Software Studio
軟體設計與實驗

Introduction to Web Programming

Hung-Kuo Chu

Department of Computer Science

National Tsing Hua University

CS2410

Basic Web Workflow

Client

Request: Website

Response: IP address

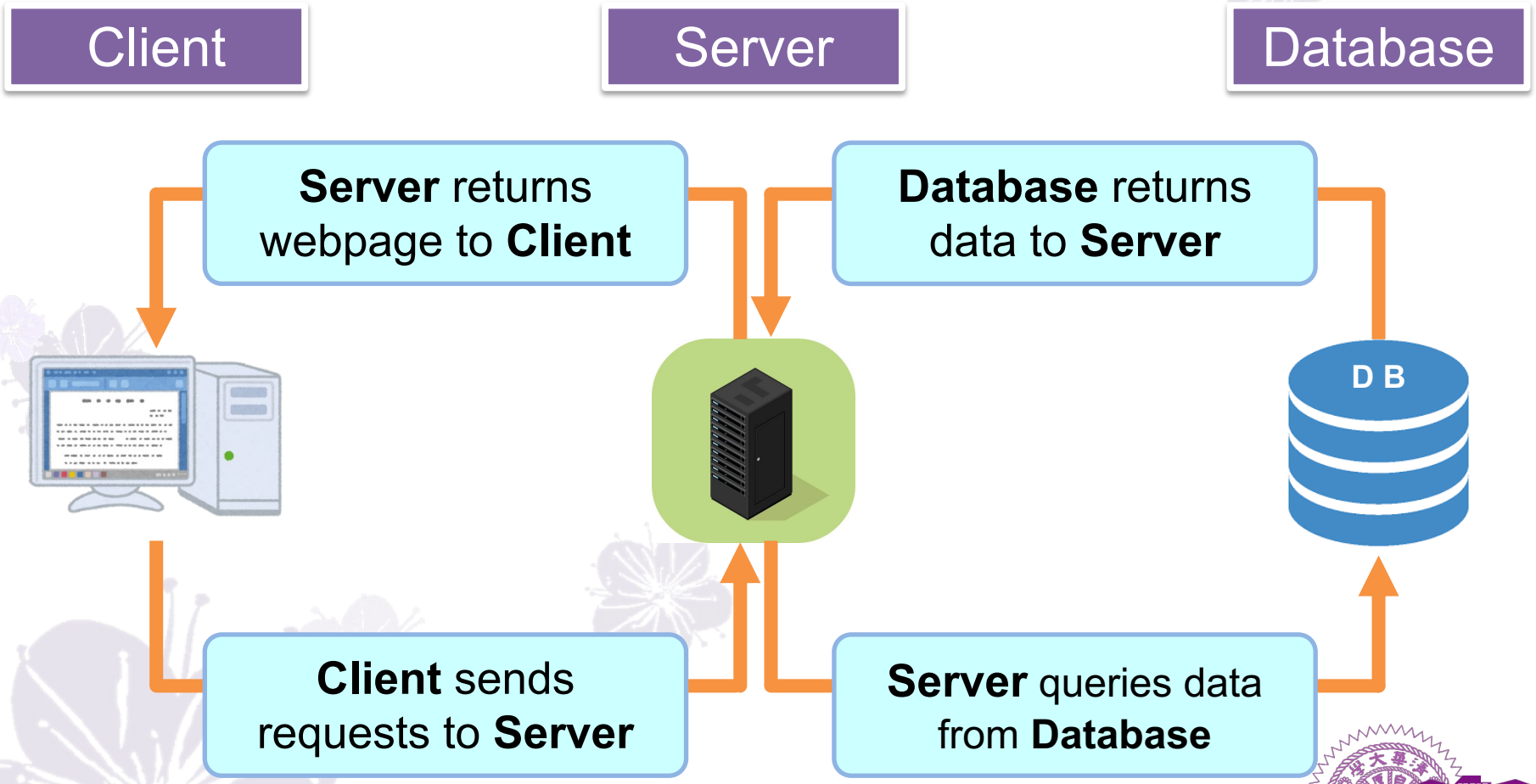
DNS Server

Request: IP address

Response: Data

Main Server

Web Architecture



Web Architecture

- **Client (Frontend)**
  - Send requests to Server.
  - Process/Display data received from Server via Web Browser.
- **Server (Backend)**
  - Place where we put the source codes of web program.
  - Send queries to Database.
  - Response to the requests from Client.
- **Database (Storage)**
  - Place where we put files (images, videos, user profiles, etc).

Fundamental Components

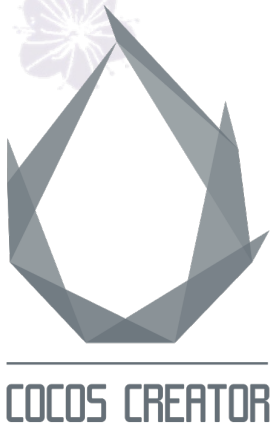
WEB PROGRAMMING

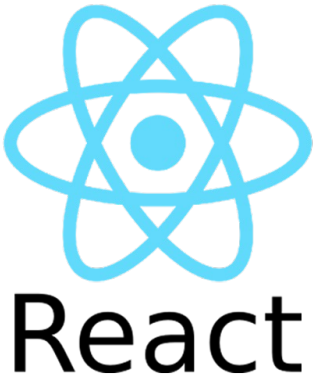
Client (Frontend)

- **HTML** (**H**yper**T**ext **M**arkup **L**anguage)
  - Using a set of **tags** and **attributes** to define the **contents** and **appearance** of web apps.
- **CSS** (**C**ascading **S**tyle **S**heets)
  - Formatting the **appearance** of web apps
  - Layout, colors, and fonts.
- **JavaScript**
  - Define the **behaviors** of web apps
  - Adding dynamic effects

Client (Frontend)

- **TypeScript**
  - A typed superset of JavaScript that compiles to JavaScript
  - Supports object-oriented programming
- **Cocos Creator**
  - A game engine for web game
  - Easy to control object in scene
- **React**
  - A javascript library for building user interfaces



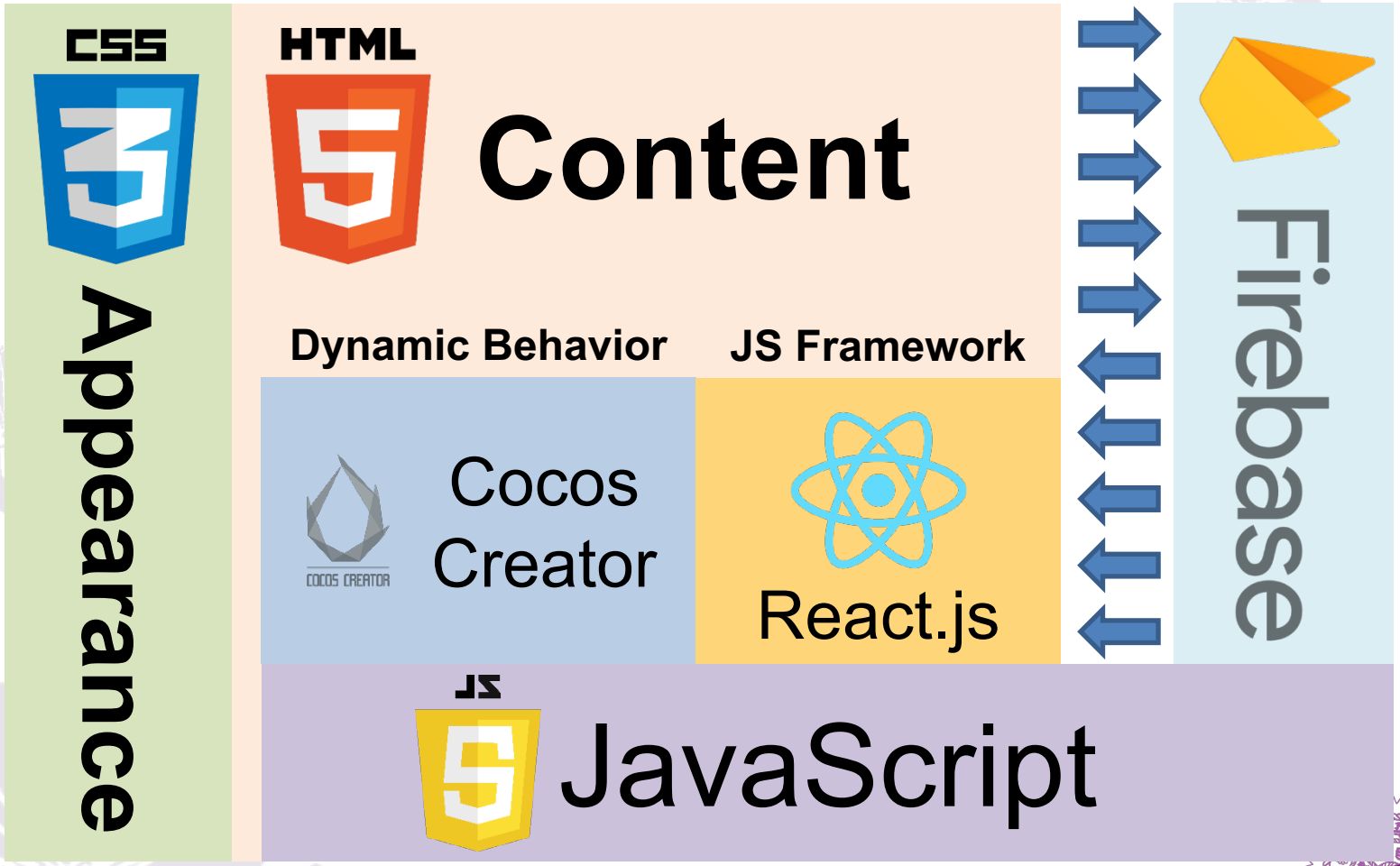


Server (Backend) and Database

- **Firebase**
  - Fully compatible to (wrote by) JavaScript.
  - Build apps fast, without managing infrastructure.
  - Backed by Google, trusted by top apps.
  - Discover more here.
- Other popular web servers we WONT cover.
  - PHP
  - Node.js

Put All Together

Web Application (Web App)



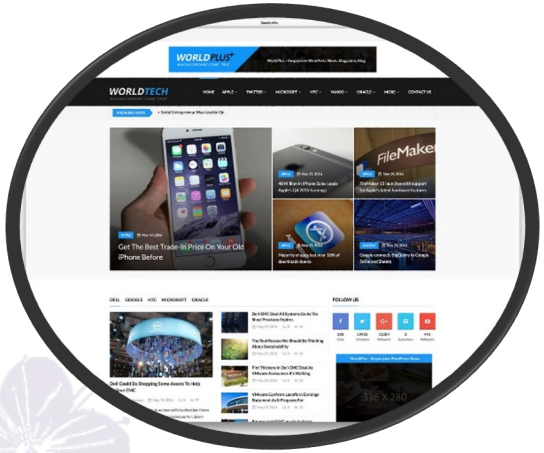
Web App: Pros

- Run "inside" a browser; no complex installation is needed.
- Require very little disk space or computing power on the client.
- Cross-platform compatibility.
- The data is stored remotely; support remote updates.
- Easy communication and cooperation.

Web App: Cons

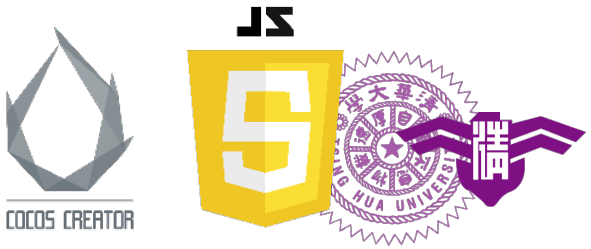
- Need to be coded and run via browsers that follow **standards.**
- Need a connection to the server where the application runs, all the time, and costs a certain **bandwidth.**
- Dependent on the server that hosts the app, **server down app gone**!
- The company offering the web application has complete control over it, causing **privacy** problems.
- Source codes are accessible to Client, causing **piracy** issues.

Web App vs. Website



Static

Dynamic



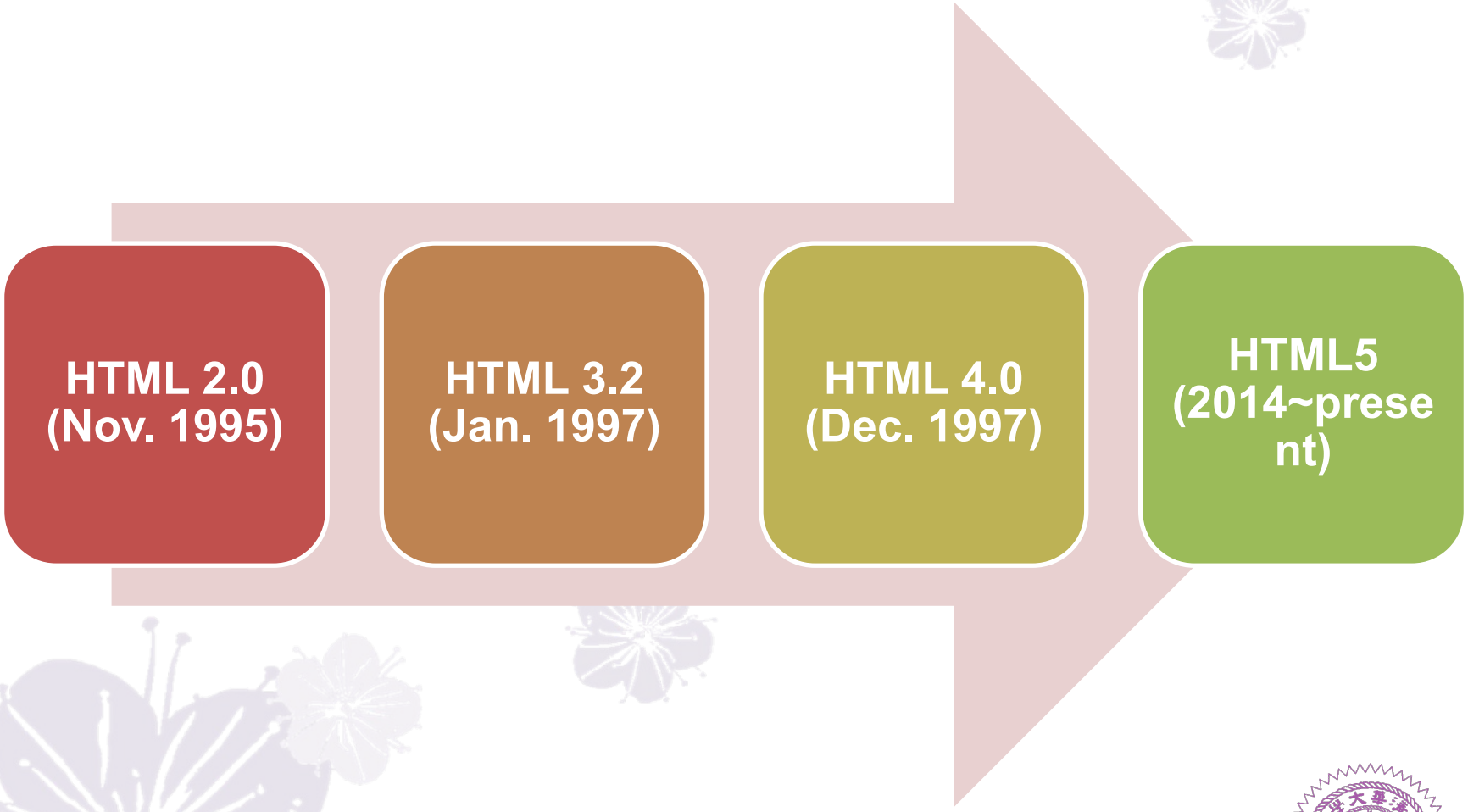
QUICK REVIEWS!


HTML

What is HTML?

- **H**yper**T**ext **M**arkup **L**anguage by W3C.
- The standard **markup language** for creating **web pages** and **web applications**.
- **Web browsers** receive HTML documents from a **web server** or from **local storage** and render them into multimedia web pages.
- HTML describes the **structure** as well as **appearance** of a web page via a pre-defined set of **tags** and **attributes.**

HTML History



Why Use HTML5?

- **Accessibility**
  - ARIA: Semantic tags
- **Video and Audio Support**
  - Flash Player and third-party players OUT!
- **Doctype**
  - No more dirty head tags filled with doctype attributes.
- **Cleaner Code**
  - Semantic code that allows you to easily separate meaning from **style** and **content**.

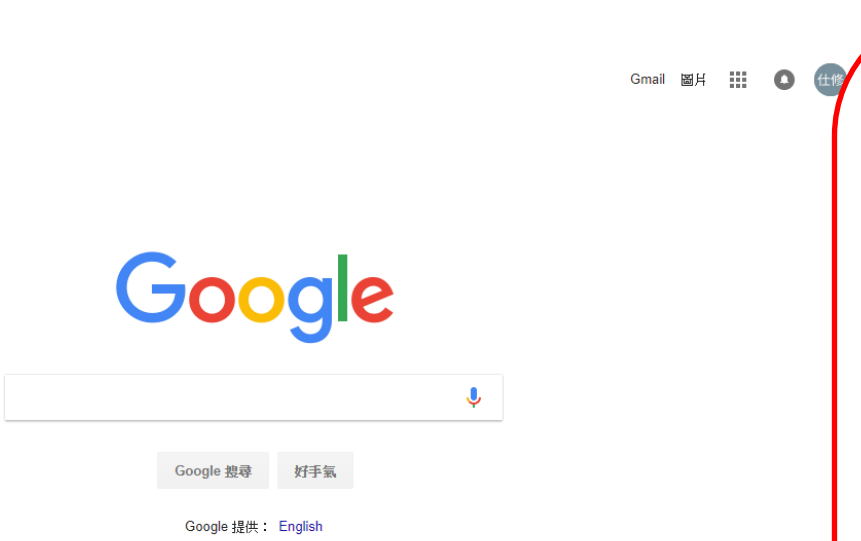
Ref: Top 10 Reasons to Use HTML5 Right Now

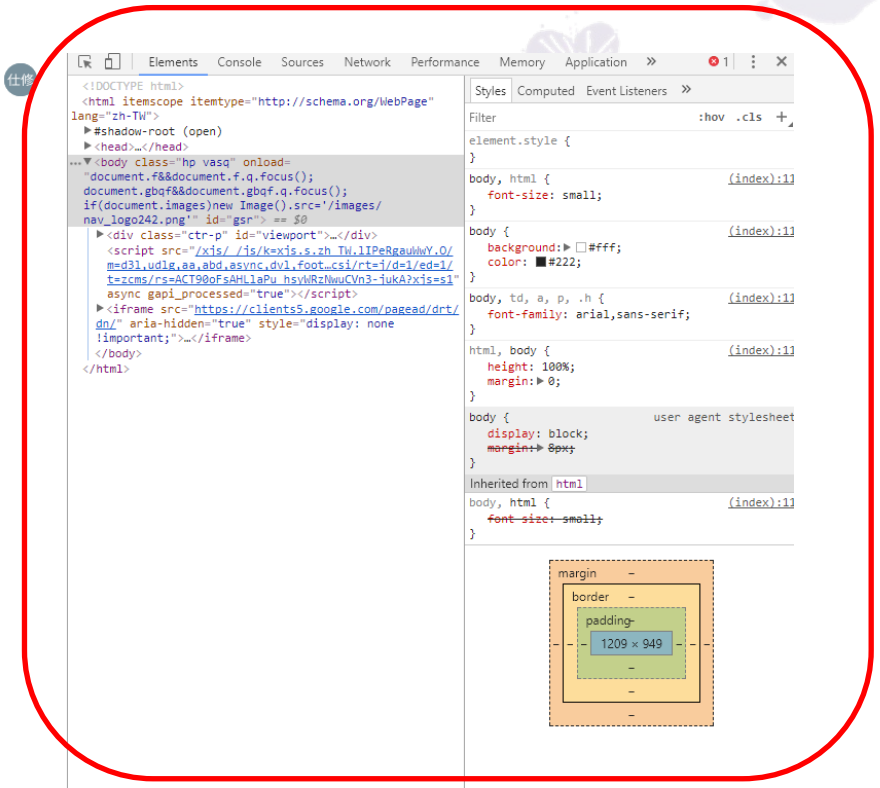
Why Use HTML5? (Cont'd)

- **Smarter Storage**
  - Better security and performance than old cookies
  - Data will persist even after the browser is closed.
- **Better Interactions**
  - New API's support (Canvas, Drag and Drop, etc.).
- **Game Development**
  - 2D/3D web-based games.
- **Legacy/Cross Browser Support**
  - Chrome, Firefox, Safari, IE9 and Opera.
- **Mobile, Mobile, Mobile**

Ref: Top 10 Reasons to Use HTML5 Right Now

Source Code Viewer





Open browser

Press F12

HTML5 Example

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
      <title>HTML example</title>
  </head>
  <body>
    <h1>Hello! HTML5!</h1>
    <h2>Let's go! Software Studio!</h2>
  </body>
</html>
```

Hello! HTML5!

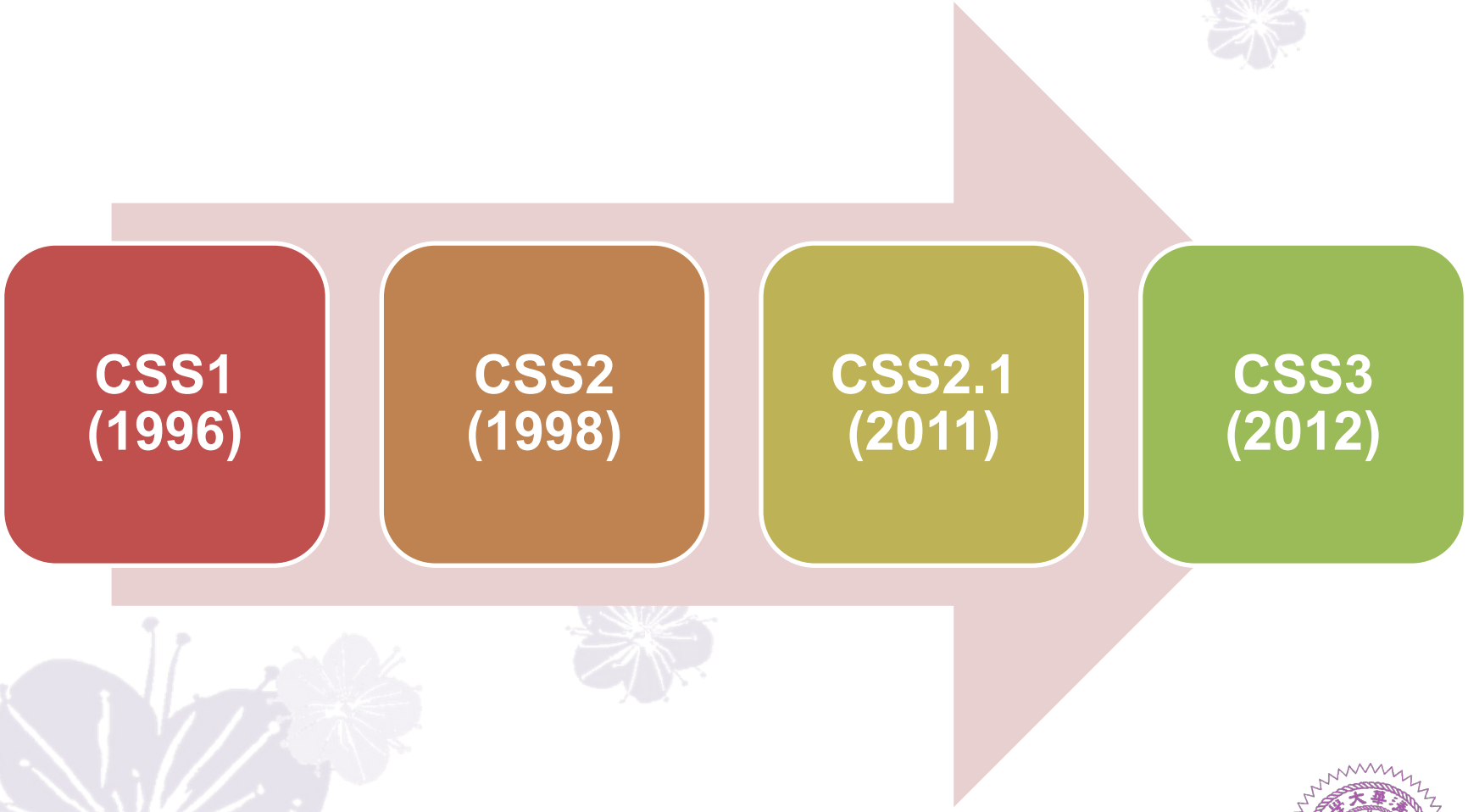
Let's go! Software Studio!


CSS
3

What is CSS?

- **C**ascading **S**tyle **S**heets by W3C.
- A style sheet language used for describing the appearance of a document written in a markup language (e.g., HTML, XML).
- CSS is designed primarily to enable the separation of **appearance** and **content.**
  - layout, colors, and fonts.

CSS History



Style Rule

selector {property1 : value1 [; property2 : value2 [; …]]}

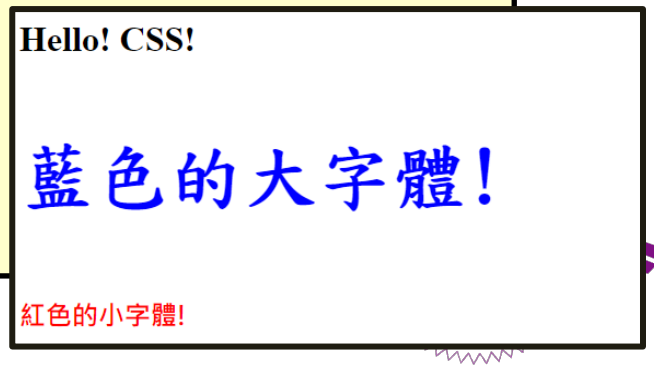
- Selector
  - Points to the HTML element you want to style.
- Declaration
  - The declaration block contains one or more declarations separated by semicolons.
  - Each declaration includes a CSS property name and a value, separated by a colon.
  - A declaration **always ends with a semicolon.**
- Example
  - body {color : white; background : red;}

CSS3 Example

```
<html>
  <meta content="text/html; charset=UTF-8">
  <head>
    <title>CSS example</title>
    <style>
      p{font-size:50pt; font-family:標楷體; font-weight:bold; color:blue}
      p2{font-size:18pt; font-family:微軟正黑體; font-weight:bold; color:red}
    </style>
  </head>
  <body>
   <h1>Hello! CSS!</h1>
    <p>藍色的大字體!</p>
    <p2>紅色的小字體!</p2>
  </body>
</html>
```

Define styles

Use styles



JavaScript

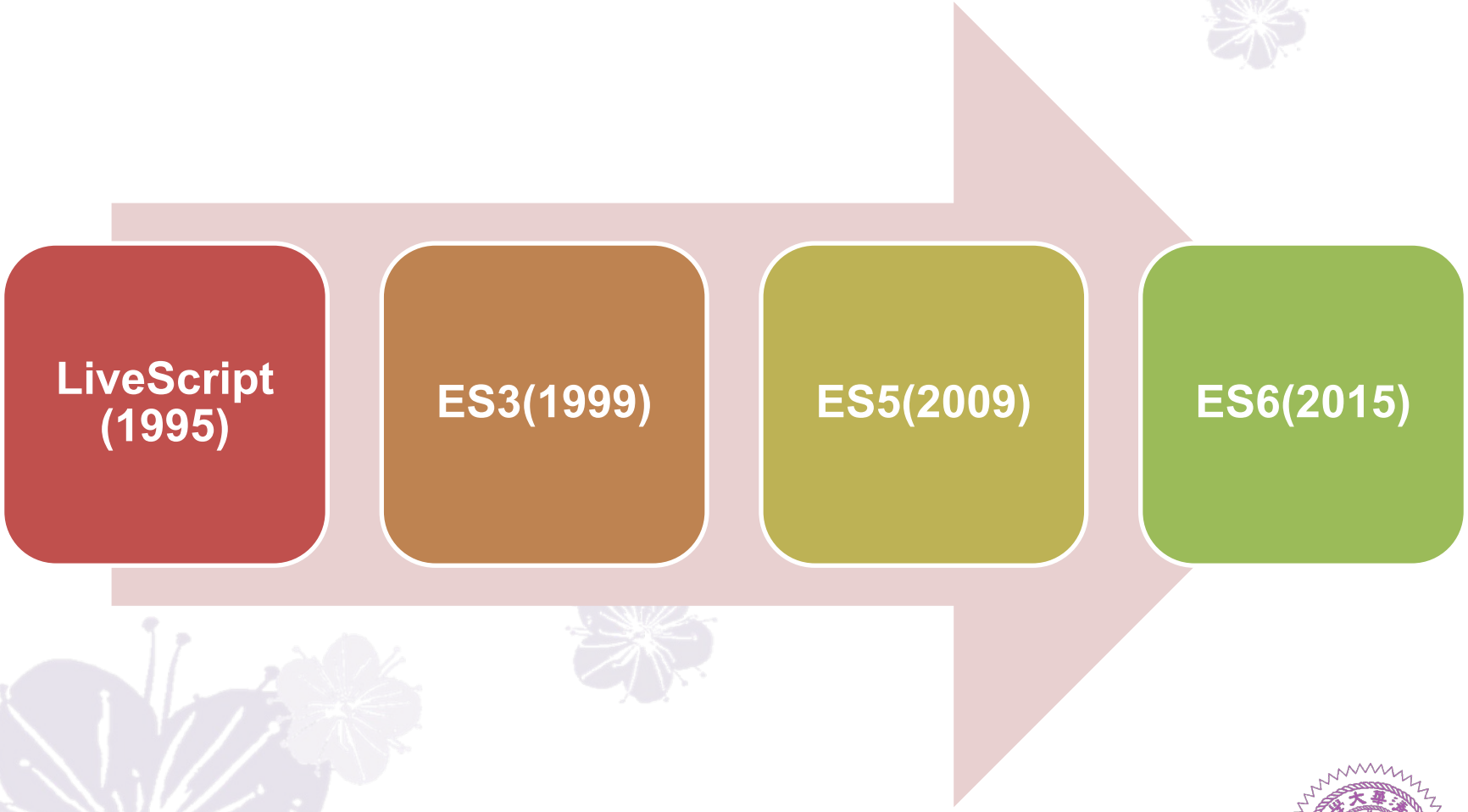
What is JavaScript?

- A high-level, interpreted programming language that enables you to create **dynamically updating content** and **user interaction.**
  - control multimedia
  - animate images
  - etc.
- Client-side/Server-side JavaScript
  - Interact with web application.
  - Communicate with database.
- **JavaScript and Java are distinct and differ!**

What is JavaScript? (Cont'd)

- JavaScript implements **ECMAScript (ES)** standardization.
  - ES5 (2009)
  - **ES6 (2015)**
  - ES7, ES8 …
- Lots of useful frameworks and libraries.
  - jQuery
  - React
  - Firebase, Node.js
  - **WebGL**

JavaScript History

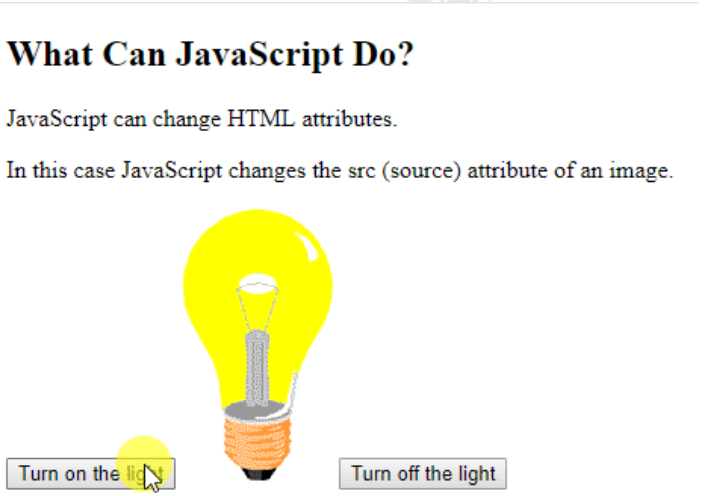


Why Use JavaScript?

- All modern web browsers support JavaScript without the need for plug-ins by means of a built-in JavaScript engine.
- In other words, JavaScript is a **cross-platform** programming language that runs on all machines with browser software.

JavaScript Example (internal)

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can change HTML attributes.</p>

<p>In this case JavaScript changes the src (source) attribute of an image.</p>

<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">
Turn on the light
</button>

<img id="myImage" src="pic_bulboff.gif" style="width:100px">

<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">
Turn off the light
</button>

</body>
</html>
```

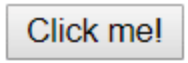


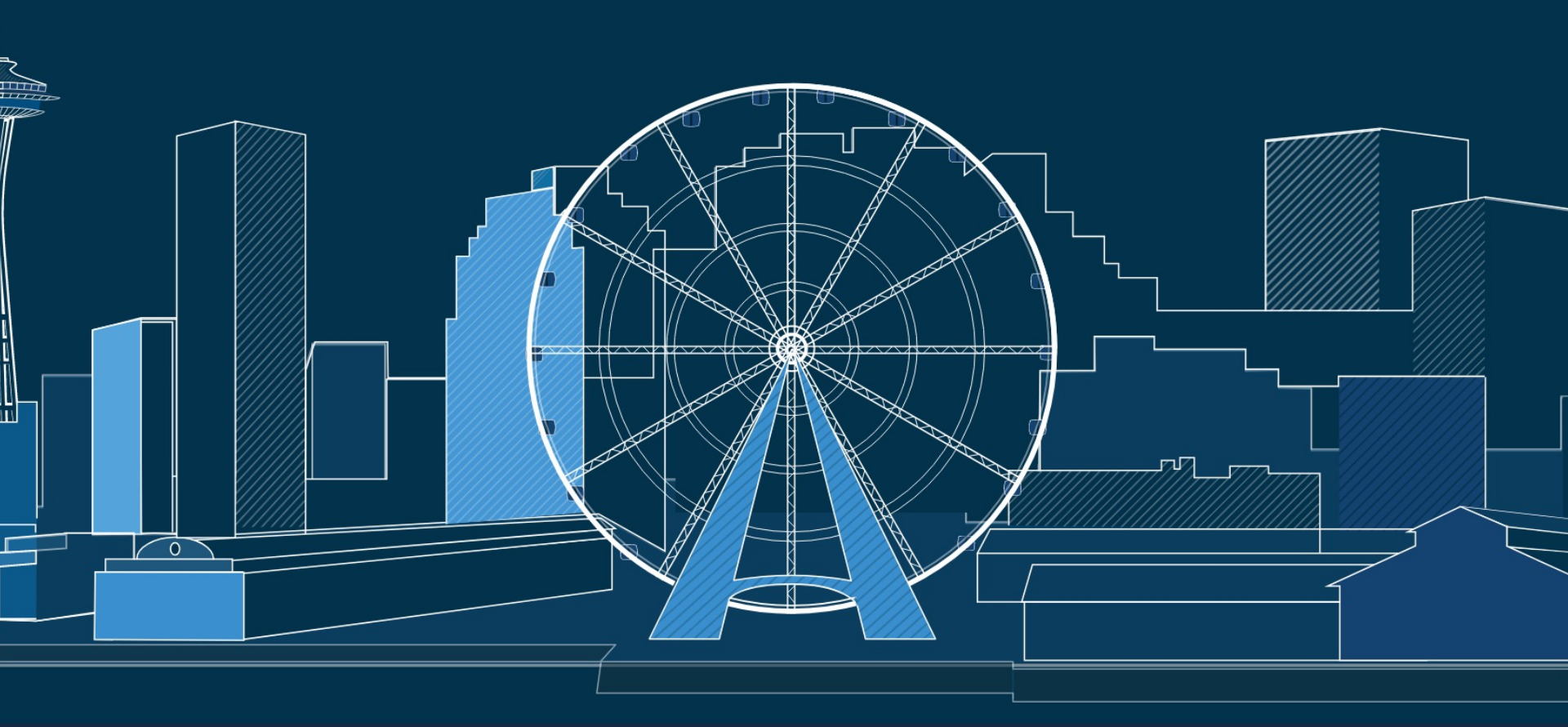
The buttons events are JavaScript code

JavaScript Example (external)

```
<!DOCTYPE html>
<html>
    <head>
            <meta charset = "utf-8">
            <title>JavaScript Example</title>
            <script src="JSexample.js"></script>
    </head>
    <body>
            <button onclick="createParagraph()">Clickme!</button>
    </body>
</html>
```

```
function createParagraph() {
            var para = document.createElement('p');
            para.textContent = 'You clicked the button!';
            document.body.appendChild(para);
}
```


Click me!



TypeScript

JavaScript that scales.

What is TypeScript?

- An open-source programming language built by Microsoft.
- A JavaScript superset, with **static typing** support.
- Make app development **as quick and easy as possible.**

TypeScript in 5 minutes(tutorial)

Why Use TypeScript?

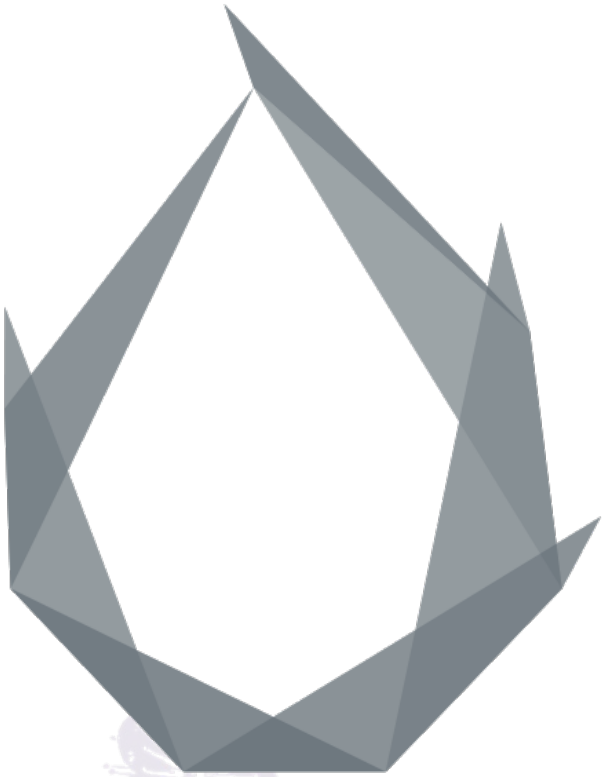
- **Type system** can enhance code quality and understandability.
- Provides **compile-time type safety** for JavaScript code.
- Supports Class, Interface and other **object-oriented programming** techniques.

TypeScript Example

```
class Person {

    private name: string;
    private age: number;
    private salary: number;

    constructor(name: string, age: number, salary: number) {
        this.name = name; this.age = age; this.salary = salary;
    }

    toString(): string {
        return `${this.name} (${this.age}) (${this.salary})`;
    }
}
```

Bind variable ‘name’ with type ‘string’

Define which type will function ‘toString’ return

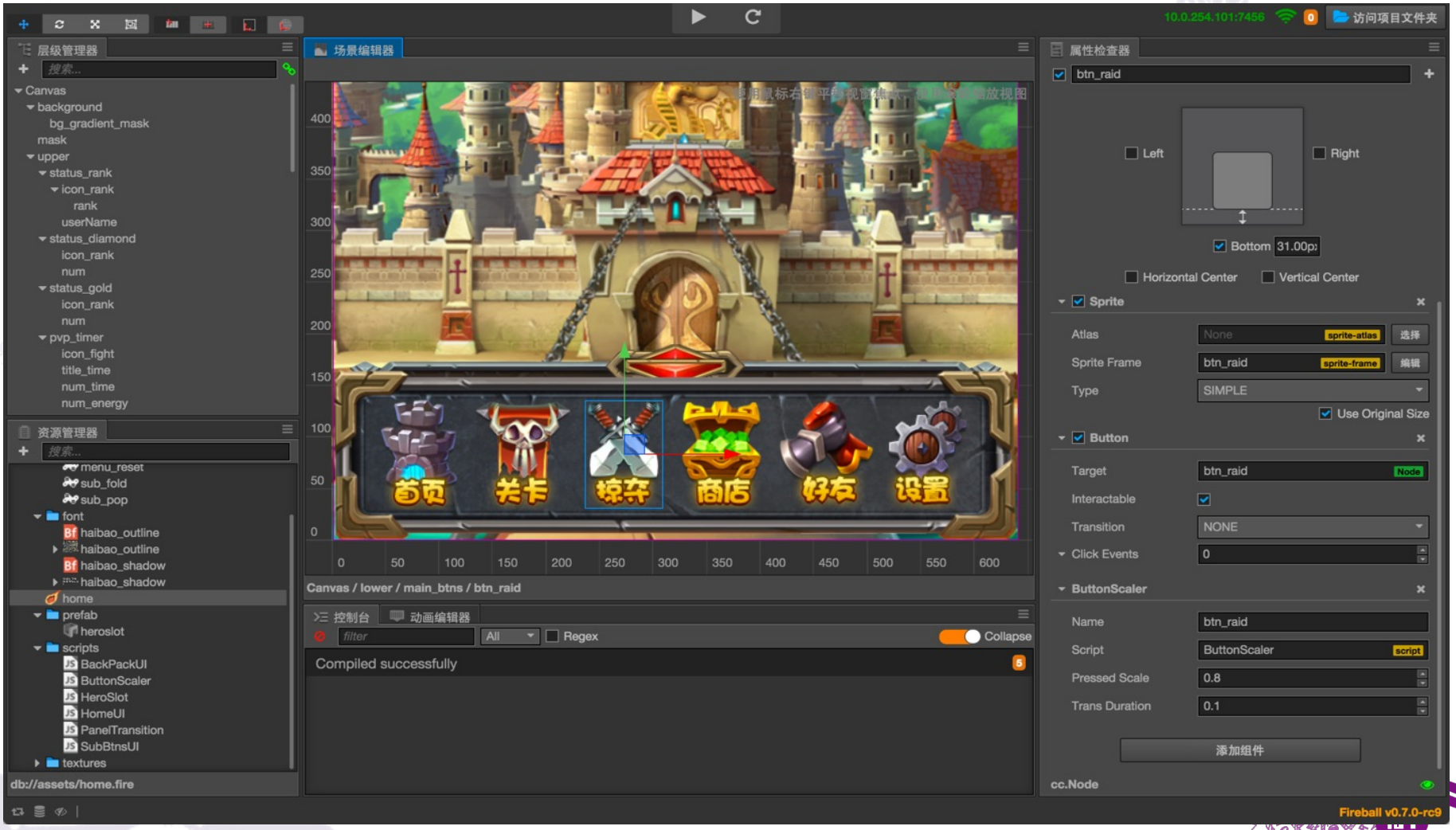


COCOS CREATOR

What is Cocos Creator?

- A web game development engine
- Easy to control object in scene
- Save space, small output package

Cocos Creator



Web Game via Cocos Creator

Web Game via Cocos Creator

Songbringer

Firebase

What is Firebase?

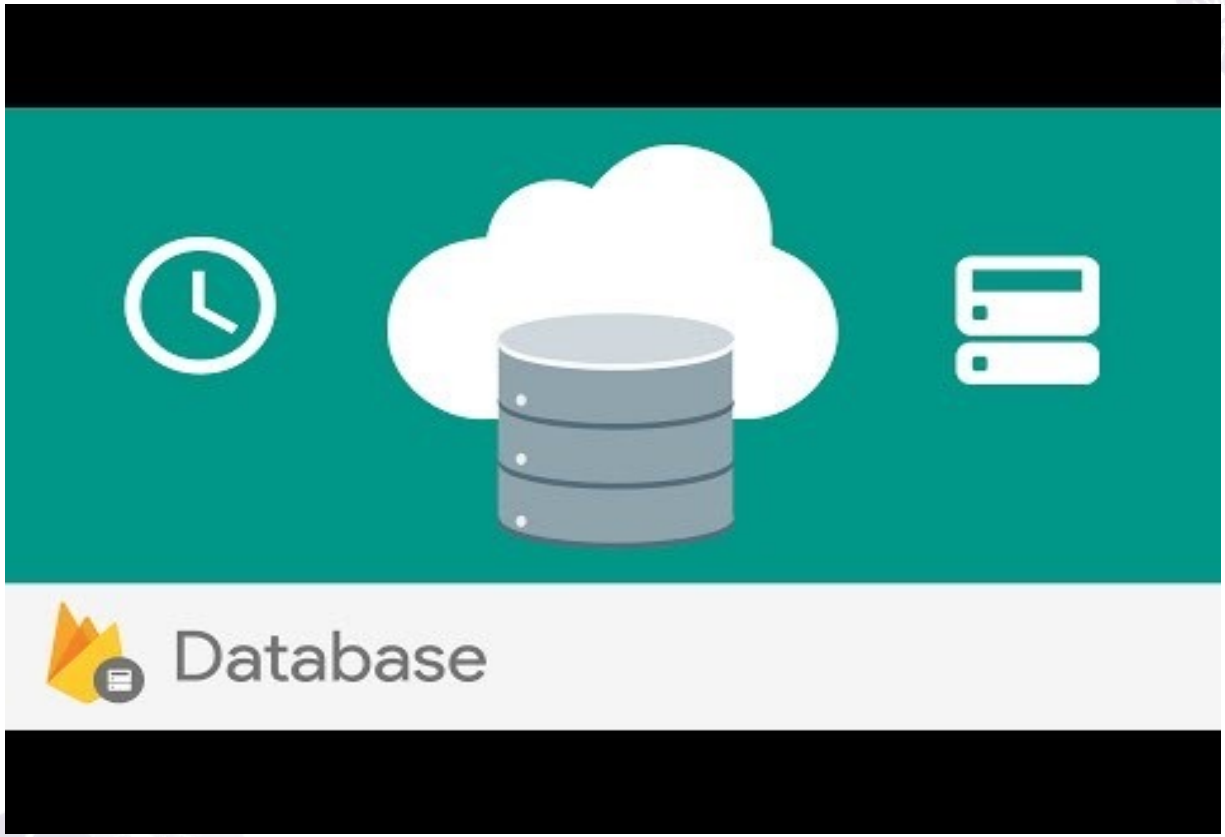
- Firebase is an application cloud development platform built by Google.
- Help developers **quickly set up** backend services in the cloud which effectively shorten the application development time.

Firebase

Why Use Firebase?

- A powerful **real-time database** which makes it an excellent candidate to drive **multi-player games.**
- Full document storage, analytics, hosting, etc.
- Using **JavaScript** API and provide user side high security.

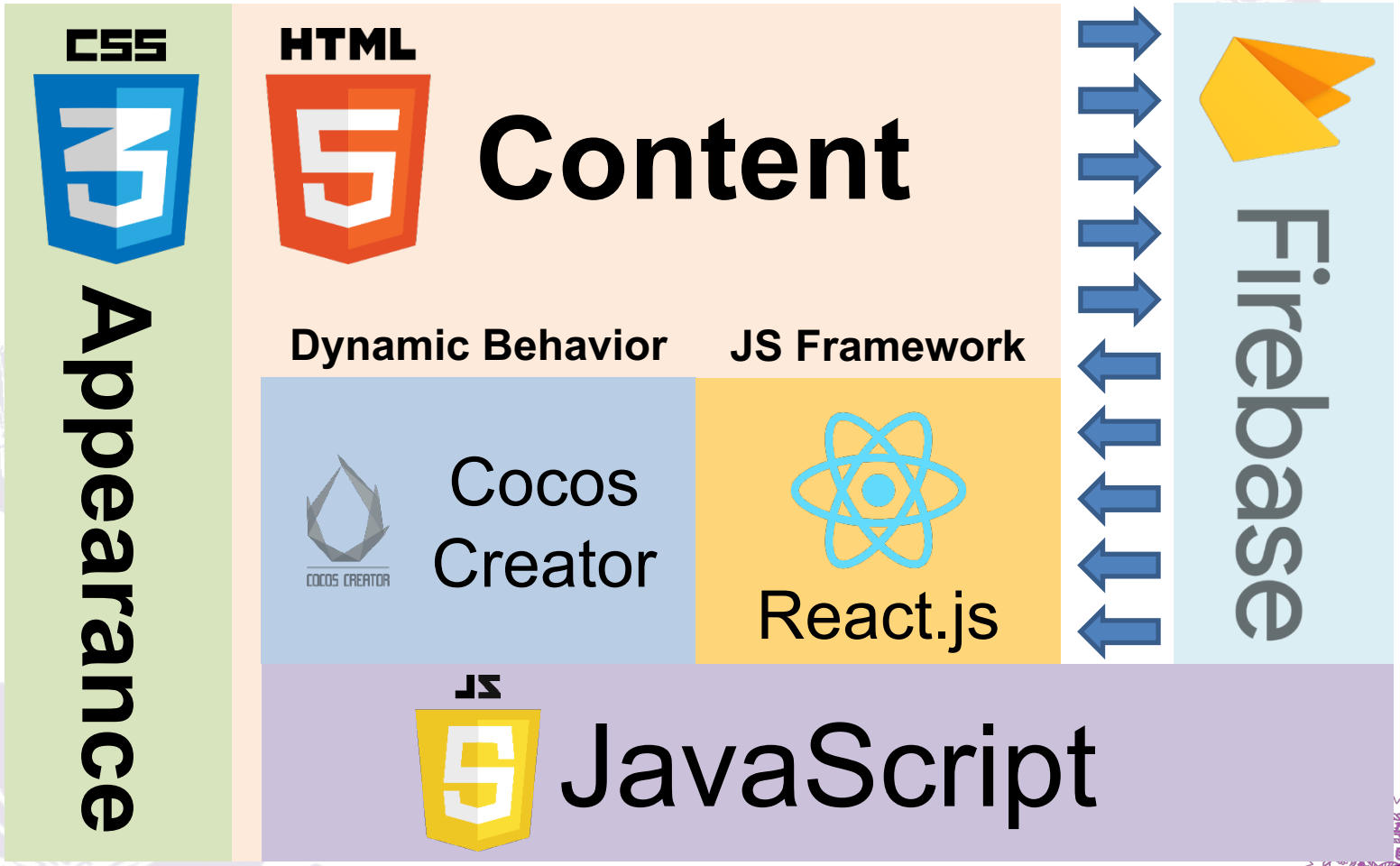
Introducing Firebase Real-time Database



https://www.youtube.com/watch?v=U5aeM5dvUpA

Who are Using Firebase?

Web Application (Web App)




thank you!
Question?