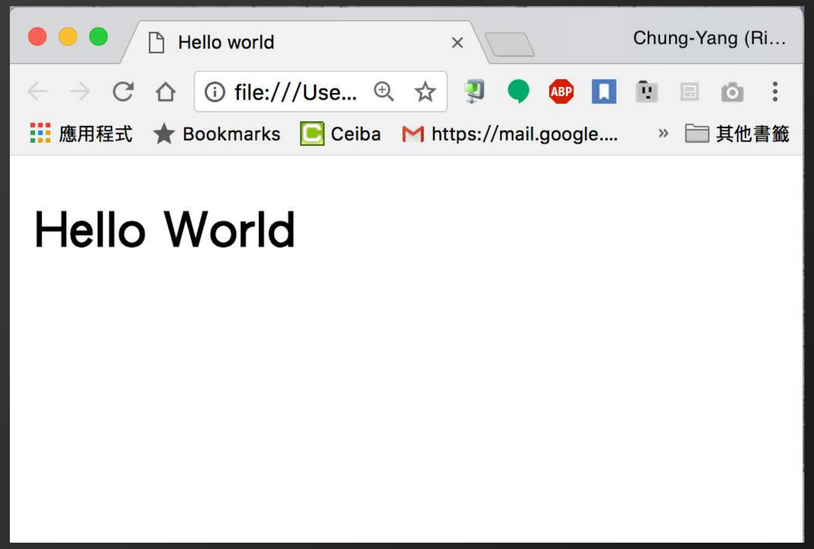




HTML (HyperText Markup Language)

- A "Hello World" example

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello world</title>
  </head>
  <body>
    <p> Hello World </p>
  </body>
</html>
```



- Try…
  - 在 "Hello World" 前後加入空白或是換行
  - 試試看 , 等其他 text elements

HTML Document Structure

<html> — Main root

<head> — Document metadata (optional)

- <link>: to specify the external resource, especially link to style sheets (CSS)
- <style>: style information for a document
- <title>: title shown in a browser's title bar
- <meta>: other meta data

<body> — Sectioning root

- Various tags to define the document layout

HTML — Tag-based Syntax

【 Tags 】

- For most of the tags…

```
<tag attribute1="value" attribute2="value">
Some text to display </tag>
```

- Some exception (no closing tag)

```
<img src="someURL" alt="someText">
```

【 字串(value) 】

- 可以用單引號，也可以用雙引號

【 Comments 】

- are in `<!--...-->`

HTML Elements

Content sectioning To organize the document content into logical pieces.	• <h1>…<h6> • <header>, <footer> • <article>, <address>, <nav>, <section>
Text content To organize blocks or sections of content in <body>…</body>	• <p>, • , : ordered, unordered list - : list items • <dl>: description list - <dt>: a term in <dl> - <dd>: details for the term <dt> • <figure>, <figcaption>, <blockquote> • <hr>

Ref: https://developer.mozilla.org/en-US/docs/Web/HTML/Element

HTML Elements

Inline text semantics	• <a href="…">: hyper link • • , <i>, <mark>, <q>, <s>, • <cite>, <code>, <time> •
Image and multimedia	• <img src="…" alt="…"> • <audio>, <video>, <track> • <map>, <area>: to define image map
Scripting	• <script type="text/javascript" src="someSource.js"> • <noscript>: in case a script type is unsupported • <canvas>: for canvas or webGL APIs

Ref: https://developer.mozilla.org/en-US/docs/Web/HTML/Element

HTML Elements

Table content	• <table> • <caption> • <th>, <tr>, <td>: header, row, data cell
Interactive elements	• <dialog> • <menu>, <menuitem>
Embedded content	• <applet>

Ref: https://developer.mozilla.org/en-US/docs/Web/HTML/Element

HTML Elements

Forms	• <form> • <label for="…"> • <input type="…" value="…"> • <output name="…"> • <button name="…"> • <select>, <option> • <fieldset>, <datalist>, <optgroup> • <meter>, <progress>

Ref: https://developer.mozilla.org/en-US/docs/Web/HTML/Element

網頁。外觀 / 排版

- 前面講的各種 HTML elements, 可以讓你建置網頁的各種樣式與內容
- 至於網頁的外觀與排版，則可以靠 tags 裡頭各式 attributes 來控制
  - align, bgcolor, border, height, size, width…
- 不過把外觀的控制散落到各個 elements 將會使得 code 很難維護，也很難保持整體外觀的一致性

CSS。化妝網頁的衣裳

- CSS: Cascading Style Sheets
  - 「階層式」、「結構化」地定義了 HTML document 的外觀樣式
- CSS 檔案 = 許多 rules 所組成
  - rule = selector {
    property: value;
    /* more properties...*/
    }
- HTML 檔案裡頭 element 的樣式就由這些 rules 來定義

CSS。化妝網頁的衣裳

- Rule precedence
  - Rule with more specific selector > less specific selector
  - 後面的 rule > 前面的 rule
- Comments are in /* … */
- CSS playground: http://dabblet.com/

CSS。應用方式

1. 直接在 <head> 裡頭 include .css file
   - `<link rel="stylesheet" type="text/css" href="path/to/style.css">`
2. 將一段 CSS code 嵌入到 HTML 裡頭
   - ```
     <style>
         a { color: purple; }
     </style>
     ```
3. 直接寫在 tag 的 attribute
   - `<div style="border: 1px solid red;">`
     ```

CSS ◦ Basic Selectors

Given an HTML element

```
<div class="class1 class2" id="anID"
     attr="value" otherAttr="en-us foo bar" />
```

We can decorate it with one of the following CSS selectors —

1. Select by one or more class names
   - `.class1 { }`
   - `.class1, .class2 { }`
2. Select by the tag name
   - `div { }`
3. Select by its id
   - `#anID { }`

CSS ◦ More on Class Selectors

```
<div class="one"> 1
  <div class="two"> 2in1
    <div class="three"> 3in2in1 </div>
  </div>
  <div class="three"> 3in1
    <div class="two"> 2in3in1 </div>
  </div>
</div>
<div class="one two"> 1and2 </div>
```

- `.one.two { /* Select 1and2 */ }`
- `.one .two { /* Select 2in1, 2in3in1 */ }`
- `.one > .two { /* Select 2in1 */ }`
- `.one,.two { /* Select 1, 2in1, 1and2 */ }`

CSS ◦ Other Selectors

- * : Select all elements
  - `* { }  /* all elements */`
  - `.parent * { }  /* all descendants */`
  - `.parent > * { } /* all children */`
- Select an element based on its adjacent sibling
  - `.i-am-just-before + .this-element { }`
- Select an element based on any sibling preceding it
  - `.i-am-any-element-before ~ .this-element { }`

CSS ◦ Pseudo Class Selectors

- selector:pseudo-class {
  property: value;
  }
- pseudo-class := hover, active, focus, first-child, nth-child(n), not(s), visited, link

CSS Selectors • Properties and Values

```css
selector {

    /* Units of length can be absolute or relative. */

    /* Relative units */
    width: 50%;         /* percentage of parent element width */
    font-size: 2em;     /* multiples of element's original font-size */
    font-size: 2rem;    /* or the root element's font-size */
    font-size: 2vw;     /* multiples of 1% of the viewport's width (CSS 3) */
    font-size: 2vh;     /* or its height */
    font-size: 2vmin;   /* whichever of a vh or a vw is smaller */
    font-size: 2vmax;   /* or greater */

    /* Absolute units */
    width: 200px;       /* pixels */
    font-size: 20pt;    /* points */
    width: 5cm;         /* centimeters */
    min-width: 50mm;    /* millimeters */
    max-width: 5in;     /* inches */
```

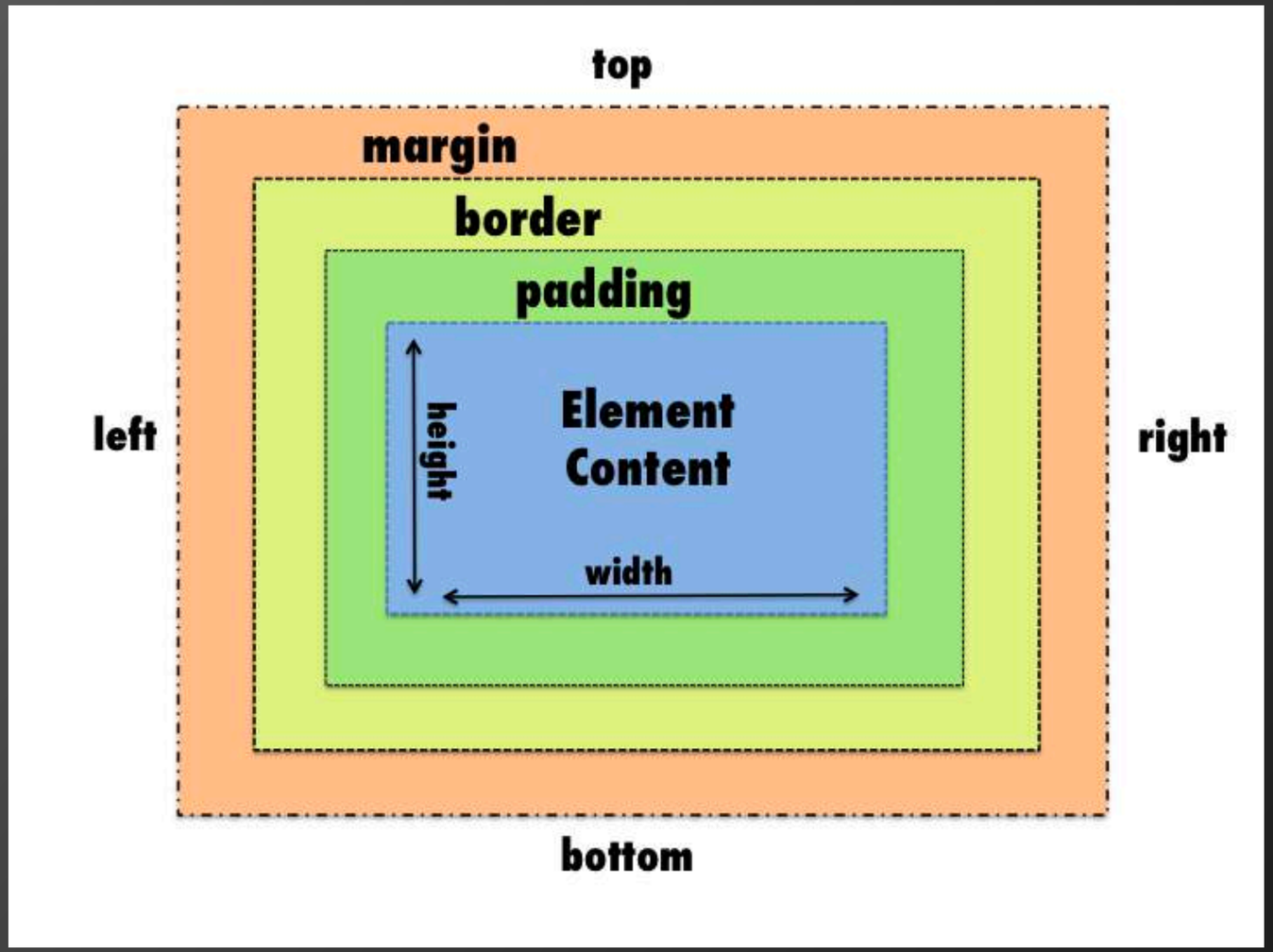
```
    /* Colors */
    color: #F6E;                  /* short hex format */
    color: #FF66EE;               /* long hex format */
    color: tomato;                /* a named color */
    color: rgb(255, 255, 255);    /* as rgb values */
    color: rgb(10%, 20%, 50%);    /* as rgb percentages */
    color: rgba(255, 0, 0, 0.3); /* as rgba values (CSS 3) Note: 0 <= a <= 1 */
    color: transparent;           /* equivalent to setting the alpha to 0 */
    color: hsl(0, 100%, 50%);     /* as hsl percentages (CSS 3) */
    color: hsla(0, 100%, 50%, 0.3); /* as hsl percentages with alpha */

    /* Borders */
    border-width:5px;
    border-style:solid;
    border-color:red;       /* similar to how background-color is set */
    border: 5px solid red; /* this is a short hand approach for the same */
    border-radius:20px;     /* this is a CSS3 property */

    /* Images as backgrounds of elements */
    background-image: url(/img-path/img.jpg); /* quotes inside url() optional */

    /* Fonts */
    font-family: Arial;
    /* if the font family name has a space, it must be quoted */
    font-family: "Courier New";
    /* if the first one is not found, the browser uses the next, and so on */
    font-family: "Courier New", Trebuchet, Arial, sans-serif;
}
```

CSS ◦ Box Model



- box-sizing: border-box | content box
- border-radius: 50%
- margin: calc(…)

CSS ◦ Positioning

- position:

  static | relative | fixed | absolute | sticky
- Other properties —
  - top, bottom, left, right

CSS。Display Modes and Flexbox

- display: block | none | inline | flex

display: flex

- flex-direction: row | column | row-reverse | column-reverse
- flex-wrap: wrap | now rap | wrap-reverse
- // 與 flex-direction 平行
  justify-content: center | flex-start | flex-end | space-around
- // 與 flex-direction 垂直
  align-items: center | flex-start | flex-end | stretch | baseline
- // Align 自己，而不是裡面的 elements
  align-self: center | flex-start | flex-end

CSS • Filter and Box-shadow Properties

Syntax —

- filter: none | blur() | brightness() | contrast() | drop-shadow() | grayscale() | hue-rotate() | invert() | opacity() | saturate() | sepia() | url();
- box-shadow: none | h-offset v-offset blur spread color | inset | initial | inherit;

CSS - Transition Property

讓 CSS 的 styling 變化看起來更 smooth 一些 —

- Properties: `transition, transition-delay, transition-duration, transition-property, transition-timing-function`
- [Example]

```
div {
  … /* width, height, background, etc */
  transition: width 2s, height 2s, transform 2s;
}
div:hover {
  … /* width, height, etc */
  transform: rotate(180deg);
}
```

CSS ◦ Element Transform

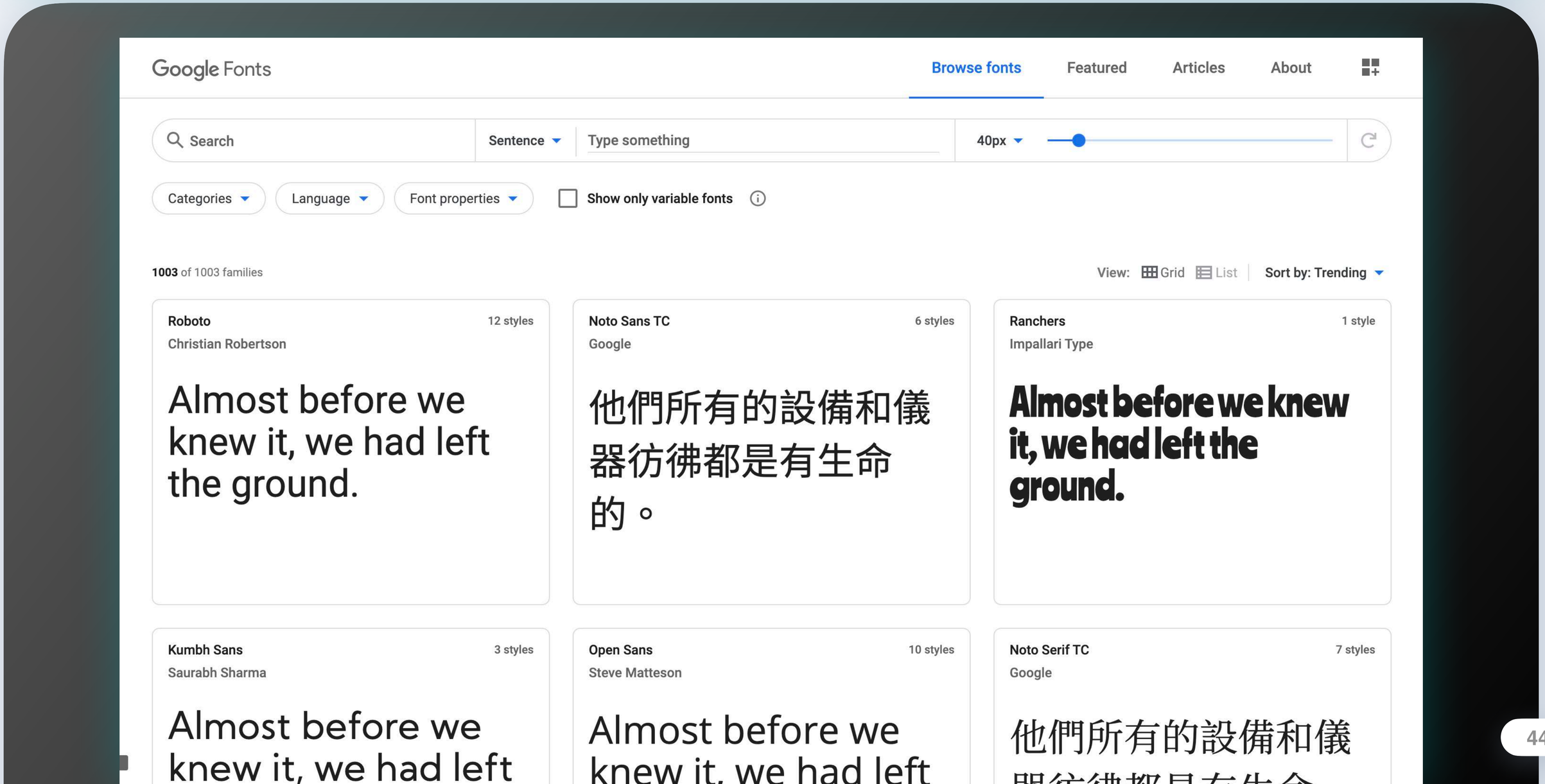
Syntax - `transform : method(…)`

- `rotate(45deg); rotate(1,57rad);`
- `translate(x-offset, y-offset);`
- `scaleX(1.5); scaleY(2.8); scale(1.5, 2.8);`
- `skew(x-angle, y-angle);`
- `rotateX(angle); rotateY(angle);`
  `rotateZ(angle);`

CSS - @keyframes and Animation

```
@keyframes rulename { /* 定義動畫規則 */
  40% { transform: rotate(0deg); }
  45% { transform: rotate(-5deg); }
  55% { transform: rotate(5deg); }
  60% { transform: rotate(0deg); }
}
someSelector {
  animation: rulename 3s infinite;
}
```

CSS • Useful Resource for Text Fonts



Take-Home Practice #1

- Design a stylish music player using HTML and CSS

感謝聆聽！

Ric Huang / NTUEE

(EE 3035) Web Programming