**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP TỐT NGHIỆP**

**XÂY DỰNG GAME SKY ANGEL**

**TRÊN NỀN TẢNG WEB**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ THÔNG TIN**

**Giảng viên hướng dẫn:** Ths. Trần Đức Doanh

**Sinh viên thực hiện:**

Phạm Thị Trúc Linh - MSSV: 1551120024 - Lớp:CN15A

Nguyễn Văn Đức - MSSV: 1551120008 - Lớp: CN15A Hà Thanh Tài - MSSV: 1551120112 - Lớp: CN15B

TP. Hồ Chí Minh, 2019

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………….……………………………………

TP. Hồ Chí Minh, ngày…… tháng……. năm…………

Giảng viên hướng dẫn

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa Công nghệ thong tin – Trường ĐH Giao thông vận tải thành phố Hồ Chí Minh đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Và đặc biệt, chúng em xin tỏ lòng biết ơn sâu sắc đến thầy Trần Đức Doanh, đã tận tình hướng dẫn trong suốt quá trình viết báo cáo thực tập tốt nghiệp. Nếu không có những lời hướng dẫn, dạy bảo của Thầy thì chúng em nghĩ bài thu hoạch này rất khó có thể hoàn thiện được.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, bài báo cáo này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Cuối cùng em kính chúc quý thầy, cô dồi dào sức khỏe và thành công trong sự nghiệp .  
 *Em xin chân thành cảm ơn!*

**LỜI CAM ĐOAN**

Tôi xin cam đoan đồ án Thực tập tốt nghiệp “Xây dựng game sky angel trên nền tảng web” là công trình nghiên cứu của nhóm. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

**MỤC LỤC**

[MỤC LỤC HÌNH ẢNH 8](#_Toc7735143)

[DANH MỤC TỪ VIẾT TẮT 9](#_Toc7735144)

[CHƯƠNG 1: LỜI MỞ ĐẦU 1](#_Toc7735145)

[CHƯƠNG 2: NHỮNG PHẦN SẼ THỰC HIỆN 2](#_Toc7735146)

[CHƯƠNG 3: NGHIÊN CỨU CÁC CÔNG NGHỆ SỬ DỤNG 3](#_Toc7735147)

[3.1. HTML5 (Hypertext Markup Language) 3](#_Toc7735148)

[3.1.1. Tổng quan về HTML5 3](#_Toc7735149)

[3.1.2. Quá trình hình thành và phát triển của HTML5 3](#_Toc7735150)

[3.1.3. Hoạt động của HTML 5](#_Toc7735151)

[3.1.4. Các thành phần HTML được cập nhật 6](#_Toc7735152)

[**3.1.4.1. Cập nhật khai báo** 6](#_Toc7735153)

[3.1.4.2 Cập nhật ngữ nghĩa 8](#_Toc7735154)

[**3.1.4.3 Các thẻ gom nhóm** 12](#_Toc7735155)

[**3.1.4.4 HTML5 FORM** 15](#_Toc7735156)

[**3.1.5.1 Ưu điểm và nhược điểm của HTML5** 19](#_Toc7735157)

[3.2. CSS3 (Cascading Style Sheets) 20](#_Toc7735158)

[3.2.1. Quá trình hình thành và phát triển 20](#_Toc7735159)

[3.2.2. Hoạt động của CSS 21](#_Toc7735160)

[3.2.3. Tổng quan các thuộc tính hay dùng 22](#_Toc7735161)

[**3.2.3.1. Màu sắc** 22](#_Toc7735162)

[**3.2.3.2. Background và Border Background-image** 22](#_Toc7735163)

[**3.2.3.3. Basic box Display** 24](#_Toc7735164)

[**3.2.3.4. Flex-box Flex** 25](#_Toc7735165)

[**3.2.3.5. Các thuộc tính text Text-align** 25](#_Toc7735166)

[**3.2.3.6. Các thuộc tính về font @font-face** 26](#_Toc7735167)

[**3.2.3.7. Các thuộc tính Animation @#keyframe** 27](#_Toc7735168)

[**3.2.3.8. Các thuộc tính Transform Perspective** 28](#_Toc7735169)

[3.3. Javascript 29](#_Toc7735170)

[3.3.1. Tổng quan về Javascript 29](#_Toc7735171)

[3.3.3. Ưu và nhược điểm 30](#_Toc7735172)

[**3.3.3.1. Ưu điểm** 30](#_Toc7735173)

[**3.3.3.2. Nhược điểm** 31](#_Toc7735174)

[3.3.4. Hoạt động của Javascript 31](#_Toc7735175)

[3.3.5. Sự khác biệt giữa Javascript và các ngôn ngữ lập trình khác 33](#_Toc7735176)

[3.3.6. Các thư viện Javascript nổi tiếng hiện nay 34](#_Toc7735177)

[3.4. Lập trình hướng đối tượng (Object Oriented Programing) 35](#_Toc7735178)

[3.4.1. Tổng quan 35](#_Toc7735179)

[3.4.2. Những đặc tính của lập trình hướng đối tượng 36](#_Toc7735180)

[**3.4.2.1. Tính đóng gói (Encapsulation)** 36](#_Toc7735181)

[**3.4.2.2. Tính kế thừa (Inheritance)** 37](#_Toc7735182)

[**3.4.2.3. Tính đa hình (Polymorphism)** 37](#_Toc7735183)

[**3.4.2.4. Tính trừu tượng (Abstraction)** 39](#_Toc7735184)

[CHƯƠNG 4: PHÂN TÍCH NỘI DUNG GAME 40](#_Toc7735185)

[4.1. Tổng quan về game 40](#_Toc7735186)

[4.2. Giao diện của game 41](#_Toc7735187)

[CHƯƠNG 5: KẾT LUẬN 47](#_Toc7735188)

[CHƯƠNG 6: HƯỚNG PHÁT TRIỂN 48](#_Toc7735189)

[CHƯƠNG 7: TÀI LIỆU THAM KHẢO 49](#_Toc7735190)

# MỤC LỤC HÌNH ẢNH

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Số thứ tự** | **Nội dung** | **Số trang** |
| **1** | Hình 3.1.2.1. | Quá trình hình thành và phát triển của HTML. | 12 |
| **2** | **Hình 3.3.2.1.** | Ba thành phần không thể thiếu của một trang web. | 29 |
| **3** | Hình 3.3.3.1 | 1 Sơ đồ hoạt động của Javascript. | 33 |
| **4** | Hình 3.4.2.1. | Đặc tính của lập trình hướng đối tượng. | 36 |
| **5** | Hình 4.2.1. | Giao diện mở đầu và hướng dẫn chơi game. | 41 |
| **6** | Hình 4.2.2. | Giao diện khi chơi game. | 42 |
| **7** | Hình 4.2.3. | Giao diện khi chơi game. | 43 |
| **8** | Hình 4.2.4. | Các thành phần trong game. | 44 |
| **9** | Hình 4.2.5. | Giao diện game over. | 45 |
| **10** | Hình 4.2.6. | Giao diện hiển thị bảng xếp hạng của người chơi. | 46 |

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên viết tắt** | **Tên đầy đủ** |
| 1 | CERN | European Organisation for Nuclear Research |
| 1 | CNTT | Công nghệ thông tin |
| 2 | CSS | Cascading Style Sheets |
| 3 | ES6 | ECMAScript 6 |
| 4 | HTML | Hyper Text Markup Language |
| 6 | W3C | World Wide Web Consortium |

# 

# CHƯƠNG 1: LỜI MỞ ĐẦU

Khi xã hội ngày càng phát triển, nhu cầu giải trí của mỗi cá nhân cũng được nâng cao, nhất là bộ phận giới trẻ và đặc biệt hơn cả là các bạn học sinh, sinh viên, họ là những chủ nhân tương ai của đất nước. Xã hội hiện đại đòi hỏi ở họ khả năng tư duy, mong đợi sự cố gắng không ngừng của họ để thành công trên con đường học tập và làm việc.

Cũng vì vậy áp lực mà họ phải chịu là rất lớn, và nhu cầu giải trí là vô cùng quan trọng trong cuộc sống, nó giúp cho họ có thể thư giãn sau những giờ học căng thẳng và mệt mỏi, hay tăng cường tinh thần đoàn kết.

Có nhiều hình thức giải trí khác nhau như mạng xã hội, các hoạt động ngoại khóa do nhà trường tổ chức v.v... Và một trong những hình thức giải trí được lựa chọn nhiều nhất đó là các trò chơi điện tử, nó nhanh chóng và có mức độ giải trí cao.

Đối với chúng em, những sinh viên khoa công nghệ thông tin, việc chọn đề tài liên quan đến game là cách để củng cố kiến thức lập trình đã học cùng với những kiến thức thực tế đã trải nghiệm trong thời gian qua và cũng là cách để học hỏi, nâng cao kiến thức bản thân. Vì những lí do đó nên nhóm chúng em đã chọn đề tài “Tìm hiểu HTML5, CSS3, ES6 và xây dựng game Sky Angel”.

Game sẽ được thiết kế bố cục sử dụng HTML (Hyper Text Markup Language) và CSS (Cascading Style Sheets) và phát triển ở phía người dùng sử dụng Javascript và các thư viện mã nguồn mở Javascript khác.

Game sẽ được phát triển ở độ phân giải máy tính bảng với 1024x768 pixel (chiều rộng x chiều cao). Nếu trò chơi được mở trong một màn hình lớn hơn, trò chơi sẽ ở giữa màn hình (theo chiều ngang và chiều dọc).

# CHƯƠNG 2: NHỮNG PHẦN SẼ THỰC HIỆN

Tìm hiểu HTML5, là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Nó dùng để định dạng bố cục, các thuộc tính liên quan đến hiển thị và được hiển thị trên một chương trình đặc biệt gọi là Browser.

Tìm hiểu Javascript, là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc định nghĩa ra, Javascript được sử dụng rộng rãi trong các ứng dụng Website và hỗ trợ hầu hết tất cả các trình duyệt hiện nay như: Google Chrome, Microsoft EDGE, Mozilla Firefox, Opera, …

Tìm hiểu CSS3, là viết tắt của Cascading Style Sheet, được dùng để tìm và định dạng lại các phần tử được tạo ra bởi HTML. Giúp cho các phầ tử này trở nên sinh động hơn về màu sắc, kiểu chữ, viền, hiệu ứng, …

Tìm hiểu về lập trình hướng đối tượng, là một trong những kỹ thuật lập trình rất quan trọng hiện nay. Áp dụng cho hầu hết các ngôn ngữ hiện đại.

Cụ thể về ứng dụng minh họa, nhóm em sẽ xây dựng một webgame nhỏ cho phép người sử dụng tương tác và điều khiển với tốc độ nhanh và trải nghiệm tốt nhất.

# CHƯƠNG 3: NGHIÊN CỨU CÁC CÔNG NGHỆ SỬ DỤNG

## HTML5 (Hypertext Markup Language)

### **3.1.1. Tổng quan về HTML5**

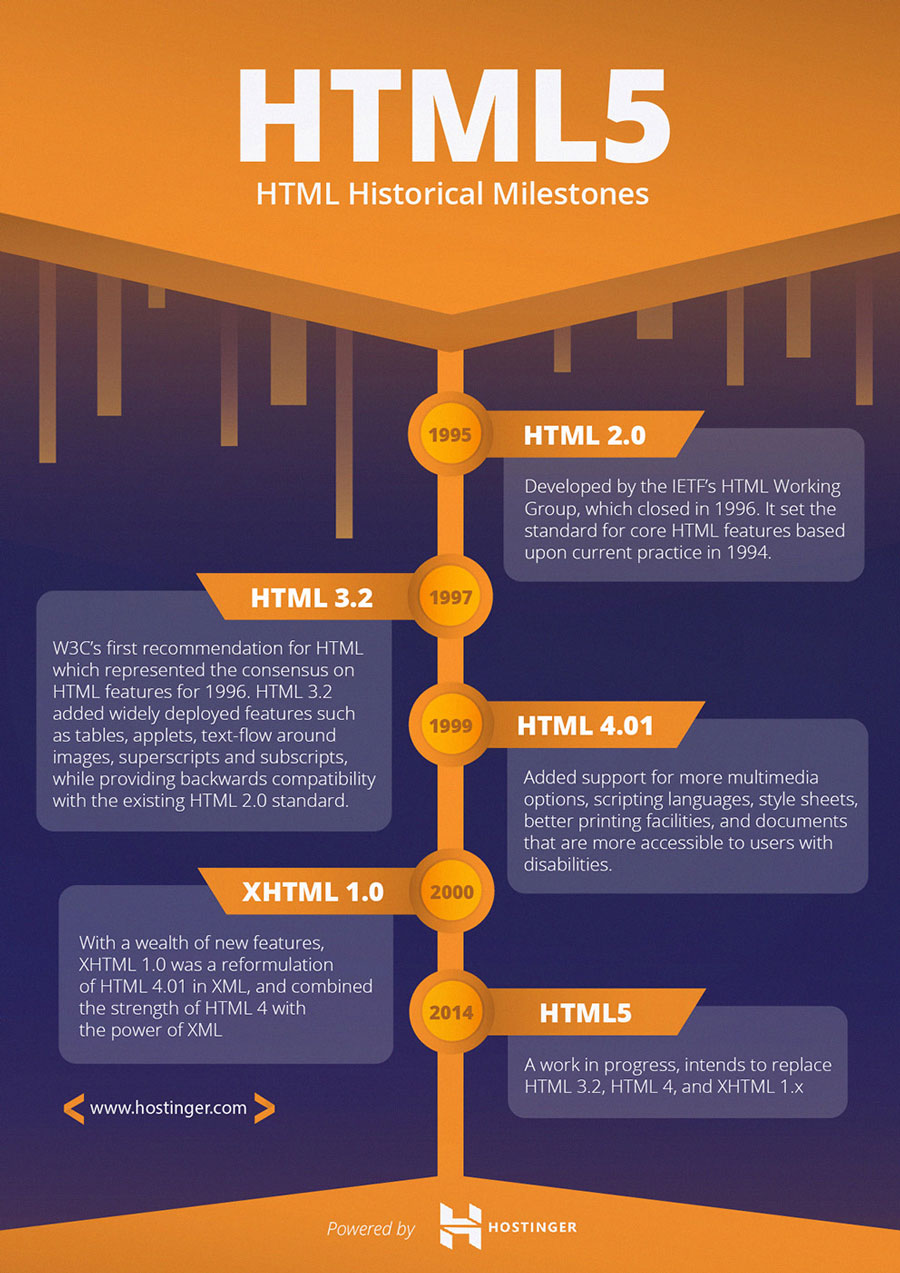
HTML5 là chuẩn mới nhất của HTML, nó có các đặc điểm:

* Cập nhật một số thành phần HTML.
* Có các thành phần và thuộc tính HTML mới (các thành phần HTML5).
* Hỗ trợ đầy đủ CSS3 (chuẩn mới nhất của CSS).
* Có thành phần video và audio.
* Hỗ trợ đồ họa 2D/3D.
* Lưu trữ cục bộ (**local storage**).
* Cơ sở dữ liệu SQL cục bộ (**local SQL database**).

HTML5 được tạo ra để đáp ứng cho sự phát triển web mạnh mẽ hiện nay, hướng đến một khái niệm web mới – web ngữ nghĩa (sematic web). Nên trọng điểm cải tiến các thành phần HTML ở chuẩn mới này là tạo ra các thẻ mang ngữ nghĩa nhiều hơn để bố cục một tài liệu HTML một cách rõ ràng hơn.

### **3.1.2. Quá trình hình thành và phát triển của HTML5**

HTML được sáng tạo ra bởi Tim Berners-Lee, là một nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Anh ta đã nghĩ ra được ý tưởng cho hệ thống hypertext trên nền Internet. Ý trưởng này xuất hiện vào năm 1989 khi Tim làm việc tại văn phòng dịch vụ điện toán ở CERN. Các nghiên cứu vật lý thường yêu cầu sự hợp tác của các nhà khoa học khắp mọi nơi trên thế giới, và để tiện cho việc tra cứu tài liệu, ý tưởng của Tim là tạo ra một nơi để mọt người kết nối vào và từ đó xem các tài liệu các thông qua các liên kết.



*Hình 3.1.2.1 Quá trình hình thành và phát triển của HTML5*

Hypertext có nghĩa là văn bản chứa liên kết, nơi người dùng có thể truy cập liên kết ngay lập tức. Anh cho ra phiên bản đầu tiên của HTML trong năm 1991 bao gồm 18 thẻ HTML. Từ đó, mỗi phiên bản mới của HTML đều có thêm các thẻ mới và thuộc tính mới.

### **3.1.3. Hoạt động của HTML**

HTML Document là các file kết thúc với đuôi **.html hay .htm**. Chúng ta có thể xem chúng bằng cách sử dụng bất kì trình duyệt web nào (như Google Chrome, Safari, Mozilla Firefox, …). Trình duyệt đọc các file HTML này và xuất nột dung lên sao cho người dùng có thể xem được nó.

Thông thường trung bình một Website chứa nhiều trang web HTML khác nhau, ví dụ như: trang chủ, trang liên hệ, trang tìm kiếm, … Tất cả đều là các trang HTML riêng biệt.

Mỗi trang chứa một bộ các thẻ (cũng được gọi là Elements), có thể xem như là việc xây dựng từng khối của một trang web. Nó tạo thành cấu trúc như một cây thư mục bao gồm các thẻ chính để chia bố cục như: header, footer, section, paragraph, và những khối nội dung khác.

Hầu hết các HTML elements đều có thẻ mở và thẻ đóng với cấu trúc

**<tag></tag>**. Bên dưới là một ví dụ nhỏ về cấu trúc của một đoạn code HTML:

<div>

<h1>Hello world!</h1>

<p>This is a paragraph</p>

<img *src*="" *alt*="">

<h2>Complete!!</h2>

</div>

Element ngoài cùng là thẻ **div** (division) bao bọc lấy các thẻ khác ở trong. Dùng để markup cho toàn bộ nội dung lớn. Bên trong bao gồm hai thẻ tiêu đề **h1, h2**, một thẻ văn bản **p** và một thẻ ảnh **img**.

### **3.1.4. Các thành phần HTML được cập nhật**

#### **3.1.4.1. Cập nhật khai báo**

**<!doctype>**

Thẻ <!doctype> cho trình duyệt xác định đây là tài liệu HTML5.

<!doctype *html*>

Một vài ví dụ khai báo kiểu tài liệu HTML4

**<meta>**

- Tag <meta /> cung cấp thông tin dữ liệu về văn bản HTML, thông tin dữ liệu sẽ không được hiển thị trên trình duyệt.

- Tag <meta /> thường được sử dụng để xác định mô tả trang (description), từ khóa (keywords), tác giả của văn bản (author), sửa đổi lần cuối (last modified), và thông tin dữ liệu khác.

- Tag <meta /> có thể được sử dụng bởi các trình duyệt (hiển thị nội dung hoặc tải lại trang), công cụ tìm kiếm (từ khoá), hoặc các dịch vụ web khác.

- Tag <meta /> luôn truyền dưới dạng các cặp tên và giá trị.

<meta *http-equiv*="X-UA-Compatible" *content*="ie=edge">

* <meta *charset*="UTF-8"> //HTML 5

**<script >**

* Tag <script> dùng để xác định một kịch bản (script) phía máy khách (clien), chẳng hạn như javascript.
* Tag <script> đòi hỏi thuộc tính [type](https://hocwebchuan.com/reference/tag/html5/tag_script.php#requireAttr), xác định kiểu [MIME](https://hocwebchuan.com/reference/tag/mime.php) của <script>.
* Tag <script> sử dụng chung cho javascript: thao tác hình ảnh, check form, thay đổi tự động nội dung.
* Ta có thể sử dụng tag <noscript> để hiển thị những nội dung mà trình duyệt bị người dùng vô hiệu hóa hoặc không hỗ trợ <script>.

<script src=”js/jquery-3.2.1.min.js”></script>

<script src=”js/jquery-ui-1.12.1//jquery-ui.min.js”></script>

<script src=”js/main.js”></script>

**<link>**

Tag <link /> xác định mối quan hệ giữa một tài liệu HTML và các tài nguyên bên ngoài.

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="js/jquery-ui-1.12.1//jquery-ui.min.css">

Tag <link /> được dùng phổ biến nhất là link tới tài liệu [css](https://hocwebchuan.com/reference/cssSection/).

Tag <link /> được viết bên trong tag [<head>](https://hocwebchuan.com/reference/tag/html5/tag_head.php)

**<body>**

Thẻ <body> xác định phần thân của tài liệu. Phần tử <body> chứa tất cả nội dung của tài liệu HTML, chẳng hạn như văn bản, siêu liên kết, hình ảnh, bảng, danh sách, v.v

<body>

<div id="wrapper">

<div id="nav">

<div id="btn">

<img src="images/fontup.png" id="fontup" alt="">

</body>

**3.1.4.2 Cập nhật ngữ nghĩa**

**<i> </i>**

* Thẻ <i> được sử dụng để hiển thị nội dung in nghiêng.
* Trong HTML4, thẻ <i> (italic) là thành phần định kiểu chữ in nghiêng.
* Trong HTML5, thẻ <i> thể hiện nội dung với “alternate voice” hoặc “mood”.
* Một số ví dụ trường hợp sử dụng thẻ <i>:
* Tên loài (Taxonomic designation)
* Thuật ngữ kỹ thuật (Technical term)
* Cụm thành ngữ từ ngôn ngữ khác (Idiomatic phrase from another language)
* Phiên dịch (Transliteration)
* Một suy nghĩ (Thought)
* Tên tàu trong các văn bản phương Tây (Ship name in Western texts)

<i *class*="fa-gift"></i>

<i *class*=”fa-graduation-cap"></i>

**<b> </b>**

Thẻ <b> được sử dụng để hiển thị nội dung in đậm.

Trong HTML4, thẻ <b> (**bold**) là thành phần định kiểu chữ in đậm.

Trong HTML5, thẻ <b> thể hiện nội dung với “**stylistically offset**”.

Một số ví dụ trường hợp sử dụng thẻ <b>:

* Từ khóa trong phần tổng quan của tài liệu (Key words in a document abstract)
* Các tên sản phẩm trong một bài đánh giá (Product names in a review)
* Các chữ mang hành động trong các phần mềm tương tác định hướng văn bản (Actionable words in interactive text-driven software)
* Phần dẫn dắt bài viết (Article lead)

**<strong></strong>**

Trong HTML4, thẻ <strong> mang nghĩa nhấn mạnh (strong emphasis).

Trong HTML5, thẻ <strong> mang nghĩa mới là rất quan trọng (strong importance).

**<textarea>**

Thẻ <textarea> xác định điều khiển nhập văn bản nhiều dòng. Một vùng văn bản có thể chứa số lượng ký tự không giới hạn và văn bản hiển thị ở phông chữ có chiều rộng cố định.

<textarea *rows*="4" *cols*="50">

Hello word

</textarea>

**<span>**

Thẻ <span> được sử dụng để nhóm các thành phần nội tuyến trong tài liệu. Thẻ <span> không cung cấp thay đổi trực quan. Thẻ <span> cung cấp một cách để thêm một mắt xích vào một phần của văn bản hoặc một phần của tài liệu.

<span id="time-count">00:00</span>

**<img>**

Thẻ <img> xác định hình ảnh trong trang HTML. Thẻ <img> có hai thuộc tính bắt buộc: src và alt.

<img src="images/fuel-512.png" alt="" id="img-fuel">

**<label>**

Thẻ <label> xác định nhãn cho phần tử <button>, <input>, <meter>,

<output>, <progress>, <select> hoặc <textarea>.

Phần tử <label > không hiển thị bất kỳ điều gì đặc biệt cho người dùng. Tuy nhiên, nó cung cấp một cải tiến về khả năng sử dụng cho người dùng chuột, bởi vì nếu người dùng nhấp vào văn bản trong phần tử <label>, nó sẽ chuyển đổi điều khiển.

Thuộc tính for của thẻ <label> phải bằng thuộc tính id của thành phần liên quan để liên kết chúng lại với nhau.

<label for=”meter”></label>

**<input>**

Thẻ <input> chỉ định vùng nhập đầu vào nơi người dùng có thể nhập dữ liệu.

Các phần tử <input> được sử dụng trong phần tử <form> để khai báo các điều khiển đầu vào cho phép người dùng nhập dữ liệu.

Một trường đầu vào có thể thay đổi theo nhiều cách, tùy thuộc vào loại thuộc tính.

<input type="text" name="name" id="name" >

**<select>**

Phần tử <select> được sử dụng để tạo danh sách thả xuống. Các thẻ

<option> bên trong phần tử <select> xác định các tùy chọn khả dụng trong danh sách.

<select *name*="select" *id*="">

<option *value*="">air craft</option>

<option *value*="">star </option>

</select>

**<time>**

Thẻ <time> xác định ngày / giờ có thể đọc được của con người.

Phần tử này cũng có thể được sử dụng để mã hóa ngày và giờ theo cách dễ đọc bằng máy để các tác nhân người dùng có thể cung cấp để thêm lời nhắc sinh nhật hoặc sự kiện theo lịch vào lịch của người dùng và công cụ tìm kiếm có thể tạo ra kết quả tìm kiếm thông minh hơn.

<time *datetime*="2019-10-10 10:10">Birth day</time>

**<object>**

Thẻ <object> xác định một đối tượng được nhúng trong tài liệu HTML. Sử dụng yếu tố này để nhúng đa phương tiện (như âm thanh, video, tiểu dụng Java, ActiveX, PDF và Flash) trong các trang web của bạn.

<object *width*="400" *height*="400" d*ata*="helloworld.swf"></object>

**<script>**

Thẻ <script> được sử dụng để xác định tập lệnh phía máy khách (JavaScript).

Phần tử <script> chứa các câu lệnh script hoặc nó trỏ đến tệp script bên ngoài thông qua thuộc tính src.

Các hành động phổ biến với JavaScript là thao tác hình ảnh, xác thực mẫu và thay đổi động của nội dung.

#### **3.1.4.3 Các thẻ gom nhóm**

**<section></section>**

Theo định nghĩa của W3C (World Wide Web Consortium) – tổ chức tiêu chuẩn quốc tế chính cho www, thành phần section thể hiện một phân đoạn chung trong một tài liệu hay một ứng dụng.

Hay nói cách khác, thẻ <section> là một thẻ <div> mang ngữ nghĩa, được dùng để nhóm **các nội dung liên quan** (theo chủ đề) lại với nhau.

**<header></header>**

Theo định nghĩa W3C, thành phần header thể hiện một nhóm các hỗ trợ giới thiệu hoặc điều hướng.

Có thể có nhiều header khác nhau trong một trang.

Thường xuất hiện trên đầu tài liệu hoặc phân đoạn, tuy nhiên nó được xác định bởi nội dung của nó hơn là vị trí của nó.

**<footer></footer>**

Theo định nghĩa của W3C, thành phần footer thể hiện cho một footer của nội dung phân đoạn hoặc thành phần phân đoạn gần nhất.

Giống như header, thành phần footer không phụ thuộc vào vị trí. Nó nên mô tả cho nội dung chưa nó.

**<aside></aside>**

Ban đầu, HTML5 định nghĩa thành phần **aside** là “quan hệ tiếp giáp với các nội dung xung quanh mình”.

Bây giờ thành phần **aside** bao phủ nhiều ngữ cảnh:

* Khi được sử dụng bên trong thành phần article, nội dung của **aside** phải liên quan đến **article** đó.
* Khi được sử dụng bên ngoài thành phần **article**, nội dung aside phải đặc biệt liên quan đến trang web.

Một thành phần aside phù hợp khi nó được dùng để thể hiện nội dung không phải là trọng tâm chính của bài viết hay trang web, nhưng nội dung đó vẫn liên quan đến bài viết hay trang web đó.

**<nav></nav>**

Theo định nghĩa của W3C, thành phần **nav** (**major navigation**) thể hiện một đoạn của một trang mà nó liên kết đến những trang khác hoặc những phần khác trong trang đó: một phân đoạn với các liên kết điều hướng.

**<article></article>**

Theo định nghĩa của W3C, thành phần article thể hiện một thành phần hoàn chỉnh (hay khép kín) trong một tài liệu, trang web hay ứng dụng và theo nguyên tắc thì nó có thể được phân phối hoặc tái sử dụng một cách độc lập.

Thành phần article là một loại section. Nó được dùng cho nội dung hoàn chỉnh.

Một số ví dụ trường hợp sử dụng thẻ <article>:

* Một bài đăng trên blog
* Một mẩu tin tức
* Một bình luận trên một bài đăng
* Một bài đánh giá

**<main></main>**

Theo định nghĩa của W3C, thành phần main thể hiện nội dung chính của phần thân của một tài liệu hoặc ứng dụng.

Và cũng theo W3C, vùng nội dung của main bao gồm các nội dung liên quan trực tiếp hoặc mở rộng dựa trên chủ đề chính của một tài liệu hoặc chức năng chính của một ứng dụng.

**<figure></figure>**

Theo W3C, thành phần figure thể hiện một đơn vị của nội dung (có thể có caption) thường được tham chiếm như một đơn vị duy nhất từ mạch tài liệu, và có thể bỏ khỏi mạch tài liệu mà không làm ảnh hưởng đến nghĩa của tài liệu.

**<figcaption></figcaption>**

Theo W3C, thành phần figcaption (**figure caption**) thể hiện cho một caption hoặc legend của một figure.

**<time></time>**

Theo W3C, thành phần time thể hiện hoặc thời gian trên đồng hồ, hoặc ngày tháng trong lịch (có thể có thêm thời gian và múi giờ).

#### **3.1.4.4 HTML5 FORM**

**<form>**

Thẻ <form> được sử dụng để tạo biểu mẫu HTML cho đầu vào của người dùng. Phần tử <form> có thể chứa một hoặc nhiều phần tử biểu mẫu sau:

* + - * + [<input>](https://www.w3schools.com/tags/tag_input.asp)
        + [<textarea>](https://www.w3schools.com/tags/tag_textarea.asp)
        + [<button>](https://www.w3schools.com/tags/tag_button.asp)
        + [<select>](https://www.w3schools.com/tags/tag_select.asp)
        + [<option>](https://www.w3schools.com/tags/tag_option.asp)
        + [<optgroup>](https://www.w3schools.com/tags/tag_optgroup.asp)
        + [<fieldset>](https://www.w3schools.com/tags/tag_fieldset.asp)
        + [<label>](https://www.w3schools.com/tags/tag_label.asp)

**<datalist></datalist>**

W3C định nghĩa thành phần datalist định nghĩa một tập các thành phần option thể hiện các lựa chọn.

Các lựa chọn được khai báo bằng thẻ **<option>** với giá trị là thuộc tính **value**.

HTML5 cung cấp nhiều loại input mới:

* search
* email
* url
* tel
* number
* range
* date
* month
* week
* time
* datetime
* datetime-local
* color

***Lưu ý:*** Các loại input này không được hỗ trợ đầy đủ ở các trình duyệt khác nhau. Nếu trình duyệt không hỗ trợ loại input thì nó sẽ chuyển về loại mặc định là text.

**Các thuộc tính mới của biểu mẫu**

HTML5 cung cấp nhiều thuộc tính của biểu mẫu:

* placeholder
* autofocus
* autocomplete
* required
* pattern
* list
* multiple
* novalidate
* formnovalidate
* form
* formaction
* formenctype
* formmethod
* formtarget

**Placeholder:**

Thuộc tính **placeholder** cho phép bạn xác định một thông điệp hiển thị bên trong trường input, thông điệp ẩn đi khi người dùng bắt đầu nhập, và hiện trở lại nếu trường input vẫn còn trống và bị mất focus (Để con nháy đi chỗ khác).

**Autofocus**

Thuộc tính **autofocus** sẽ giúp tự động focus (đưa con nháy) vào trường input xác định khi trang web được nạp lên trình duyệt.

**Required**

Nếu bạn thêm thuộc tính **required** vào input, khi biểu mẫu được gửi đi (submit), người dùng sẽ nhận được thông báo lỗi nếu trường nhập còn trống.

Thẻ HTML5 **<video></video>** quy định một phương thức chuẩn để nhúng một video vào một trang web.

Thuộc tính **controls** thêm trình điều khiển video như phát, tạm dừng và âm lượng.

Thuộc tính **autoplay** giúp video tự động phát khi trang web được nạp lên trình duyệt.

Thuộc tính kích thước **width** và **height** nên được thêm vào để trình duyệt biết kích thước của video. Nếu không có 2 thuộc tính này thì khi video được nạp lên thì trang web sẽ bị thay đổi (hay flicker).

Đoạn văn bản nằm giữa cặp thẻ <video> </video> chỉ hiển thị trên trình duyệt không hỗ trợ thành phần video.

Thẻ **<source>** dùng để xác định file video nguồn “**src**” và định dạng video “**type**”.

***<source> không có thẻ đóng.***

Nhiều thành phần source có thể liên kết đến nhiều tập tin video khác nhau. Trình duyệt sẽ sử dụng thẻ có định dạng video được nhận diện đầu tiên.

Hiện tại có định dạng video được hỗ trợ: MP4, WebM và Ogg.

Cũng giống như video, phương pháp chuẩn để nhúng audio vào trang web chỉ có từ HTML5, bằng cách dùng thẻ **<audio></audio>**.

#### **3.1.5. Ưu điểm và nhược điểm của HTML5**

##### **3.1.5.1. Ưu điểm**

* + - * + Ngôn ngữ được sử dụng rộng rãi ngày nay và có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng cực lớn.
        + Sử dụng mượt mà trên hầu hết mọi trình duyệt.
        + Có quá trình học đơn giản và trực tiếp.
        + Mã nguồn mở và hoàn toàn miễn phí.
        + Markup gọn gàng và đồng nhất.
        + Chuẩn chính của web được vận hành bởi World Wide Web Consortium (W3C).
        + Dễ dàng tích hợp với ngôn ngữ backend như PHP.

##### **3.1.5.2. Khuyết điểm**

* + - * + Được dùng chủ yếu cho web tĩnh. Đối với các tính ăng động, phải sử dụng thêm Javascript hoặc ngôn ngữ backend như PHP.
        + Có thể thực thi một số logic nhất định cho người dùng. Vì vậy, hầu hết các trang đều cần được tạo riêng biệt kể cả header và footer.
        + Một số trình duyệt chậm hỗ trợ tính năng mới.
        + Khó kiểm soát cách thực thi của trình duyệt (ví dụ những trình duyệt cũ không hiển thị được thẻ mới).

## CSS3 (Cascading Style Sheets)

### **3.2.1. Quá trình hình thành và phát triển**

CSS là một ngôn ngữ tạo phong cách và định kiểu cho những thành phần được viết dưới dạng ngôn ngữ đánh dấu như HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người phát triển web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang web bằng cách điều khiển bố cục, màu sắc và font chữ.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản: HTML không được thiết kết để gắn tag giúp định dạng trang web. Ta chỉ có thể dùng nó để “dánh dấu” văn bản.

Những thẻ như **<font>** được ra mắt trong HTML phiên bản 3.2, nó gây ra nhiều rắc rối cho lập trình viên. Vì Website có thể sử dụng nhiều font khác nhau, màu sắc và phong cách cũng khác nhau. Để viết lại code cho cả trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo ra bởi W3C để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS là rất mật thiết. HTML là ngôn ngữ đánh dấu (nền tảng cảu một Website) và CSS định hình phong cách (tất cả những gì tạo nên giao diện cho trang web), do đó chúng không thể tách rời nhau. CSS về lý thuyết thì sự có mặt của nó là không quan trọng, nhưng khi đó, trang web sẽ chỉ là một trang chứa văn bản mà không có gì khác.

### **3.2.2. Hoạt động của CSS**

CSS sử dụng cấu trúc tiếng Anh đơn giản để tạo ra một bộ các quy tắc có thể tận dụng. Như đã nhắc đến ở trên, HTML không được dùng để tạo phong cách, nó chỉ dùng để tạo cơ sở, và chia bố cục từng phần cho trang web. CSS sẽ đảm nhiệm phần đó. CSS rất đơn giản, cấu trúc gồm 2 phần: phần chọn thẻ và phần khai báo style cho thẻ đó. Phần chọn thẻ sẽ trỏ về thẻ HTML muốn tạo style, phần khai báo style sẽ gồm một hay nhiều dòng khai báo cách nhau bởi dấu chấm phẩy. Mỗi khai báo phải được để trong cập dấu ngoặc nhọn.

html, body {

background-color: #fff; color: #636b6f;

font-family: 'Nunito', sans-serif; font-weight: 200;

height: 100vh; margin: 0;

}

.*full-height* { height: 100vh;

}

.*flex-center* {

align-items: center; display: flex;

justify-content: center;

}

.*position-ref* { position: relative;

}

### **3.2.3. Tổng quan các thuộc tính hay dùng**

#### **3.2.3.1. Màu sắc**

**Color**

Đặt màu sắc cho văn bản

color: #212529;

**Background-color**

Đặt màu sắc nền cho phần tử

background-color: #fff;

**Opacity**

Đặt độ trong suốt cho thành phần đó

opacity: 1;

#### **3.2.3.2. Background và Border Background-image**

Đặt hình nền cho phần tử:

background-image: url("data:image/svg+xml,%3csvg [xmlns='http://www.w3.org/2000/svg'](http://www.w3.org/2000/svg%27) viewBox='0 0 8 8'%3e%3cpath fill='%23fff' d='M6.564.75l-3.59 3.612-1.538-

1.55L0 4.26 2.974 7.25 8 2.193z'/%3e%3c/svg%3e");

**Background-position**

Chỉ định vị trí của hình nền

background-position: 1rem 0;

**Background-repeat**

Xác định thành phần có được lặp lại không

background-repeat: no-repeat;

**Border**

Thiết lập đường viền cho văn bản

border: solid transparent;

**Border-color**

Xác định màu cho đường viền

border-color: #28a745;

**Boder-width**

Thiết lập độ rộng cho cả bốn đường viền

border-width: 1px 0;

**Border-radius**

Thiết lập đường bo các góc cho thành phần

border-radius: 0.3rem;

**Box-shadow**

Gắn một hoặc nhiều đổ bóng vào phần tử.

box-shadow: 0 0 0 0.2rem rgba(40, 167, 69, 0.25);

#### **3.2.3.3. Basic box Display**

Xác định loại hiển thị của thành phần\

display: block;

**Position**

Thuộc tính vị trí xác định loại phương thức định vị được sử dụng cho một phần tử (tĩnh, tương đối, tuyệt đối, cố định hoặc dính).

position: relative;

**Margin**

Các thuộc tính căn lề CSS được sử dụng để tạo không gian xung quanh các phần tử, bên ngoài bất kỳ đường viền xác định nào.

Với CSS, bạn có toàn quyền kiểm soát việc căn lề. Có các thuộc tính để căn lề cho mỗi bên của một phần tử (trên cùng, bên phải, dưới cùng và bên trái).

margin-top: 0; margin-bottom: 1rem;

**Padding**

Thiết lập vùng đệm cho thành phần

padding: 0.2rem 0.4rem; padding-top: 0.75rem; padding-bottom: 0.75rem;

**Visibility**

Xác định phần tử có được thấy hay không

visibility: visible *!important*;

**z-index**

xác định vị trí xếp chồng của các thành phần có thuộc tính position.

z-index: 5;

#### **3.2.3.4. Flex-box Flex**

Xác định lại độ dài của các mục hoặc tính tương đối của nó đối với các thành phần khác trong flexbox.

flex: 0 0 auto;

**Flex-direction**

Xác định hướng sắp đặt của các thành phần.

flex-direction: column;

**Justify-content**

Xác định sự căn chỉnh của các thành phần bên trong khối với nhau khi các thành phần này không chiếm hết khoảng cách có sẵn.

justify-content: center;

**Order**

Thiết lập thứ tự hiển thị của thành phần so với các thành phần còn lại.

order: 2;

#### **3.2.3.5. Các thuộc tính text Text-align**

Xác định căn chỉnh nội dung theo chiều ngang

textalign:center;

**Line-height**

Xác định độ cao của dòng.

line-height: inherit;

**Text-indent**

Xác định khoảng cách thụt đầu dòng của mỗi đoạn văn bản.

text-indent: -999px;

**Text-transform**

Xác định các định dạng chữ (in hoa, thường, viết hoa chữ cái đầu).

text-transform: uppercase *!important*;

**Word-spacing**

Tăng giảm khoảng cách các từ trong nội dung

word-spacing: normal;

#### **3.2.3.6. Các thuộc tính về font @font-face**

Khai báo tên cùng đường lẫn tới thư mục lưu font chữ

*@font-face* {

font-family: myFirstFont;

src: url(sansation\_light.woff);

}

**Font**

Đặt tất cả các thuộc tình font trong lời khai báo.

font: 0/0 a;

**Font-family**

Xác định font chữ cho văn bản

font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";

**Font size**

Xác định kích thước font chữ cho đoạn văn bản

font-size: 1rem;

**Font-weight**

Xác định độ đậm cho font chữ .

font-weight: 400;

#### **3.2.3.7. Các thuộc tính Animation @#keyframe**

Xác định mã animation và hành động của animation

*@keyframes* progress-bar-stripes { from {

background-position: 1rem 0;

}

to {

background-position: 0 0;

}

}

**Animation**

Là thuộc tính viết tắt cho các thuộc tính animation

animation: progress-bar-stripes 1s linear infinite;

**Animation-name**

Xác định tên của @keyframes animation

animation-name: some-animation-name;

**Animation-duration**

Xác định thời gian để hoàn thành một chu kỳ của animation

animation-duration: 10s, 35s, 230ms;

**Animation-delay**

Xác định độ trễ của animation

animation-delay: 2s;

#### **3.2.3.8. Các thuộc tính Transform Perspective**

Xác định chiều sâu

perspective: 100px;

**Transform**

Xác đinh chuyển đổi 2D hoặc 3D lên một thành phần.

transform: translateX(0.75rem);

**Transform-duration**

Xác định thời gian để chuyển đổi thực hiện xong

transition-duration: 5s

**Transfom-delay**

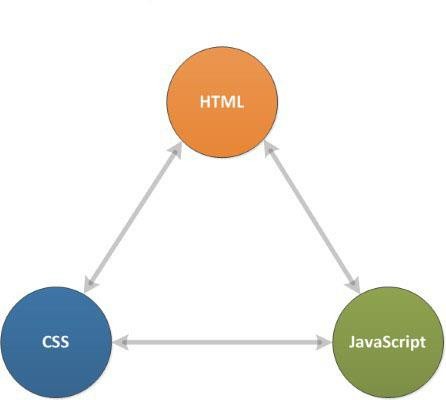
Xác định độ trễ của chuyển đổi

transition-delay: 2s;

## Javascript

### **3.3.1. Tổng quan về Javascript**

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra. Javascript được sử dụng rộng rãi trong các ứng dụng web. Javascript được hỗ trợ ở hầu hết các trình duyệt web phổ biến hiện nay như: Google Chome, Mozilla Filrefox, Safari, Opera, … thậm chí ngay cả trên các thiết bị di động như điện thoại.

Javascript là ngôn ngữ mang lại sự sinh động của Website. Nó khác với HTML (chuyên cho nội dung, bố cục) và CSS (chuyên cho phong cách các thẻ HTML) và khác hẳn với PHP (PHP chạy trên server chứ không chạy dưới máy client).

*Hình 3.3.2.1 Ba thành phần không thể thiếu của một trang web*

**3.3.2. Quá trình phát triển**

Javascript được tạo ra trong vòng ba mươi ngày bởi Brandan Eich, một nhân viên của Netscape vào tháng 9 năm 1995. Được đặt tên là Mocha, tên của nó được đổi thành Mona rồi LiveScript trước khi thât sự trở thành Javascript nổi tiếng như bây giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế. Nhưng nó tiếp tục phát triển theo thời gian, nhờ một phần vào cộng đồng các lập trình viên đã liên tục làm việc với nó.

Trong năm 1996, Javascript được chính thức đặt tên là ECMAScript. ECMAScript 2 được phát hành vào năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành Javascript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính.

Javascript tiếp tục phát triển, chỉ trong 20 năm nó từ một ngôn ngữ lập trình đã trở thành một phần không thể thiếu trong bọ công cụ lập trình web.

### **3.3.3. Ưu và nhược điểm**

#### **3.3.3.1. Ưu điểm**

Javascript có rất nhiều ưu điểm vượt trội so với các đối thủ, đặc biệt trong các trường hợp thực tế:

* + - * + Chúng ta không cần một compiler vì trình duyệt web có thể biên dịch nó.
        + Nó dễ học hơn các ngôn ngữ khác.
        + Lỗi dễ phát hiện và vì thế dễ sửa hơn.
        + Nó có thể được gắn trên một số Element của treang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới.
        + Javascript hoạt động trên nhiều trình duyệt và nền tảng.
        + Giúp Website tương tác tốt hơn với người sử dụng, nâng cao tính thân thiện với người dùng.
        + Nhanh và nhẹ hơn các ngôn ngữ khác.

#### **3.3.3.2. Nhược điểm**

Mọi ngôn ngữ lập trình đều có các khuyết điểm. Một phần vì ngôn ngữ đó khi phát triển đến một mức độ như Javascript thì sẽ thu hút lượng lớn hacker, scammer, những người có tư tưởng tìm kiếm các lỗ hổng bảo mật để lơi dụng nó.

* + - * + Dễ bị khai thác.
        + Có thể được dùng để thực thi mã độc trên máy tính của nạn nhân.
        + Các phiên bản mới không hỗ trợ các trình duyệt cũ.
        + Javascript code snippet lớn.
        + Có thể bị triển khai khác nhau tùy từng thiết bị dẫn tới việc không đồng nhất.

### **3.3.4. Hoạt động của Javascript**

JavaScript thường được nhúng trực tiếp vào một trang web hoặc được tham chiếu qua file **.js** riêng. Nó là ngôn ngữ phía client, tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập.

Tải Javascript vào trang web: Cách phổ biến nhất để tải JavaScript trong web page là sử dụng thẻ **<script>**. Tùy thuộc vào nhu cầu, chúng ta có thể sử dụng một trong các phương pháp sau:

* + - * Tải một tệp javascript bên ngoài vào một trang web như sau:

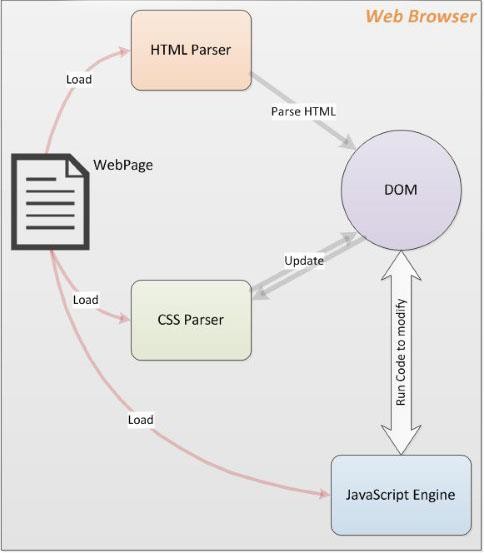
*<script type="text/javascript" src="/path/to/javascript"></script>*

* + - * Nhúng trực tiếp vào HTML:

*<script type="text/javascript"> alert ("Page is loaded"); </script>*

Khi web browser tải một web page, HTML parser bắt đầu phân tích cú pháp HTML code và tạo DOM. Bất cứ khi nào trình phân tích cú pháp gặp một chỉ thị CSS hoặc JavaScript (inline hoặc externally loaded), nó sẽ được chuyển giao cho CSS parser hoặc JavaScript engine theo yêu cầu. JavaScript engine tải các tệp JavaScript bên ngoài và inline code, nhưng không chạy mã ngay lập tức, mà đợi HTML và CSS phân tích hoàn tất.

Khi điều này được thực hiện, JavaScript được thực thi theo thứ tự chúng được tìm thấy trên trang web: các biến và hàm được xác định, các lời gọi hàm được thực thi, trình xử lý sự kiện được kích hoạt, ... Các hoạt động này dẫn đến việc DOM được cập nhật bởi JavaScript và được hiển thị ngay lập tức bởi trình duyệt.

*Hình 3.3.3.1 Sơ đồ hoạt động của Javascript*

### **3.3.5. Sự khác biệt giữa Javascript và các ngôn ngữ lập trình khác**

Lý do vì sao JavaScript là một trong các ngôn ngữ lập trình phổ biến nhất là nó rất linh hoạt. Trên thực tế, có nhiều lập trình viên chọn nó làm ngôn ngữ chính và chỉ sử dụng các ngôn ngữ khác trong danh sách bên dưới:

|  |  |
| --- | --- |
| **JavaScript** | JavaScript hoặc JS sẽ giúp tăng tính tương tác trên website. Script này chạy trên các trình duyệt của người dùng thay vì trên server và thường sử dụng thư vuiên của bên thứ 3 nên có thể tăng thêm chức năng cho website mà không phải code từ đầu. |
| **HTML** | Viết tắt của “Hypertext Markup Language”, HTML là một trong số các ngôn ngữ lập trình phổ biến nhất trên web và xây dựng nên các khối chính của một trang web. Ví dụ về HTML tags là **<p>** cho đoạn văn và  **<img>** cho hình ảnh. |
| **PHP** | PHP là ngôn ngữ phía server, khác với JavaScript chạy trên máy client. Nó thường được sử dụng trong các hệ quản trị nội dung nền PHP như WordPress, nhưng cũng thường được dùng với lập trình back-end và có thể tạo ra kênh truyền thông tin hiệu quả nhất tới và từ database. |
| **CSS** | CSS viết tắt của “Cascading Style Sheets”, nó giúp webmaster xác định styles và định nghĩa nhiều loại nội dung. Bạn có thể làm vậy thủ công với mọi yếu tố trong HTML, nhưng nếu vậy bạn sẽ cứ lặp đi lặp lại thành phần đó mà bạn dùng ở nhiều nơi khác nhau. |

### **3.3.6. Các thư viện Javascript nổi tiếng hiện nay**

Những ứng dụng to lớn của Javascript khiến người ta không thể quên nó được.

Hiện nay có rất nhiều libraries và framework được viết từ Javascript như:

* + - * **AngularJS:** Một thư viện để xây đựng ứng dụng Web Single Page.
      * **NodeJS:** Một thư viện được pát triển phía server dùng để xây dựng ứng dụng thời gian thực.
      * **ReactJS:** Thư viện để xây dựng ứng dung Web Single Page.
      * **React Native:** Xây dựng ứng dụng mobie.
      * **Jquery:**Thư viện giúp xây dựng các chức năng bằng Javascript dễ dàng hơn.

## 3.4. Lập trình hướng đối tượng (Object Oriented Programing)

### **3.4.1. Tổng quan**

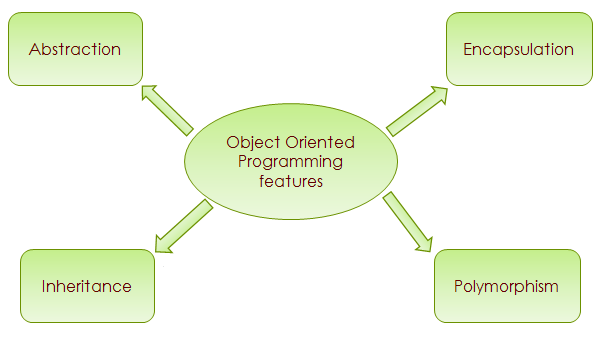
Lập trình hướng đối tượng (OOP) là một trong những kỹ thuật lập trình rất quan trọng hiện nay. Nó được áp dụng ở hầu hết các ứng dụng thực tế xây dựng tại các doanh nghiệp. Hầu hết các ngôn ngữ lập trình và Framework lập trình phổ biến hiện nay như Java, Javascript, PHP, .NET, Ruby, … Đều hỗ trợ lập trình hướng đối tượng. Các lập trình viên đa phần đã được học về lập trình hướng đối tượng ở trường đại học.

Lập trình hướng đối tượng là một kỹ thuật lập trình cho phép lập trình viên tạo ra các đối tượng trong code của mình để trừu tượng hóa các đối tượng thực tế trong cuộc sống. Hướng tiếp cận này hiện đang rất thành công và trở thành một khuôn mẫu trong phát triển phần mềm.

Khi phát triển ứng dụng sử dụng OOP, chúng ta sẽ định nghĩa các lớp (class) để mô hình các đối tượng thực tế. Trong ứng dụng, các lớp này được khởi tạo thành các đối tượng và trong suốt quá trình ứng dụng chạy, các phương thức (method) của các đối tượng này sẽ được gọi.

Lớp định nghĩa đối tượng sẽ bao gồm những thuộc tính và phương thức đặc trưng cho đối tượng đó. Một đối tượng chỉ là một thể hiện của lớp. Các lớp tương tác với nhau qua các public API là các phương thức và thuộc tính public của nó.

### **3.4.2. Những đặc tính của lập trình hướng đối tượng**



*Hình 3.4.2.1 Đặc tính của lập trình hướng đối tượng*

#### **3.4.2.1. Tính đóng gói (Encapsulation)**

Tính đóng gói tức là quy tắc yêu cầu trạng thái bên trong của một đối tượng được bảo vệ và tránh truy cập được từ code bên ngoài (tức là code bên ngoài không thể trực tiếp nhìn thấy và thay đổi trạng thái của đối tượng đó). Bất cứ truy cập nào tới trạng thái bên trong này bắt buộc phải thông qua một public API để đảm bảo trạng thái của đối tượng luôn hợp lệ bởi vì các public API chịu trách nhiệm thực hiện kiểm tra tính hợp lệ cũng như trình tự cập nhật trạng thái của đối tượng đó.

Nói chung trạng thái đối tượng không hợp lệ thường do: chưa được kiểm tra tính hợp lệ, các bước thực hiện không đúng trình tự hoặc bị bỏ qua nên trong OOP có một quy tắc quan trọng cần nhớ đó là phải luôn khai báo các trạng thái bên trong của đối tượng là private và chỉ cho truy cập qua các public/protected method/property. Khi sử dụng các đối tượng ta không cần biết bên trong nó làm việc như thế nào, ta chỉ cần biết các public API là gì và điều này đảm bảo những gì thay đổi đối tượng sẽ được kiểm tra bởi các quy tắc logic bên trong, tránh đối tượng bị sử dụng không chính xác.

Nguyên lý đóng gói như thế này ở đâu ta cũng có thể bắt gặp ví dụ như thiết kế viên thuốc, chúng ta chỉ biết nó chữa bệnh này, bệnh kia và một số thành phần chính còn cụ thể bên trong nó có những gì thì hoàn toàn không biết.

#### **3.4.2.2. Tính kế thừa (Inheritance)**

Khi bắt đầu xây dựng ứng dụng chúng ta sẽ bắt đầu việc thiết kế các lớp, thông thường chúng ta sẽ thấy có trường hợp một số lớp dường như có quan hệ với những lớp khác, chúng có những đặc tính khá giống nhau. VD: 3 lớp AndroidPhone, IPhone, WindowsPhone

Mỗi lớp đều đại diện cho một loại smartphone khác nhau nhưng lại có những thuộc tính giống nhau. Thay vì sao chép những thuộc tính này, sẽ hay hơn nếu ta đặt chúng ở một nơi có thể dùng bởi những lớp khác. Điều này được thực hiện bởi tính kế thừa trong OOP: chúng ta có thể định nghĩa lớp cha – base class (trong trường hợp này là Smartphone ) và có những lớp con kế thừa từ nó (derived class), tạo ra một mối quan hệ cha/con.

Bây giờ, các lớp con có thể kế thừa 3 thuộc tính từ lớp cha. Nếu các chức năng của lớp cha đã được định nghĩa đầy đủ thì lập trình viên sẽ không phải làm bất cứ việc gì ở lớp con. Còn nếu một lớp con muốn chức năng khác so với định nghĩa ở lớp cha thì nó có thể ghi đè (override) chức năng đã được định nghĩa trên lớp cha này.

#### **3.4.2.3. Tính đa hình (Polymorphism)**

Với đa số lập trình viên thì tính Kế thừa và Đóng gói trong OOP khá dễ hiểu còn tính Đa hình khi mới tiếp cận sẽ thấy khó hiểu hơn một chút. Tuy nhiên đây lại là một tính chất có thể nói là chứa đựng hầu hết sức mạnh của lập trình hướng đối tượng. Hiểu một cách đơn giản: Đa hình là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

Ví dụ như ở phần trên, mỗi một smartphone kế thừa từ lớp Smartphone nhưng có thể lưu trữ dữ liệu trên cloud theo những cách khác nhau:

AndroidPhone lưu trữ bằng Google Drive Iphone lưu trên iCloud WindowsPhone sử dụng SkyDrive.

Bởi vì tất cả đều là Smartphone nên nếu ta viết một hàm dùng kiểu Smartphone làm tham số thì khi gọi hàm ta có thể truyền vào một đối tượng kiểu AndroidPhone, Iphone hoặc WindowsPhone bởi vì chúng đều kế thừa từ lớp Smartphone nên được chấp nhận (hiểu nôm na một AndroidPhone, Iphone, WindowsPhone cũng là một Smartphone). Bên cạnh đó hàm này thậm chí không cần quan tâm smartphone nào được truyền vào do nó chỉ cần biết đối tượng đang xử lý ở đây là Smartphone với những public method/property đã được định nghĩa. Nếu các lớp con không định nghĩa lại (override) phương thức CloudStore() thì phương thức CloudStore() trên lớp cha (Smartphone) sẽ được gọi. Còn nếu lớp con override lại phương thức CloudStore() của lớp cha như ở hình trên thì phương thức CloudStore() trên lớp con sẽ được gọi mặc dù code trong hàm đang thao tác với đối tượng kiểu Smartphone.

Tính Đa hình như trên là một tính chất rất mạnh mẽ bởi vì nó mang lại cho code khả năng tổng quát hóa cao. Chúng ta không cần tạo ra phương thức cho mỗi kiểu kế thừa từ lớp cha Smartphone mà chỉ cần nhận một biến kiểu Smartphone và có thể làm việc với bất cứ lớp nào kế thừa từ nó. Điều duy

nhất không làm được ở đây là sử dụng những phương thức mà chỉ được khai báo trên các lớp con. VD: nếu ta có một phương thức trên lớp IPhone gọi là OpenSiri() nhưng không được khai báo trên lớp Smartphone, khi đó muốn gọi nó sẽ bắt buộc phải ép kiểu từ Smartphone sang IPhone trước khi gọi.

#### **3.4.2.4. Tính trừu tượng (Abstraction)**

Tính trừu tượng là một tiến trình ẩn các chi tiết trình triển khai và chỉ hiển thị tính năng tới người dùng. Tính trừu tượng cho phép chúng ta loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.

Tính trừu tượng giúp chúng ta tập trung vào những cốt lõi cần thiết của đối tượng thay vì quan tâm đến cách nó thực hiện.

Chúng là sử dụng các **abstract class** và **abstract interface** để có tính trừu tượng.

# CHƯƠNG 4: PHÂN TÍCH NỘI DUNG GAME

## Tổng quan về game

* **Giới thiệu game:**

Sky Angle là một game được thiết kế chạy trên nền tảng trang web.

Game hiển thị tốt nhất trên màn hình có độ phân giải từ 1024 x 768 trở lên (laptop, PC).

* **Mô tả game:**

**Sky Angle** là một trò chơi điều khiển máy bay trên bầu trơi xanh để tránh các con chim và chạm lấy những biểu tượng ngôi sao để có điểm và đồng thời cũng phải lấy biểu tượng dù để tăng nhiên liệu cho máy bay.

* **Hướng dẫn chơi:**

1. Di chuyển máy bay bằng cách rê chuột vào biểu tượng mũi tên trên màn hình.

2. Trong chuyến bay, máy bay cần:

+ Tránh những con chim bay trên bầu trời. Nếu máy bay bay trúng một con chim, trò chơi kết thúc.

+ thu thập dù để tăng nhiên liệu cho máy bay.

+ Thu thập các ngôi sao rơi trên bầu trời để tăng số điểm.

3. Bạn có thể:

+ Tạm dừng trò chơi bằng cách nhấp vào biểu tượng tạm dừng hoặc nhấn vào thanh dấu cách hoặc nhấn phím “p” trên bàn phím. Và nhấn lại các hình thức đó để tiếp tục trò chơi của bạn.

+ Tắt hoặc bật âm thanh trong trò chơi bằng cách nhấ vào biểu tượng âm thanh trên màn hình.

+ Tăng hoặc giảm kích thước chữ số hiển thị trong bằng cách nhấn vào biểu tượng chữ “A” trên màn hình.

4. Bắt đầu thử thách chuyến bay trong Sky Angle …….!

## Giao diện của game

**Giao diện mở đầu và hướng dẫn chơi game**

Hình 4.2.1 Giao diện mở đầu và hướng dẫn chơi game

**** **Giao diện khi chơi game**

*Hình 4.2.2 Giao diện khi chơi game*

**

*Hình 4.2.3 Giao diện khi chơi game*

**Các thành phần trong game**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Ghi chú** |
| 1 | Máy bay | Đối tượng được điểu khiển bởi người dùng. |  |
| 2 | Mây | Đối tượng di chuyển từ phải sang trái để tạo ấn tượng về chuyển động của máy bay trong bầu trời. |  |
| 3 | Chim | Đối tượng mà máy bay cầ phải tránh. |  |
| 4 | Sao | Đối tượng mà máy bay cần phải chạm lấy để tăng bộ đếm sao. |  |
| 5 | Dù | Đối tượng mà máy bay cần chạm lấy để tăng nhiên liệu. |  |
| 6 | Bộ đếm nhiên liệu | Đối tượng hiện lượng nhiên liệu có sẵn. |  |
| 7 | Bộ đếm sao | Đối tượng hiện số lượng sao mà máy bay lấy được. |  |
| 8 | Thời gian | Đối tượng cho thấy máy bay đã bay được bao lâu. |  |
| 9 | Nút âm thanh | Cho phép bật và tắt âm thanh trong game. |  |
| 10 | Nút cỡ chữ | Lên và xuống để tăng giảm cỡ chữ. |  |
| 11 | Nút dừng/tiếp tục game | Nút để tạm dừng hoặc tiếp tục game. |  |
| 12 | Logo | Hiện logo trong game. |  |

*Hình 4.2.4 Các thàn Thành phần trong game*

**Giao diện game over và lưu thông tin người chơi**

****

*Hình 4.2.5 Giao diện game over*

**Giao diện xem bảng xếp hạng của người chơi**



*Hình 4.2.6 Giao diện hiển thị bảng xếp hạng của người chơi*

# CHƯƠNG 5: KẾT LUẬN

Với việc xây dựng game Sky Angel. Nó đã giúp cho nhóm hiểu rõ hơn về ngôn ngữ lập trình HTML5, CSS3, ES6 cũng như hiểu rõ hơn về nhu cầu thực tế của việc giải trí qua game của người dùng, qua đó đưa ra được hướng phát triển cho game Sky Angel trong tương lai. Việc thực hiện đề tài cũng giúp các thành viên nâng cao khả năng tự học, tiếp thu, học hỏi lẫn nhau, và đặc biệt là sự quan tâm giúp đỡ tận tình cùng với những lời nhận xét sâu sắc của giáo viên hướng dẫn và các thấy cô trong Khoa sẽ tạo tiền đề cho nhóm rút ra được kinh nghiệm và hoàn thành tốt luận văn tốt nghiệp sắp tới.

Trong quá trình thực hiện đề tài, nhóm em đã thực hiện những việc sau:

* + - Nghiên cứu các công nghệ để phát triển và xây dựng giao diện người dùng là HTML5, CSS3, Javascript.
    - Thiết kế bố cục, hình ảnh và âm thanh cho game.
    - Phân tích và áp dụng hướng đối tượng vào game giúp cấu trúc và code trở nên chuyên nghiệp và dễ bảo trì hơn.
    - Sau khi hoàn thành đồ án thực tập tốt nghiệp này nhóm em đã có thêm nhiều kinh nghiệm trong việc phân tích thiết kế và phát triển một ứng dụng web cụ thể ở đây là game Sky angel và kinh nghiệm tối ưu hóa code để nâng cao tốc độ và hiệu suất cho ứng dụng.

# CHƯƠNG 6: HƯỚNG PHÁT TRIỂN

Qua quá trình nghiên cứu đề tài, nhóm em thấy được tầm quan trọng của việc phân tích ứng dụng và áp dụng được các kỹ thuật, công nghệ đang là xu hướng hiện nay vào dự án. Tuy nhiên, đây mới chỉ là một vài chức năng cơ bản của ứng dụng.

Trong tương lai, nhóm em có nhiều thời gian hơn và sẽ củng cố kiến thức, nắm rõ hơn các kỹ thuật, và nhóm em sẽ:

* Phát triển thêm nhiều level.
* Tăng đồ họa và tương tác giao diện.
* Tăng thêm hiệu ứng trong game.
* Tối ưu hóa game, tăng độ mượt mà.

# CHƯƠNG 7: TÀI LIỆU THAM KHẢO

1. Website: <https://hocwebchuan.com/>
2. Website: <https://www.w3schools.com/html/>
3. Website: <https://www.w3schools.com/css/>
4. Website: <https://www.w3schools.com/js/>
5. Website: <https://viblo.asia/p/tim-hieu-ve-lap-trinh-huong-doi-tuong-object-oriented-programming-bxjvZwlBGJZ>
6. Website: <https://nhungdongcodevui.com/2016/03/27/bai-2-1-html5/>