

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC BÁCH KHOA KHOA KHOA HỌC ỨNG DỤNG

----------

BÁO CÁO

THỰC TẬP TỐT NGHIỆP

CÔNG TY AKSELOS

GVHD: Thầy Nguyễn Thái Hiền

SVTH : Lê Nguyễn Hoài Linh MSSV:1913942

*TP.Hồ Chí Minh, tháng 08 năm 2022*

**BẢN ĐÁNH GIÁ THỰC TẬP TỐT NGHIỆP**

*Nhằm đánh giá kết quả SV thực tập tốt nghiệp, đề nghị Quý Vị phụ trách hướng dẫn cho ý kiến đánh giá về quá trình thực tập của sinh viên:*

*. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .*

*theo mẫu sau:*

|  |  |
| --- | --- |
| **ĐÁNH GIÁ THỰC TẬP** | |
| Nội dung công việc thực tập |  |
| Ý thức tổ chức kỷ luật  *(Tốt, Khá, Trung bình, Kém)* |  |
| Tính tích cực năng động học hỏi *(Tốt, Khá, Trung bình, Kém)* |  |
| Đánh giá chung bằng điểm  *(cho điểm cụ thể theo thang điểm 10)* |  |
| Quý Vị có thể tiếp tục nhận SV thực tập cho những năm tới không?  Số lượng SV? |  |
| **KHẢ NĂNG HƯỚNG DẪN LUẬN VĂN TỐT NGHIỆP** | |
| Khả năng phát triển thành luận văn tốt nghiệp *(Có – Không)* |  |
| Tên hoặc hướng đề tài TN dự kiến |  |
| Nếu được, Quý vị có thể đảm nhiệm hướng dẫn không? *(hướng dẫn chính - đồng hướng dẫn)* |  |
| Ý kiến khác: | |

|  |  |
| --- | --- |
| *Người hướng dẫn:*  *Họ và tên : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .*  *Bệnh viện, cơ quan: . . . . . . . . . . . . . . . . . . . . . . . . . . . . .* | *Chữ ký,* |
| *. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .* |  |

*Số ĐT hoặc Mobile: . . . . . . . . . . . . . . . . . . . . . . . . . . . . .*

###### NHẬN XÉT

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

###### LỜI NÓI ĐẦU

Với sự biết ơn sâu sắc, lời đầu tiên em xin chân thành gửi lời cảm ơn đến quý thầy cô khoa Khoa Học Ứng Dụng trường đại học Bách Khoa thành phố Hồ Chí Minh đã truyền dạy cho chúng em vốn kiến thức và kinh nghiệm quý báu để là nền tảng vững chắc để xây dụng tương lai phía trước, đặc biệt xin cảm ơn thầy Nguyễn Thái Hiền bộ môn Cơ Kỹ Thuật đã giúp đỡ em có một kỳ thực tập tại công ty AKSELOS.

Em xin chân thành gửi lời cảm ơn đến Ban lãnh đạo công ty AKSELOS đã cho phép em thực hiện khóa Thực Tập Tốt Nghiệp, tạo điều kiện và hỗ trợ sinh hoạt trong suốt thời gian thực tập tại công ty. Cảm ơn anh Tiến, anh Hoàng và toàn và các anh chị trong công ty đã hướng dẫn giúp đỡ em trong suốt thời gian thực tập tại công ty.

Thực tập tốt nghiệp là cơ hội để em có thể hiểu tìm hiểu và tiếp xúc với công việc thực tế trong tương lai. Trong thời gian thực tập, ngoài sự nỗ lực cố gắng của bản thân, em còn nhận được sự quan tâm, hướng dẫn rất tận tình từ anh chị tạo mọi điều kiện để em có thể tiếp xúc với thực tế, áp dụng những kiến thức đã học vào thực tế, cũng như học hỏi thêm được nhiều kinh nghiệm, đồng thời giúp em có thể hoàn thành tốt bài báo cáo thực tập của mình.

Vì kiến thức bản thân còn hạn chế trong quá trình thực tập em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp để em hoàn thiện hơn.

Sinh viên thực hiện

Lê Nguyễn Hoài Linh

Mục Lục

[Chương 1: VỀ CÔNG TY AKSELOS 6](#_Toc110375074)

[1.1 Giới thiệu 6](#_Toc110375075)

[1.2 Lịch sử 6](#_Toc110375076)

[1.3 Giải pháp 6](#_Toc110375077)

[1.3.1 Thiết kế 6](#_Toc110375078)

[1.3.2 Vận hành 7](#_Toc110375079)

[1.3.3 Kéo dài vòng đời kết cấu 8](#_Toc110375080)

[1.4 Công nghệ 8](#_Toc110375081)

[1.4.1 Quy mô 8](#_Toc110375082)

[1.4.2 Tốc độ 8](#_Toc110375083)

[1.4.3 Độ chính xác 8](#_Toc110375084)

[1.5 Lĩnh vực và đối tác 8](#_Toc110375085)

[Chương 2: TÌM HIỂU NGÔN NGỮ SỬ DỤNG ĐỂ LẬP TRÌNH WEB 10](#_Toc110375086)

[2.1 Tìm hiểu về HTML. 10](#_Toc110375087)

[2.1.1 HTML là gì ? 10](#_Toc110375088)

[2.1.2 Lịch sử của ngôn ngữ. 10](#_Toc110375089)

[2.1.3 HTML hoạt động như thế nào ? 10](#_Toc110375090)

[2.1.4 HTML dùng để làm gì? 11](#_Toc110375091)

[2.1.5 Cấu trúc HTML 11](#_Toc110375092)

[2.1.6 Ưu điểm của HTML 12](#_Toc110375093)

[2.1.7 Nhược điểm của HTML 12](#_Toc110375094)

[2.1.8 Một số thuật ngữ HTML thường gặp 13](#_Toc110375095)

[2.2 Tìm hiểu về CSS 14](#_Toc110375096)

[2.2.1 CSS là gì ? 14](#_Toc110375097)

[2.2.2 Ưu điểm 15](#_Toc110375098)

[2.2.3 Nhược điểm 15](#_Toc110375099)

[2.2.4 Cách thức hoạt động của CSS 16](#_Toc110375100)

[2.2.5 Internal, External và Inline CSS Styles 17](#_Toc110375101)

[2.3 Tìm hiểu về JavaScript 18](#_Toc110375102)

[2.3.1 JavaScript là gì ? 18](#_Toc110375103)

[2.3.2 Lịch sử phát triển của Javascript 18](#_Toc110375104)

[2.3.3 Javascript hoạt động trên trang web như thế nào? 19](#_Toc110375105)

[2.3.4 Javascript dùng để làm gì? 19](#_Toc110375106)

[2.3.5 Ưu điểm 20](#_Toc110375107)

[2.3.6 Nhược điểm 21](#_Toc110375108)

[2.3.7 Cách viết một chương trình JS 22](#_Toc110375109)

[2.3.8 Điểm khác biệt JavaScript với ngôn ngữ lập trình khác 23](#_Toc110375110)

[Chương 3: GIẢI QUYẾT NHỮNG VẤN ĐỀ CƠ BẢN 25](#_Toc110375111)

[3.1 Thiết kế giao diện webside cơ bản với HTML 25](#_Toc110375112)

[3.1.1 Đề bài 25](#_Toc110375113)

[3.1.2 Các phần tử sẽ sử dụng 26](#_Toc110375114)

[3.1.3 Kết quả 27](#_Toc110375115)

[3.2 Thiết kế giao diện webside cơ bản với HTML kết hợp CSS 28](#_Toc110375116)

[3.2.1 Đề bài 28](#_Toc110375117)

[3.2.2 Các thẻ và thuộc tính sử dụng 29](#_Toc110375118)

[3.2.3 Kết quả 30](#_Toc110375119)

[3.3 Thiết kế ứng dụng quản lý học sinh trên web 31](#_Toc110375120)

[3.3.1 Đề bài 31](#_Toc110375121)

[3.3.2 Các lệnh sẽ dùng 31](#_Toc110375122)

[3.3.3 Kết quả 32](#_Toc110375123)

[Chương 4: PHÁT TRIỂN ỨNG DỤNG 33](#_Toc110375124)

[4.1 Tạo ứng dụng có thể tương tác đồ họa với người dùng 33](#_Toc110375125)

[4.2 Tạo những tính năng vẽ cơ bản cho ứng dụng 34](#_Toc110375126)

[4.2.1 Line 35](#_Toc110375127)

[4.2.2 Circle 35](#_Toc110375128)

[4.2.3 Spline 36](#_Toc110375129)

[4.3 Lưu những dữ liệu thành đối tượng 38](#_Toc110375130)

[4.3.1 Point 39](#_Toc110375131)

[4.3.2 Line 39](#_Toc110375132)

[4.3.3 Area 39](#_Toc110375133)

[4.4 Tạo tính năng phát hiện diện tích 39](#_Toc110375134)

[4.5 Tạo tính năng tương tác với đối tượng 43](#_Toc110375135)

[4.5.1 Click chọn 44](#_Toc110375136)

[4.5.2 Quét chọn 44](#_Toc110375137)

[4.6 Import & export dữ liệu 45](#_Toc110375138)

DANH MỤC HÌNH

[Hình 1‑1 Vòng lặp hồi tiếp số 7](#_Toc110375139)

[Hình 2‑1 Minh họa HTML 10](#_Toc110375140)

[Hình 2‑2 Minh họa CSS 14](#_Toc110375141)

[Hình 2‑3 Cách CSS hoạt động 16](#_Toc110375142)

[Hình 2‑4 Minh họa Javascript 18](#_Toc110375143)

[Hình 2‑5 Minh họa ưu điểm JS 20](#_Toc110375144)

[Hình 2‑6 Minh họa nhược điểm của JS 21](#_Toc110375145)

[Hình 2‑7 Chương trình HTML cơ bản 22](#_Toc110375146)

[Hình 2‑8 JS trong HTML 23](#_Toc110375147)

[Hình 3‑1 Đề bài HTML 25](#_Toc110375148)

[Hình 3‑2 Kết quả web HTML 27](#_Toc110375149)

[Hình 3‑3 Đề bài CSS 28](#_Toc110375150)

[Hình 3‑4 Kết quả bài CSS 30](#_Toc110375151)

[Hình 3‑5 Đề bài JS 31](#_Toc110375152)

[Hình 3‑6 Kết quả bài JS 32](#_Toc110375153)

[Hình 4‑1 Quy trình xử lý FE problem 33](#_Toc110375154)

[Hình 4‑2 Vùng đồ họa Canvas 34](#_Toc110375155)

[Hình 4‑3 Vẽ đường thẳng trên canvas 35](#_Toc110375156)

[Hình 4‑4 Vẽ hình tròn trên canvas 35](#_Toc110375157)

[Hình 4‑5 Minh họa spline bậc 3 37](#_Toc110375158)

[Hình 4‑6 Spline tạo từ những đoạn thẳng nhỏ 38](#_Toc110375159)

[Hình 4‑7 Cấu trúc dữ liệu 38](#_Toc110375160)

[Hình 4‑8 Minh họa phát hiện diện tích 40](#_Toc110375161)

[Hình 4‑9 Các loại đồ thị phẳng 41](#_Toc110375162)

[Hình 4‑10 Planar grahp 42](#_Toc110375163)

[Hình 4‑11 Định hướng trái sang phải 43](#_Toc110375164)

[Hình 4‑12 Xác định diện tích 43](#_Toc110375165)

[Hình 4‑13 quá trình kiểm tra đối tượng 44](#_Toc110375166)

[Hình 4‑14 cấu trúc file export 45](#_Toc110375167)

# VỀ CÔNG TY AKSELOS

## Giới thiệu

Akselos là một công ty Thụy Sĩ cung cấp nền tảng và giải pháp mô phỏng kỹ thuật dựa trên phương pháp phần tử hữu hạn giảm thiểu (Reduced-basis finite element analysis). Nền tảng này được sử dụng để tạo ra bản sao số của kết cấu nhằm cải thiện việc thiết kế, bảo trì, độ tin cậy và vòng đời.

## Lịch sử

* Năm 2011, công nghệ từ dự án “Mô phỏng độ phân giải cao cho phân tích hệ thống” từ Viện Công nghệ Massachusetts (MIT) được triển khai cho Akselos.
* Năm 2012, công ty Akselos được thành lập bởi David Knezevic, Thomas Laurent và Huỳnh Đình Bảo Phương, những người đã tham gia vào quá trình nghiên cứu ban đầu.
* Akselos đã gây 2,2 triệu đô vòng đầu tư đầu tiên vào năm 2016. Ở vòng đầu tư thứ hai, 10 triệu đô từ Innogy Ventures và Shell Ventures đã được gây vốn thành công.
* Năm 2020, Akselos được chọn là một tiên phong công nghệ ở Diễn đàn Kinh tế Thế giới

## Giải pháp

### Thiết kế

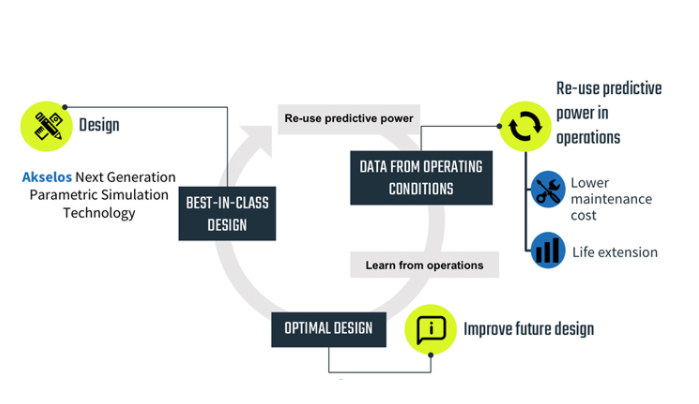
Thiết kế tối ưu là điều cốt yếu cho những kết cấu ngoài khơi phức tạp. Akselos tạo điều kiện cho các kỹ sư và nhà thiết kế giảm số mẫu thử vật lý cần thiết để có một thiết kế hiệu quả và tốt hơn.

#### Giảm chi phí tài sản cố định

Phần mềm của Akselos cho độ chính xác cao, giúp khách hàng tránh việc thiết kế dư thừa, tạo ra những kết quả và thiết kế mang tính cách mạng. Akselos cung cấp mô hình giảm bậc có tốc độ xử lý nhanh hơn 1000 lần hơn các phần phềm phân tích truyền thống, cho phép nhiều lựa chọn thiết kế được mô phỏng hiệu quả. Việc mô hình số nhanh giúp nhiều cấu hình được phân tích trong vài phút thay vì vài tháng.

#### Vòng lặp hồi tiếp số

Vòng lặp Hồi tiếp Số (Digital Feedback Loop) của Akelos cho phép dữ liệu vận hành thực tế được thu thập từ bản sao số có tính dự đoán của Akselos được sử dụng để thẩm định tính toàn vẹn kết cấu và làm quá trình thiết kế hiệu quả hơn. Các tiếp cận này tạo ra những thiết kế tốt hơn và bền vững hơn. Nghiên cứu của Akselos cho thấy thiết kế được tối ưu hóa bởi Vòng lặp Hồi tiếp số có thể tiết kiệm 30% chi phí tài sản cố định cho vật liệu và 10% cho thiết kế.



Hình ‑ Vòng lặp hồi tiếp số

### Vận hành

Digital Guardian của Akselos đang đem lại một cuộc cách mạng cho việc Quản lý Tính Toàn vẹn của Tài sản bằng cách giúp những người vận hành cải tiến chiến thuật Kiểm tra dựa trên Rủi ro với Giám sát tình trạng thời gian thực.

#### Tối ưu quá trình vận hành và giảm chi phí vận hành

Bằng việc kết hợp giám sát tình trạng với chiến thuật kiểm tra tập trung hơn để giảm chi phí kiểm tra, khách hàng của akselos đã thành công trong việc giảm chi phí vận hành bằng cách giảm chi phí vận hành. Điều này đạt được bằng cách tăng cường kiểm tra dựa trên rủi ro với các mô phỏng vật lý chính xác.

#### Thay thế các ước tính dư thừa với độ chính xác

Công nghệ mô phỏng truyền thống không đủ mạnh mẽ cho việc giám sát tình trạng toàn bộ kết cấu toàn thời gian. Điều này nghĩa là thông tin về tình trạng kết cấu sẽ luôn thiếu sót. Kết quả là một góc nhìn sai lệch về rủi ro và các ước tính dư thừa. Digital Guardian của Akselos là một mô hình vật lý có cảm biến phản ánh tình trạng toàn bộ kết cấu trong thời gian thực. Nó nhanh hơn 1000 lần so với phần mềm truyền thống, cho phép phân tích đầy đủ kết cấu với quy mô không giới hạn trong vài phút

#### Hệ thống Hỗ trợ Quyết định

Nền tảng đám mây của Akselos cho phép truy cập ngay lập tức vào Hệ thống Hỗ trợ Quyết định (Decision Support System – DSS) cho những cái nhìn chính xác vào tình trạng kết cấu. Nó làm nổi bật những khu vực đáng lo ngại và giúp việc quyết định kế hoạch kiểm tra và bảo dưỡng hiệu quả hơn.

### Kéo dài vòng đời kết cấu

Digital Guardian hỗ trợ kéo dài vòng đời bằng cách giúp những người vận hành hiểu rõ về kết cấu và loại bỏ sự bảo thủ không mong muốn, cung cấp đánh giá chính xác về vòng đời kết cấu, trong khi vẫn đảm bảo các tiêu chuẩn của ngành.

## Công nghệ

### Quy mô

Công nghệ RB-FEA của Akselos là cách tiếp cận mô hình giảm bậc được phát triển tại MIT. Nó cho phép mô hình toàn diện và chi tiết những kết cấu quan trọng. RB-FEA cho phép những mô hình lớn hơn nhiều so với FEA; 100 triệu bậc tự do hoặc nhiều hơn có thể được xử lý hiệu quả. Mô hình toàn diện loại bỏ nhu cầu mô hình đa tầng điển hình đối với phương pháp truyền thống.

### Tốc độ

RB-FEA nhanh hơn 1000 lần FEA truyền thống. Nó cho phép nhiều kịch bản được xem xét vì 1000 phiên bản khác nhau của mô hình có thể chạy song song.

### Độ chính xác

Mô hình của Akselos cho phép dữ liệu tình trạng (bao gồm vết nứt, ăn mòn, hư hại được tích hợp vào bản sao số của toàn bộ cấu trúc.

## Lĩnh vực và đối tác

Hiện tại, Akselos hoạt động chính nhưng không giới hạn ở các ngành:

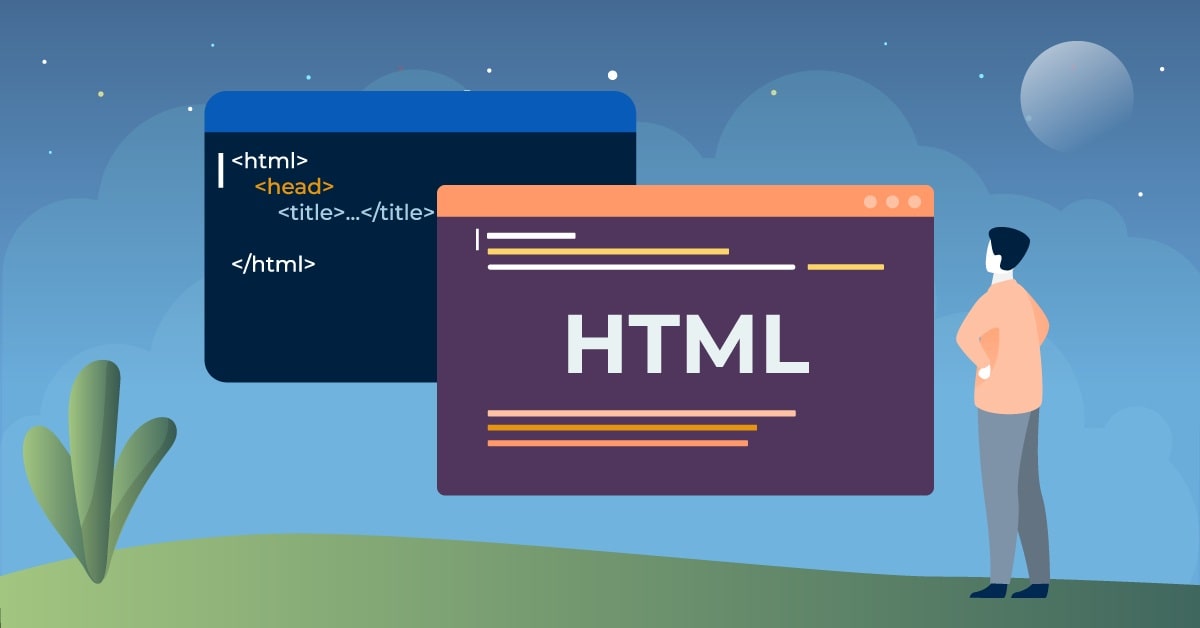
* Năng lượng gió (các kết cấu xa bờ và trên bờ)
* Dầu khí ngoài khơi
* Dầu khí hạ nguồn
* Hàng không và quốc phòng
* Đóng tàu

Các đối tác của Akselos bao gồm:

* Lamprell
* Worley
* TechnoStar
* Cosmi
* CCS
* Enerbull Limited
* Logaritm

# TÌM HIỂU NGÔN NGỮ SỬ DỤNG ĐỂ LẬP TRÌNH WEB

## Tìm hiểu về HTML.



Hình ‑ Minh họa HTML

### HTML là gì ?

HTML là từ viết tắt của Hypertext Markup Language, là sự kết hợp của Hypertext và Markup, hay còn được gọi là ngôn ngữ siêu văn bản. HTML có chức năng giúp người dùng xây dựng và cấu trúc các phần trong trang web hoặc ứng dụng, thường được sử dụng trong phân chia các đoạn văn, heading, link, blockquotes,…

### Lịch sử của ngôn ngữ.

* Cha đẻ của HTML là Tim Berners – Lee – nhà vật lý học, là người nghĩ ra ý tưởng dựa trên hệ thống hypertext trên nền internet.
* Năm 1991, xuất bản phiên bản đầu tiên của HTML gồm 18 tag HTML.
* Năm 1998, HTML phiên bản 4.01 ra đời.
* Năm 2000, các phiên bản HTML được thay thế bằng XHTML.
* Năm 2000, các phiên bản HTML được thay thế bằng XHTML.

### HTML hoạt động như thế nào ?

Dấu hiệu nhận biết HTML documents chính là files có kết thúc đuôi là .html hoặc htm và có thể xem bằng cách sử dụng bất kỳ trình duyệt web nào, chẳng hạn như Safari, Google chrome, Microsoft edge,… Các trình duyệt sẽ đọc được files HTML và xuất bản nội dung lên internet, tại đây người dùng hoàn toàn dễ dàng đọc được nó.

Thực tế, một web sẽ chứa nhiều trang web HTML, có thể kể đến như trang chủ, trang about, trang liên hệ,… tất cả các trang đều có HTML riêng. Cụ thể, mỗi trang HTML chứa một bộ các tag, hay còn được gọi là elements, được hiểu là các yếu tố để xây dựng từng khối của một trang web.

Các HTML elements tạo thành cấu trúc cây thư mục là section, paragraph, heading và những khối nội dung khác. Các HTML elements đều có tag mở và tag đóng, có cấu trúc <tag></tag>.

### HTML dùng để làm gì?

Hiểu được khái niệm ngôn ngữ html là gì và hình thức hoạt động của HTML để trả lời cho câu hỏi HTML dùng để làm gì? Cụ thể HTML là một loại ngôn ngữ đánh dấu siêu văn bản, với mục đích cấu trúc thành các cấu trúc cơ bản của một trang web để website trở thành một hệ thống hoàn chỉnh.

HTML dùng ngôn ngữ của mình để đánh dấu siêu văn bản, điều này sẽ giúp các văn bản trên trên website được chia bố cục rõ ràng, chia khung sườn các thành phần của trang web, tạo trang web thành một hệ thống hoàn chỉnh.

### Cấu trúc HTML

HTML có nhiều dạng thẻ khác nhau, mỗi thẻ sẽ có những nhiệm vụ khác nhau và ý nghĩa nhất định, có ảnh hưởng và tác động đến nhau

* Phần khai báo chuẩn của html, xhtml. Có cấu trúc là <!Doctype>.  
  Phần này cho người dùng biết được trình duyệt đang sử dụng hiện đang dùng phiên bản HTML nào. Trên trang web hiện đang rất nhiều loại HTML khác nhau và mỗi trình duyệt chỉ một loại HTML nhất định.
* Phần tiêu đề: Phần khai báo ban đầu, khai báo về về meta, little, javascript, css,…  
  Phần này có cấu trúc bắt đầu bằng thẻ <head> và kết thúc bởi thẻ <head>. Đây là phần chứa tiêu đề và tiêu đề được hiển thị trên thanh điều hướng của trang web. Cụ thể, tiêu đề là phần nội dung nằm giữa cặp thẻ <title> và </title>. Bên cạnh đó phần tiêu đề còn chứa các khai báo có thông tin nhằm phục vụ SEO.
* Phần thân: Phần chứa nội dung của trang web, là nơi hiển thị nội dung của trang web.  
  Phần này nằm phía sau tiêu đề, bao gồm các thông tin mà bạn muốn hiển thị trên trang web bao gồm văn bản, hình ảnh và các liên kết. Phần thân bắt đầu bằng thẻ <body> và kết thúc bằng thẻ </body>.
* Thẻ cặp nằm ngoài cùng, cả phần tiêu đề và phần thân đều phải nằm trong cặp thẻ <html> và </html>. Đây là cặp thẻ nằm ngoài cùng, cặp thẻ này có nhiệm vụ bao hết nội dung của trang web lại.

### Ưu điểm của HTML

* HTML được ra đời từ rất lâu, do đó HTML có nguồn tài nguyên khổng lồ, hỗ trợ một cộng đồng người dùng lớn. Bên cạnh đó, cộng đồng HTML ngày càng phát triển trên thế giới.
* Mã nguồn của HTML là mã nguồn mở, do đó người dùng có thể sử dụng hoàn toàn miễn phí.
* HTML được sử dụng và được sử dụng trên nhiều trình duyệt được nhiều người dùng ưa chuộng hiện nay như Internet Explorer, Chrome, FireFox, Cốc cốc,…
* Học và tìm hiểu HTML đơn giản nên người học dễ dàng nắm được kiến thức và vận dụng trong xây dựng trang web căn bạn.
* HTML được quy định theo một tiêu chuẩn nhất định nên việc markup sẽ trở nên gọn gàng, đồng nhất bởi HTML được vận hành bởi World Wide Web Consortium (W3C).
* HTML được thực hiện dễ dàng bởi HTML được tích hợp nhiều ngôn ngữ khác nhau như PHP, Java, NodeJs, Ruby,…Điều này sẽ giúp tạo thành một website hoàn chỉnh với nhiều tính năng.

### Nhược điểm của HTML

* Nhược điểm lớn nhất của HTML đó chính là chỉ có thể web tĩnh, web tĩnh có thể hiểu là những trang web chỉ hiện thông tin mà không có sự tương tác cho người dùng. Do đó, khi xây dựng tính năng động hoặc xây dựng hệ thống website có sự tương tác với người dùng, lập trình viên cần phải dùng thêm JavaScript hoặc ngôn ngữ backend của bên thứ ba.
* HTML thường chỉ có thể thực thi những thứ logic và cấu trúc nhất định, HTML không có khả năng tạo sự khác biệt và mới mẻ.
* Một số trình duyệt vẫn còn chậm trong viết hỗ trợ các phiên bản mới của HTML, đặc biệt là HTML5.
* Một số trình duyệt không thể render những tag mới trong HTML5.

### Một số thuật ngữ HTML thường gặp

Trong HTML có rất nhiều thuật ngữ khác nhau cần người dùng hiểu để có thể sử dụng một các dễ dàng và thuận lợi hơn. Sau đây, chúng ta sẽ tìm hiểu 3 thuật ngữ phổ biến thường xuyên xuất hiện trong tệp HTML, chính là Elements, Tags, Attributes.

* Elements  
  Elements được hiểu là các chỉ định dùng để xác định nội dung, đối tượng trong website. Trong tệp HTML, Elements được bao quanh và xác định bằng dấu ngoặc <>. Cụ thể, những yếu tố thường được sử dụng phổ biến như đoạn văn <p>, các cấp độ tiêu đề từ <h1> đến <h6>, danh sách tiếp tục bao gồm <a>, <span>, <strong> hay <em>,…
* Tags  
  Tags được biết đến là một element được bao quanh bởi các dấu ngoặc < >, điều này sẽ tạo ra các thẻ. Chẳng hạn thẻ mở là dấu hiệu nhận biết cho sự bắt đầu của một Element (ví dụ: <div>). Thẻ đóng sẽ được đánh dấu vào cuối của một Element, thẻ đóng có hình thức là dấu ngoặc nhỏ + dấu chéo + dấu ngoặc lớn (ví dụ: </div>). Ở giữa thẻ mở và thẻ đóng chính là nội dung của Element.
* Attributes  
  Attributes được biết đến là thuộc tính được dùng để cung cấp các thông bổ sung cho một Element. Các Attributes bao gồm tên và giá trị, được xác định sau tên của một thành phần và xuất hiện trong thẻ mở. Định dạng của Attributes như sau: tên thuộc tính + dấu bằng + giá trị thuộc tính được trích dẫn.

## Tìm hiểu về CSS



Hình ‑ Minh họa CSS

### CSS là gì ?

CSS – Cascading Style Sheet là một ngôn ngữ lập trình được thiết kế vô cùng đơn giản, dễ sử dụng. Với CSS hướng tới mục tiêu chính là giúp đơn giản hóa được quá trình tạo ra các website.

Nhiệm vụ chính của CSS là thực hiện việc xử lý giao diện của một trang web cụ thể. Đó là những yếu tố như màu sắc văn bản, hay khoảng cách giữa các đoạn, hoặc kiểu font chữ, hình ảnh, bố cục, màu nền,… đều có thể thay đổi, chỉnh sửa theo ý muốn với hỗ trợ của CSS.

Sử dụng CSS chính là một công cụ hỗ trợ hữu ích và cần thiết cho quá trình phát triển, thiết kế website. Những công việc mà HTML không xử lý được sẽ được giải quyết tốt với CSS. Nhờ có CSS giúp mang lại khả năng kiểm soát mạnh mẽ, hiệu quả với quá trình trình bày một tài liệu HTML cụ thể. Bởi thế mà nó là công cụ thường được kết hợp với các ngôn ngữ như HTML hay XHTML.

Đối với CSS hiện nay có nhiều kiểu khác nhau được đưa vào sử dụng. Song xét một cách cơ bản nhất thì nó được phân chia thành các loại thông dụng là:

* CSS tùy chỉnh hình nền – Background
* CSS tùy chỉnh cách hiển thị đoạn text – Text
* CSS tùy chỉnh kiểu chữ và kích thước – Font
* CSS tùy chỉnh bảng – Table
* CSS tùy chỉnh danh sách – Link
* Mô hình box model có kết hợp với padding, margin, border – Box model

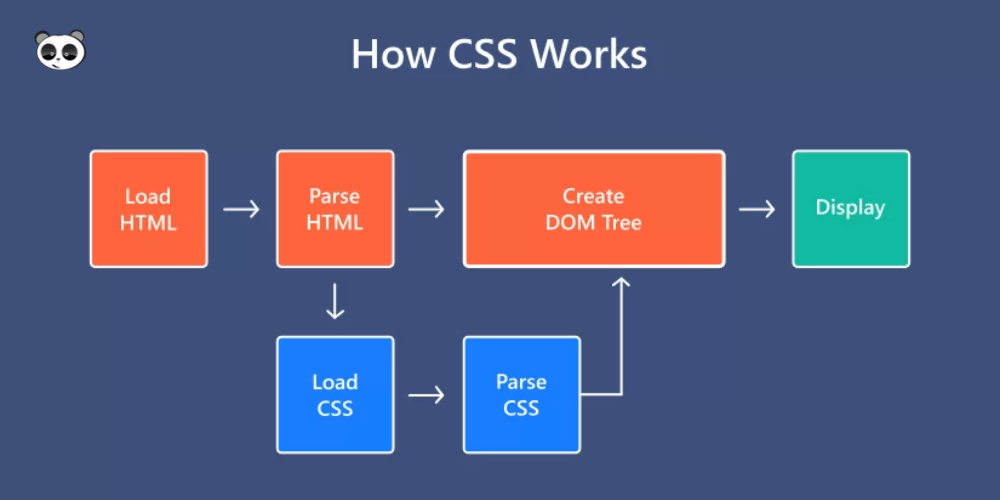
### Ưu điểm

* Khả năng tiết kiệm thời gian  
  CSS khi sử dụng có thể viết lần đầu, đồng thời sử dụng lại trong các trang HTML tiếp theo. KHông chỉ vậy, chúng ta hoàn toàn có thể xác định được một kiểu cho từng thành phần HTML và cũng áp dụng được cho nhiều trang web khác khi cần thiết.
* Khả năng tải trang nhanh chóng  
  Đặc trưng của CSS khi sử dụng là ít mã hơn. Bởi thế mà thời gian tải xuống nhanh chóng, giúp tiết kiệm thời gian đáng kể. Khi viết được một quy tắc CSS của một thẻ cụ thể và lúc này chúng ta có thể áp dụng nó cho mọi lần xuất hiện tiếp theo của thẻ đó, đồng thời hoàn toàn không cần tiến hành thay đổi thuộc tính của thẻ HTML.
* Dễ dàng khi thực hiện bảo trì  
  Khi cần chúng ta chỉ cần thay đổi một kiểu và mọi thành phần trong tất cả những website thì lúc này CSS sẽ hỗ trợ cập nhật hoàn toàn tự động.
* Sở hữu thuộc tính rộng  
  CSS được đánh giá cao nhờ sở hữu những thuộc tính rộng hơn nếu đánh giá và so sánh với HTML.
* Khả năng tương thích tốt  
  Khả năng tương thích với nhiều thiết bị cũng là ưu điểm của CSS. Cùng sử dụng một tài liệu HTML song với nhiều phiên bản website khác nhau được trình bày ở từng thiết bị di động đều được hỗ trợ tốt, có độ tương thích cao.

### Nhược điểm

* CSS hoạt động khác biệt cho từng trình duyệt  
  Với những thay đổi ban đầu của CSS trên một website rất dễ dàng. Tuy nhiên, khi thay đổi đã được thực hiện đòi hỏi chúng ta phải xác nhận được tính tương thích khi CSS hiển thị hiệu ứng thay đổi tương tự cho từng trình duyệt. Điều này xuất hiện do CSS sẽ hoạt động khác biệt cho từng trình duyệt cụ thể.
* Khá khó khăn cho người mới  
  Ngôn ngữ lập trình phát triển đa dạng và vô cùng phúc tạp, đặc biệt là khó khăn với những người mới bắt đầu. Bởi thế, với nhiều cấp độ của CSS thì việc tìm hiểu, có thể nắm bắt để sử dụng càng trở nên khó khăn hơn.
* Định dạng của web có khả năng gặp rủi ro  
  CSS là hệ thống dựa trên văn bản mở nên việc truy cập khá dễ dàng. Điều này khiến định dạng toàn bộ của web hoàn toàn có thể chịu tổn thương, gặp gián đoạn khi có hành động, hoặc tai nạn nào xảy ra với tệp. Lúc này nó sẽ yêu cầu truy cập đọc hoặc ghi vào web dự định để có thể ghi đè lên được các thay đổi.

### Cách thức hoạt động của CSS



Hình ‑ Cách CSS hoạt động

Nhờ CSS giúp mang lại phong cách riêng biệt cho từng website thông qua việc tương tác với những yếu tố của HTML. Từ những phần từ là các thành phần HTML riêng lẻ của một website cụ thể.

Ví dụ chúng ta muốn một đoạn cụ thể trên website xuất hiện với màu xanh và đậm để người xem trang dễ dàng nhìn thấy nó một cách ấn tượng chúng ta cần nhập mã CSS là:

p{màu: xanh; font-weight: in đậm;}

Ở ví dụ này thì <p> chính là đoạn văn được coi là bộ chọn của một bộ điều chỉnh trong công cụ CSS. Bên cạnh đó thì một bộ chọn sẽ được ghi ở bên trái của dấu ngoặc nhọn { đầu tiên. Đối với những thông tin nằm ở giữa các dấu ngoặc nhọn sẽ là khai báo và trong đó có chứa đầy đủ các thuộc tính, các giá trị được áp dụng cho một bộ chọn.

Ngoài ra, chúng ta có nhiều cách được áp dụng để tiến hành nhúng CSS vào website. Cụ thể có thể cân nhắc các giải pháp là:

* Cách 1: tiến hành chèn vào trong thẻ <style></style>
* Cách 2: thực hiện chèn thuộc tính style của phần tử HTML.
* Cách 3: thực hiện bằng cách chèn file.css thông qua thẻ <link> vào trong thẻ <head> với tài liệu HTML.

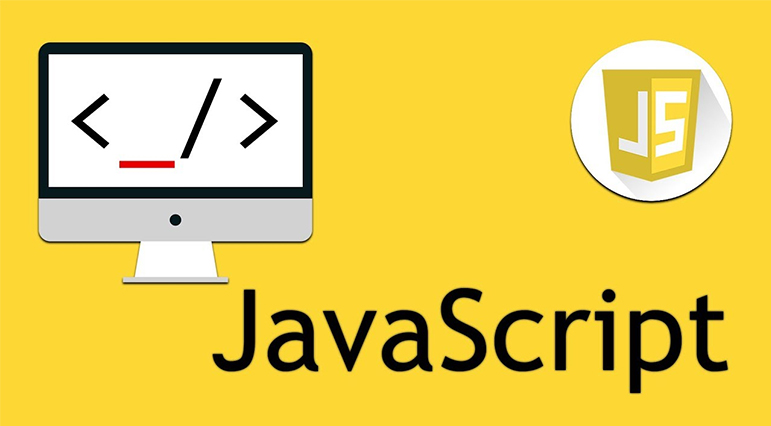
### Internal, External và Inline CSS Styles

Style CSS Internal là style được tải lên mỗi khi trang web được refresh. Vì vậy nó tăng thời gian tải trang. Ngoài ra, bạn sẽ không dùng một style CSS cho nhiều trang vì nó chỉ áp dụng cho từng trang một. Tuy nhiên, lợi ích của style CSS Internal là khi mọi thứ đã đặt trong một trang thì nó dễ chia sẽ trang để xem trước hơn.

Phương pháp dùng Style External là thuận tiên nhất. Mọi thứ được lưu trong file .css. Có nghĩa là bạn có thể tạo phong cách ở file khác áp dụng CSS vào trang bạn muốn. External style sẽ cải thiện thời gian tải trang rất nhiều.

Cuối cùng, về style CSS Inline. Inline hoạt động với một yếu tố nhất định có tag <style>. Mỗi thành phần đều cần được tạo phong cách riêng, vì vậy đây không hẵn là cách tốt nhất và dễ nhất để xử lý CSS. Nhưng có thể khá tiện lợi, vì nếu bạn muốn thay đổi chỉ một yếu tố, nhanh chóng xem trước thay đổi, bạn không cần truy cập trực tiếp vào file CSS để chỉnh sửa mà sử dụng Inline CSS.

## Tìm hiểu về JavaScript



Hình ‑ Minh họa Javascript

### JavaScript là gì ?

Javascript chính là một ngôn ngữ lập trình web rất phổ biến ngày nay. Javascript được tích hợp đồng thời nhúng vào HTML để hỗ trợ cho website trở nên sống động hơn. Chúng cũng đóng vai trò tương tự như một phần của website, cho phép Client-side Script từ người dùng tương tự máy chủ (Nodejs) để tạo ra những website động.

### Lịch sử phát triển của Javascript

* Brendan Eich chính là người đã phát triển Javascript tại Netscape với tiền thân là Mocha. Sau đó, Mocha được đổi thành LiveScript và cuối cùng mới đổi thành JavaScript.
* Năm 1998, JavaScript với phiên bản mới nhất là ECMAScript 2 phát hành và đến năm 1999 thì ECMAScript 3 được ra mắt.
* Năm 2016, ứng dụng JavaScript đã đạt kỷ lục lên tới 92% website sử dụng, đồng thời cũng được đánh giá là một công cụ cực kỳ quan trọng đối với lập trình viên.

### Javascript hoạt động trên trang web như thế nào?

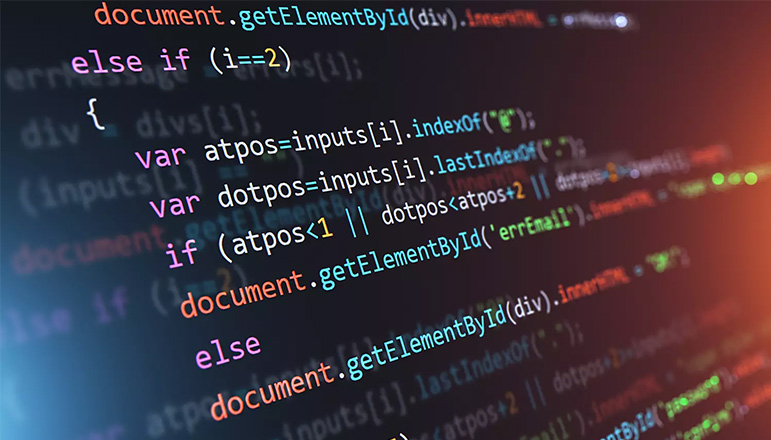
* Thông thường, JavaScript sẽ được nhúng trực tiếp vào một website hoặc chúng được tham chiếu qua file .js hoặc .JavaScript.
* Đây là một ngôn ngữ đến từ phía Client nên Script sẽ được download về máy client khi truy cập.
* Tại đây, chúng sẽ được hệ thống xử ý. Vì vậy, bạn không cần phải tải về máy server rồi chờ cho chúng xử lý xong mới phản hồi được kết quả đến client.

### Javascript dùng để làm gì?

Bên cạnh việc tìm hiểu javascript là ngôn ngữ gì thì chúng được sử dụng để làm gì cũng rất quan trọng. Việc nắm bắt được mục đích của ngôn ngữ đặc biệt này sẽ giúp bạn dễ dàng sử dụng chúng hơn trong công việc. Cụ thể như sau:

* Thay đổi nội dung HTML: Một trong số nhiều phương thức HTML JavaScript chính là getElementById (). Chúng được sử dụng để tìm một phần tử của HTML với id =”demo” và dùng để thay đổi nội dung của phần từ (Internal HTML) sang thành “Hello JavaScript”
* Thay đổi giá trị thuộc tính HTML: Tổng quan về javascript còn có thể sử dụng để thay đổi các giá trị của thuộc tính. Ví dụ: thay đổi thuộc tính src (source) của tag<img>.
* Thay đổi kiểu HTML: Đây chính là một hoạt động biến thể của việc thay đổi thuộc tính của HTML ở trên. Ví dụ: document.getElementById(‘demo’).style.fontSize = ’35px;
* Ẩn các phần tử HTML: Một hoạt động tiếp theo là Javascript có thể ẩn được các phần tử HTML. Chúng có thể được thực hiện thông qua hoạt động thay đổi kiểu hiển thị các phần tử HTML.
* Hiển thị các phần tử HTML: Một điểm đặc biệt là JavaScript có thể hiển thị được các yếu tố HTML ẩn. Đồng thời, cũng có thể thực hiện được thông qua cách thay đổi kiểu hiển thị phần tử.

### Ưu điểm



Hình ‑ Minh họa ưu điểm JS

* Chương trình rất dễ học.
* Những lỗi Javascript rất dễ để phát hiện, từ đó giúp bạn sửa lỗi một cách nhanh chóng hơn.
* Những trình duyệt web có thể dịch thông qua HTML mà không cần sử dụng đến một compiler.
* JS có thể hoạt động ở trên nhiều nền tảng và các trình duyệt web khác nhau.
* Được các chuyên gia đánh giá là một loại ngôn ngữ lập trình nhẹ và nhanh hơn nhiều so với các ngôn ngữ lập trình khác.
* JS còn có thể được gắn trên một số các element hoặc những events của các trang web.
* Những website có sử dụng JS thì chúng sẽ giúp cho trang web đó có sự tương tác cũng như tăng thêm nhiều trải nghiệm mới cho người dùng.
* Người dùng cũng có thể tận dụng JS với mục đích là để kiểm tra những input thay vì cách kiểm tra thủ công thông qua hoạt động truy xuất database.
* Giao diện của ứng dụng phong phú với nhiều thành phần như Drag and Drop, Slider để cung cấp đến cho người dùng một Rich Interface (giao diện giàu tính năng).
* Giúp thao tác với người dùng phía Client và tách biệt giữa các Client với nhau.

### Nhược điểm



Hình ‑ Minh họa nhược điểm của JS

* JS Code Snippet khá lớn.
* JS dễ bị các hacker và scammer khai thác hơn.
* JS cũng không có khả năng đa luồng hoặc đa dạng xử lý.
* Có thể được dùng để thực thi những mã độc ở trên máy tính của người sử dụng.
* Những thiết bị khác nhau có thể sẽ thực hiện JS khác nhau, từ đó dẫn đến sự không đồng nhất.
* Vì tính bảo mật và an toàn nên các Client-Side Javascript sẽ không cho phép đọc hoặc ghi các file.
* JS không được hỗ trợ khi bạn sử dụng ở trong tình trạng thiết bị được kết nối mạng.

### Cách viết một chương trình JS

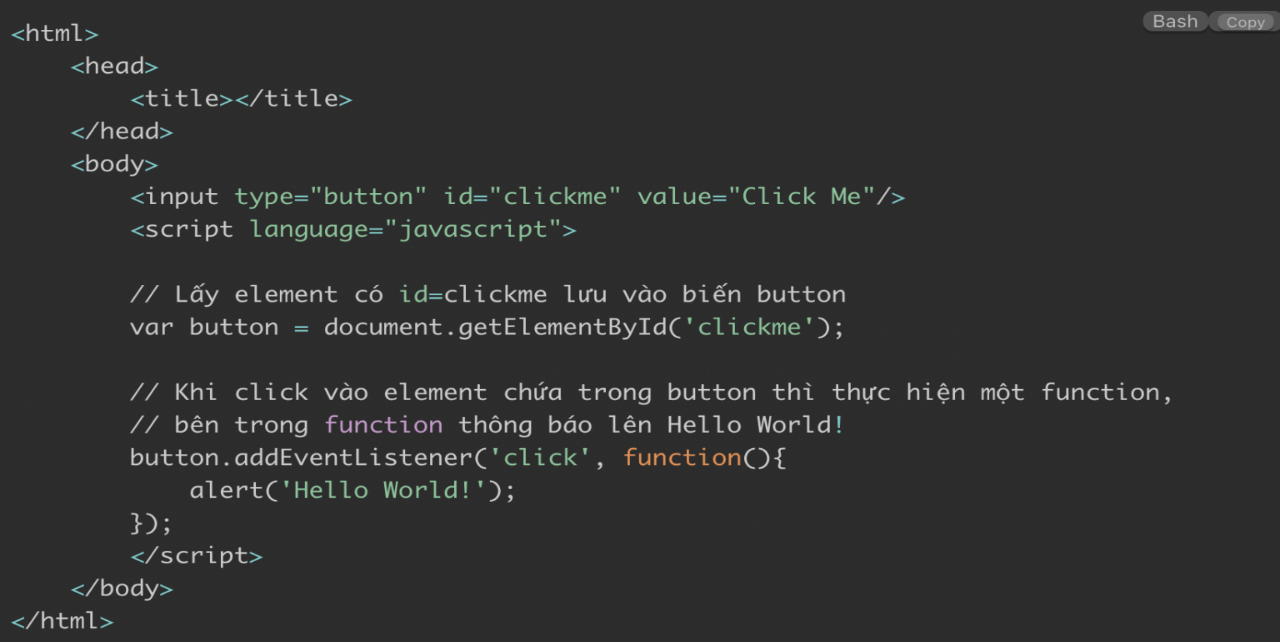
Thực ra, để viết một chương trình Javascript cũng tương đối đơn giản. Khi bạn đã tìm hiểu về javascript và biết được JS là ngôn ngữ gì thì tiếp theo chúng tôi sẽ hướng dẫn bạn cách viết chương trình đơn giản như sau:

Bước 1: Bạn tạo 1 file index.html vào bất cứ vị trí nào ở trên máy tính, miễn là chúng có phần mở rộng .html. Sau đó, bạn viết một đoạn mã như hình sau:



Hình ‑ Chương trình HTML cơ bản

Bước 2: Bạn có thể viết mã chương trình Javascript khi click chuột vào button với id=“clickme”.



Hình ‑ JS trong HTML

Chỉ với những bước đơn giản trên là bạn đã có thể hoàn tất viết một chương trình Javascript đầu tiên của mình.

### Điểm khác biệt JavaScript với ngôn ngữ lập trình khác

Sở dĩ Javascript trở thành một ngôn ngữ lập trình phổ biến nhất ngày nay là nhờ vào tính linh hoạt vốn có. Thực tế, có rất nhiều dân lập trình viên chuyên nghiệp lựa chọn JS làm ngôn ngữ chính và dùng những ngôn ngữ khác trong danh sách sau. Vậy, điểm khác biệt nào giữa JS với các loại ngôn ngữ này?

|  |  |
| --- | --- |
| Javascript | Giúp nâng cao tính tương tác ở trên web. Script chạy ở trên những trình duyệt của người dùng thay vì việc phải chạy trên server. JS thường sử dụng thư viện của một bên thứ 3 để làm tăng thêm chức năng cho web mà không cần phải code lại từ đầu. |
| HTML | Là một trong số những loại ngôn ngữ lập trình được sử dụng nhiều nhất trên web. Đồng thời, HTML còn giúp xây dựng nên những khối chính trong một trang web. |
| CSS | Cascading Style Sheets giúp cho webmaster xác định được styles. Đồng thời cũng định nghĩa được nhiều loại nội dung. Bạn có thể thực hiện thủ công với các yếu tố HTML. Tuy nhiên, bạn sẽ lặp đi lặp lại một thành phần mà bạn sử dụng ở các vị trí khác nhau. |

# GIẢI QUYẾT NHỮNG VẤN ĐỀ CƠ BẢN

## Thiết kế giao diện webside cơ bản với HTML

### Đề bài

Thiết kế giao diện web KFC như mẫu dưới



Hình ‑ Đề bài HTML

### Các phần tử sẽ sử dụng

* ***<head>*** : là bộ chứa siêu dữ liệu (metadata) được đặt trong thẻ <html> và thẻ <body>. Siêu dữ liệu HTML là dữ liệu về tài liệu HTML và không được hiển thị.Siêu dữ liệu thường định nghĩa tiêu đề văn bản, bộ kí tự, kiểu cách, đường dẫn, kịch bản và các thông tin khác
* ***<meta>***: xác định bộ kí tự được sử dụng, mô tả trang, từ khóa, tác giả và các thông tin siêu dữ liệu khác. Siêu dữ liệu được trình duyệt dùng để hiển thị nội dung, bộ tìm kiếm dùng để hiển thị từ khóa và các dịch vụ web khác.
* ***<header>***: Sử dụng ám chỉ phần đầu trang
* ***<nav>***:Thẻ này trình bày một phân đoạn của trang nó chứa các liên kết để điều hướng đến các trang của website
* ***<main>***: Phần thân trang, nó thường được dùng để chứa các nội dung chính của trang web.
* ***<footer>***:Được sử dụng khá rộng rãi với nhiều mục đích. Thông thường <footer> sử dụng ám chỉ đền phần chân trang
* ***<div>*** : viết tắt của division – là thẻ được dùng để nhóm nhiều phần tử HTML lại với nhau. Mỗi thẻ trong HTML đều có mục đích khác nhau. Ví dụ như thẻ <p> (viết tắt của paragraph) được dùng để chỉ định một đoạn văn bản nhất định, hay thẻ <br> dùng để ngắt đoạn văn bản trong HTML. Thẻ <div> là thẻ đánh dấu một khối (block) gồm nhiều thẻ khác trong khối. Như tên gọi, thẻ <div> giúp các tài liệu HTML được chia thành các phần (khối) riêng biệt.
* ***<section>***: Dùng để phân chia một cách logic một trang, một article.
* ***<h1>***: Tạo tiêu đề, đề mục trong cấu trúc trang web với các thẻ heading <h1> đến <h6>
* ***<image>***: Dùng để chèn hình ảnh vào trang web
* ***<p>***:Dùng để xác định một đoạn văn bản
* ***<ul>,<li>***: Thẻ <ul> dùng để xác định một "danh sách không có thứ tự".Tuy nhiên, để tạo các danh mục trong danh sách thì phải sử dụng kết hợp với thẻ <li>.
* ***<table>***: Định nghĩa bảng.
* ***<tr>***: Định nghĩa một hàng trong một bảng.
* ***<th>***: Định nghĩa phần header (dòng đầu tiên) của bảng.
* ***<td>*** : Định nghĩa 1 ô của bảng.

### Kết quả

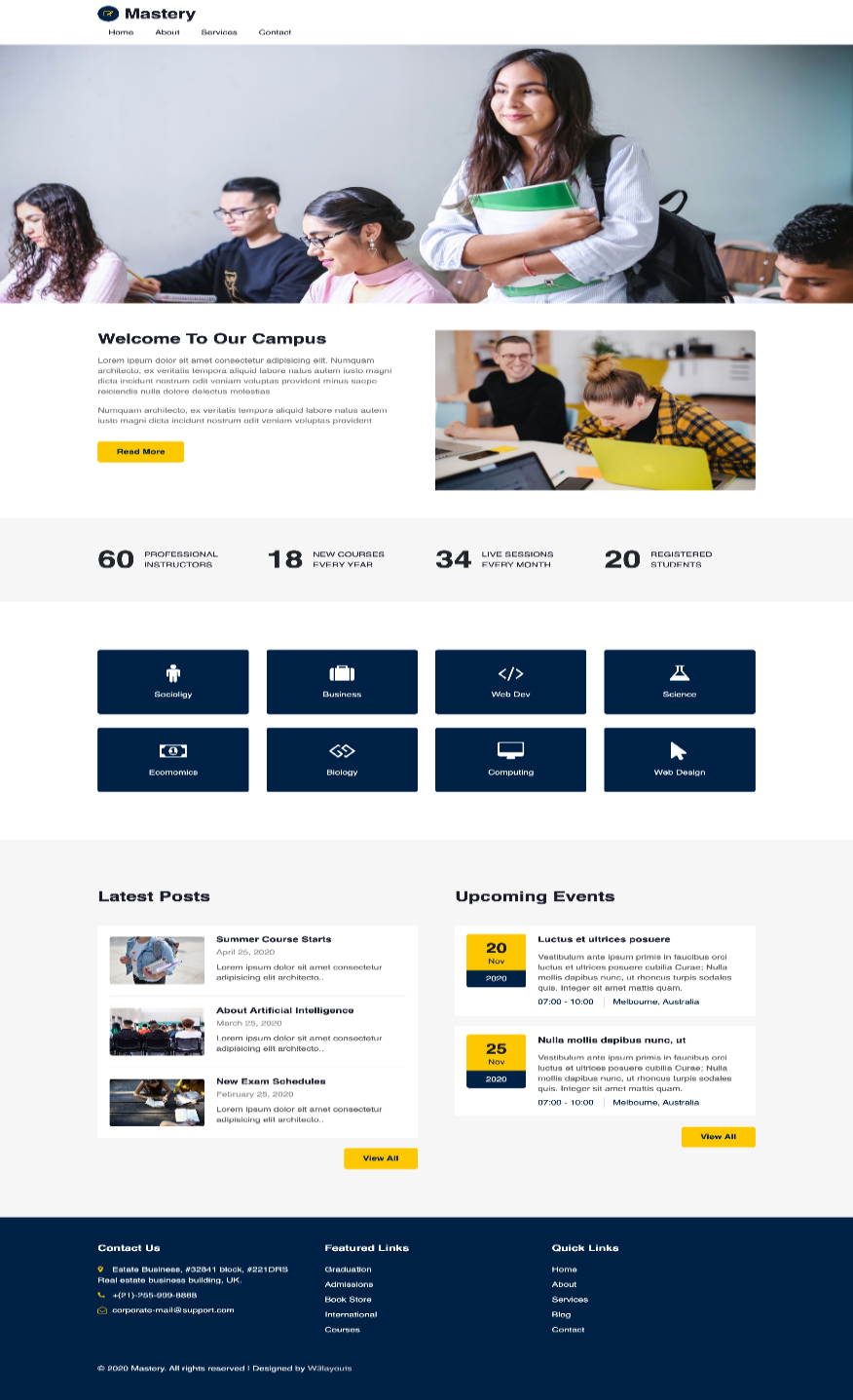


Hình ‑ Kết quả web HTML

## Thiết kế giao diện webside cơ bản với HTML kết hợp CSS

### Đề bài

Thiết kế web kết hợp HTML và CSS như hình dưới



Hình ‑ Đề bài CSS

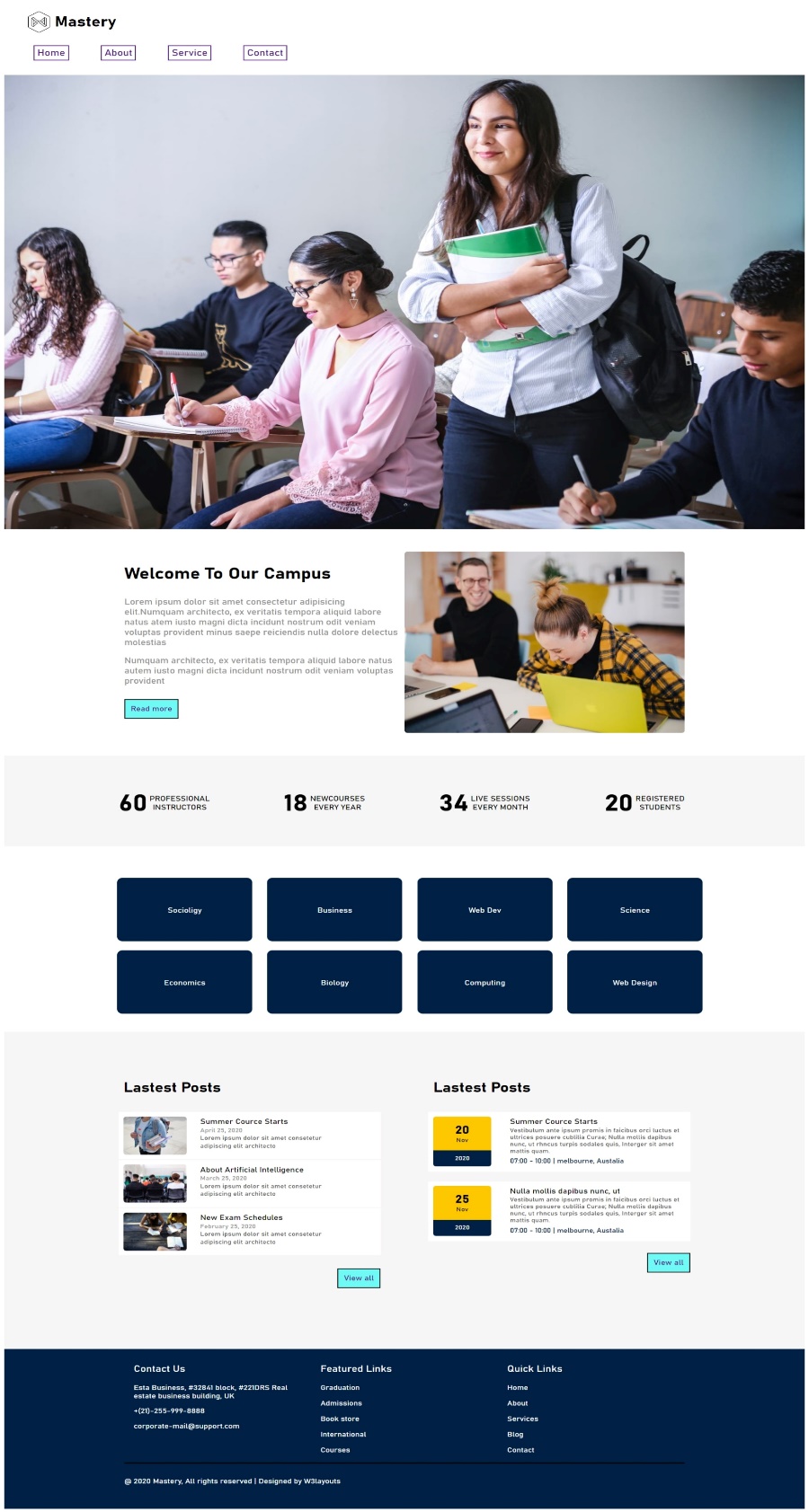
### Các thẻ và thuộc tính sử dụng

***HTML***:Như các thẻ ở bài trước

**CSS**: gồm các thuộc tính thường quan trọng sau :

* ***display***:xác định thành phần hiển thị theo các kiểu khác nhau
* ***width***:thiết lập chiều rộng cho thành phần
* ***height***: thiết lập chiều cao cho thành phần
* ***text-align***: Thuộc tính text-align sắp xếp các nội dung theo chiều ngang.
* ***color***: Dùng để chỉ màu chữ cho văn bản
* ***background-color***: Dùng để thiết lập "màu nền" của phần tử HTML.
* ***margin***: Dùng để canh lề cho thành phần
* ***padding*** :Thuộc tính padding thêm vào khoảng không cho thành phần.
* ***border***: Được dùng để định dạng đường viền cho thành phần.
* ***font-size***: Thiết lập kích cỡ chữ cho văn bản
* ***font-family***: Thiết lập font chữ cho văn bản
* ***position***: được sử dụng để định vị vị trí hiển thị của các phần tử thẻ HTML và thường sử dụng để xây dựng CSS cho menu nhiều cấp, tooltip và một số chức năng khác liên quan đến vị trí.
* ***left***: Thuộc tính left được dùng để định vị trí bên trái (left) cho thành phần khi sử dụng thuộc tính position, với khoảng cách được tính từ mép trái ngoài cùng của thành phần bao ngoài.
* ***top***: Thuộc tính top được dùng để định vị trí bên trên (top) cho thành phần khi sử dụng thuộc tính position, với khoảng cách được tính từ mép trên ngoài cùng của thành phần bao ngoài.
* ***right***: Thuộc tính right được dùng để định vị trí bên phải (right) cho thành phần khi sử dụng thuộc tính position, với khoảng cách được tính từ mép phải ngoài cùng của thành phần bao ngoài.
* ***bottom***: Thuộc tính bottom được dùng để định vị trí dưới (bottom) cho thành phần khi sử dụng thuộc tính position, với khoảng cách được tính từ mép dưới cùng của thành phần bao ngoài.

### Kết quả

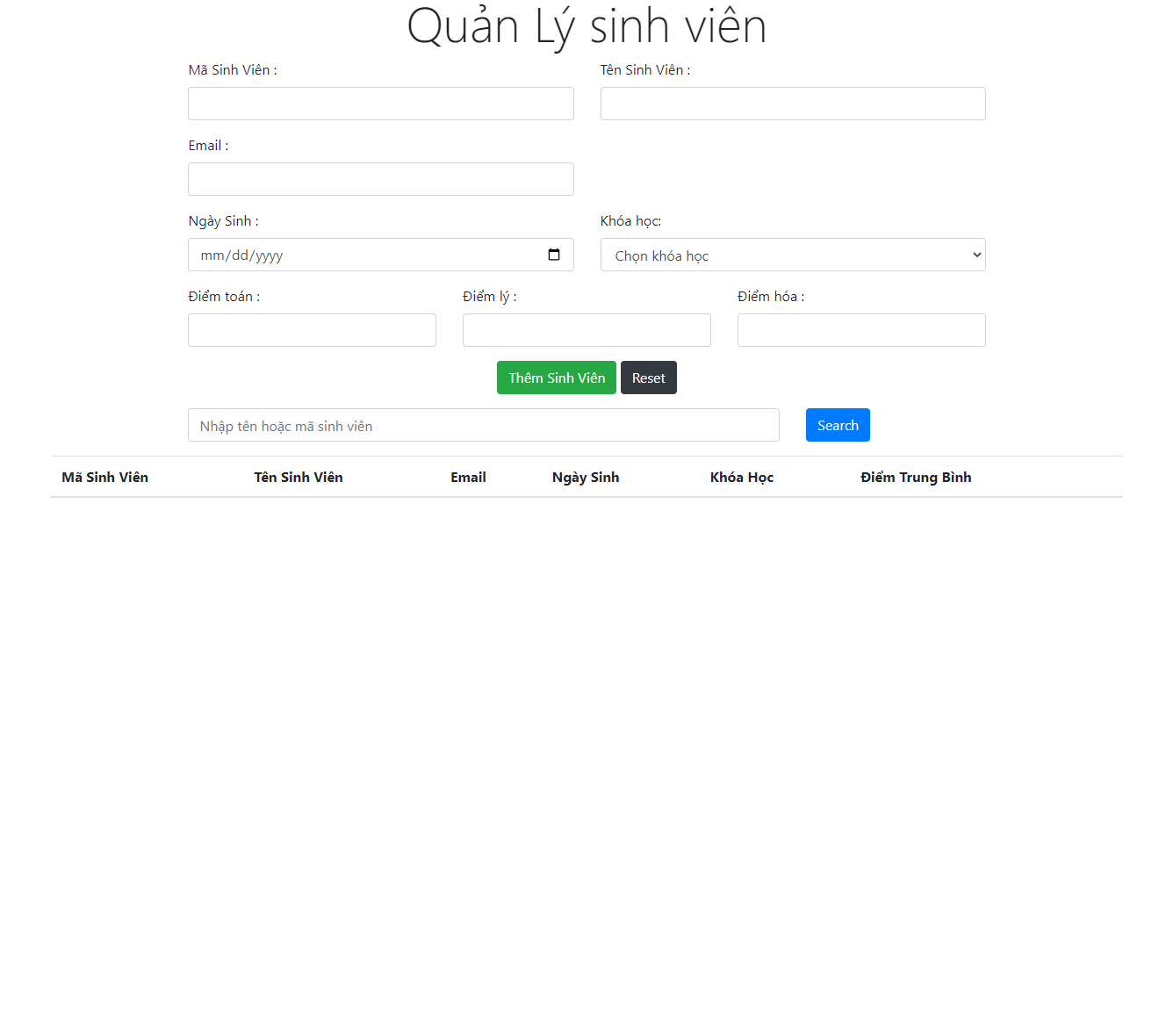


Hình ‑ Kết quả bài CSS

## Thiết kế ứng dụng quản lý học sinh trên web

### Đề bài

Thiết kế web quản lý sinh viên như hình



Hình ‑ Đề bài JS

### Các lệnh sẽ dùng

***HTML:*** Như những bài trước

***CSS:*** Tương tự bài trước

***JavaScript*:**

* ***var***: Khai báo biến (funtion scope)
* ***let***: Khai báo biến (block scope)
* ***class***: Định nghĩa một class
* ***funtion***: Định nghĩa một hàm
* ***for***: Vòng lặp khi xác định sẵn số lần lặp
* ***while***: Vòng lặp theo điều kiện
* ***break***: Thoát khỏi một vòng lặp
* ***if***: Câu điều kiện
* ***document.getElementById***: Phương thức trả về phần tử có thuộc tính ID là giá trị được chỉ định
* ***addEventListener***: addEventListener là một phương thức được tích hợp sẵn vào các đối tượng HTML thông qua cơ chế DOM. Khi sử dụng addEventListener thì bạn có thể bổ sung rất nhiều hành động vào sự kiện tại nhiều thời điểm khác nhau.

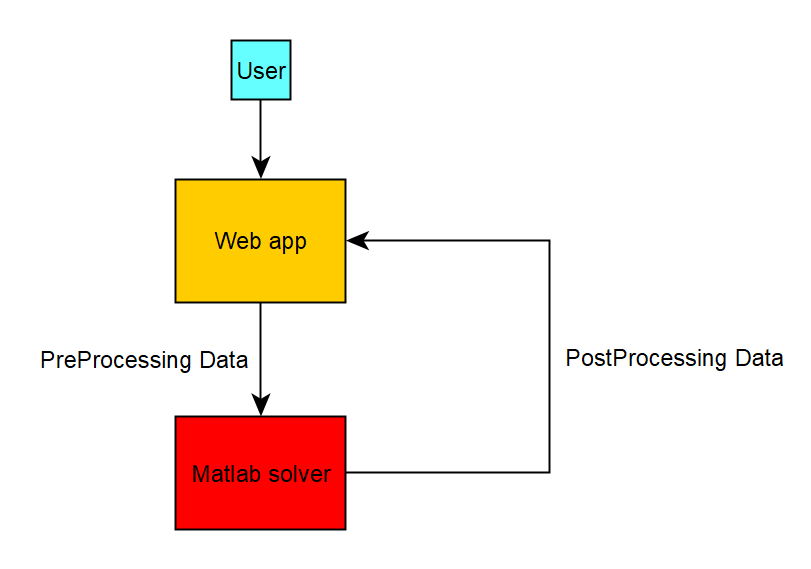
### Kết quả



Hình ‑ Kết quả bài JS

# PHÁT TRIỂN ỨNG DỤNG

Quá trình thực tập, nhiệm vụ chính của nhóm là phát triển một ứng dụng xây dựng mô hình bài toán phần tử hữu hạn trên môi trường web.Ứng dụng có giao diện được tối giản để tương thích với người dùng, cho phép đơn giản hóa quá trình mô hình những bài toán.Sau đó dữ liệu mô hình sẽ được đưa matlab sử lý và trả kết quả về web app và tương tác với người dùng.

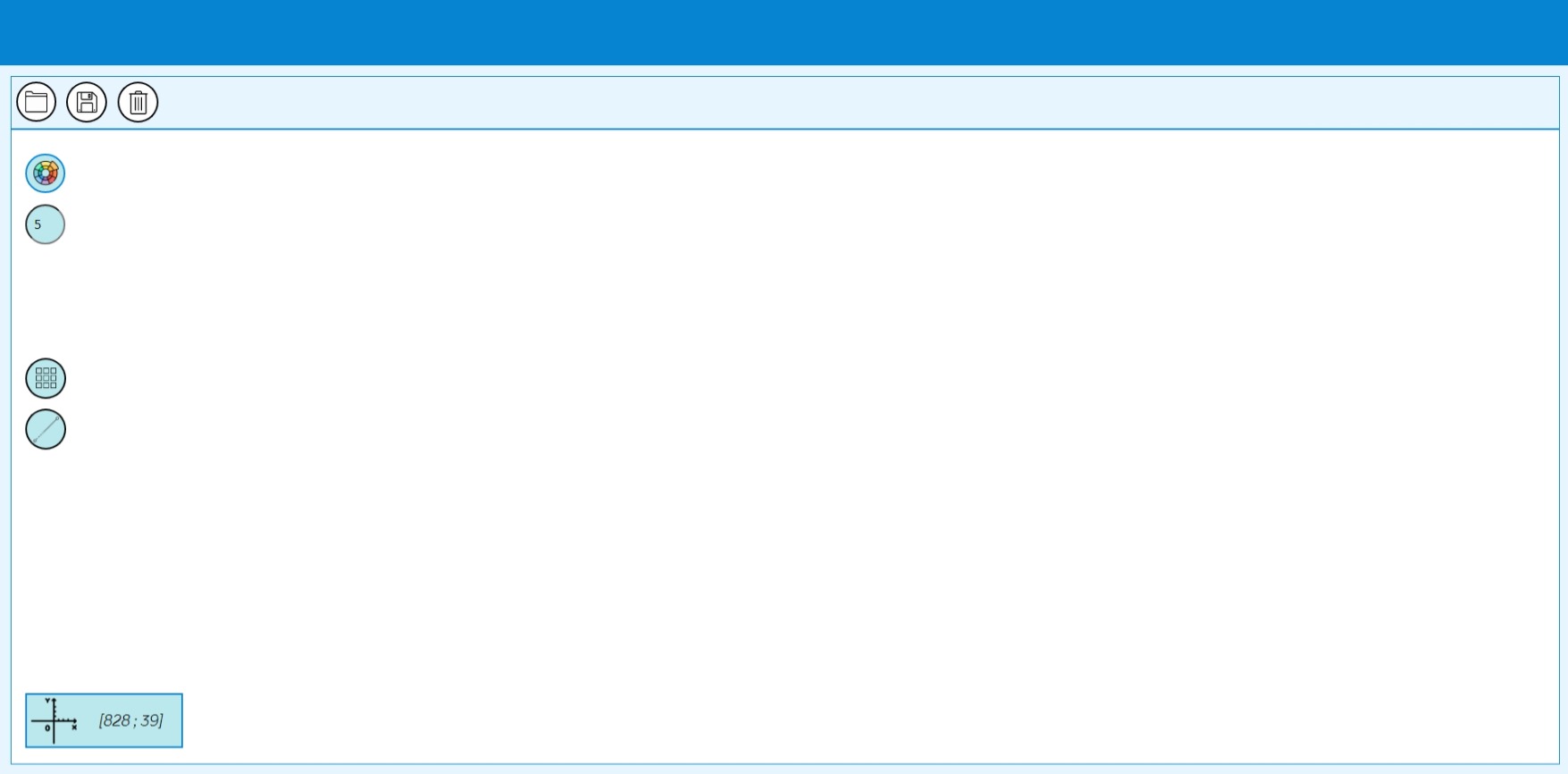


Hình ‑ Quy trình xử lý FE problem

## Tạo ứng dụng có thể tương tác đồ họa với người dùng

Sử dụng thẻ *<canvas>* trong HTML để tạo một cùng đồ họa cho phép vẽ lên những đối tượng đồ họa, thẻ *<canvas>* có thể tương tác với người dùng thông qua JavaScript.

* *<canvas>*: Sử dụng để vẽ các thành phần đồ họa từ đơn giản như các đường đến các đồ họa phức tạp (đường thẳng, hình hộp, hình tròn, viết chữ, chèn hình ảnh...).  
  Cũng tương tự như những thẻ HTML khác, để sử dụng thì bạn phải đặt thẻ *<canvas></canvas>* tại vị trí muốn hiển thị. Tuy nhiên có một sự khác biệt rõ rệt nhất là nội dung của nó lại được xây dựng từ Javascript.*<canvas>* chỉ có chức năng chứa các đối tượng đồ họa. Bạn bắt buộc phải sử dụng JavaScript để vẽ.
* Khai báo một thẻ <canvas>:  
   <canvas id="myCanvas" width="200" height="100"></canvas>



Hình ‑ Vùng đồ họa Canvas

## Tạo những tính năng vẽ cơ bản cho ứng dụng

Ta Listen những Event tương tác với vùng canvas, xử lý những Event đó và lấy những dữ liệu cần thiết để xác định đối tượng.Sử dụng các MouseEvent:

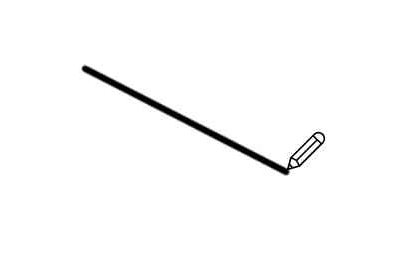
* *MouseDown*: Event được kích hoạt tại một element khi chuột được nhấn trong khi con trỏ chuột đang ở trong phần tử đó.
* *MouseUp*: Event được kích hoạt tại một element khi chuột được thả ra trong khi con trỏ chuột đang ở trong phần tử đó.
* *MouseMove*: Event được kích hoạt tại một element khi chuột được di chuyển ra trong khi con trỏ chuột đang ở trong phần tử đó.
* *Event.clientX, Event.clientY*: Cho tọa độ của của ứng dụng mà tại đó sự kiện đã xảy ra.

Và các phương thức vẽ 2D trên canvas:

* *beginPath()*: Phương thức khởi tạo một Path hoặc đặt lại Path hiện tại
* *moveTo()*: Phương thức di chuyển Path đến điểm chỉ định mà không cần tạo một đường thẳng
* *lineTo()*: Phương thức thêm điểm mới và tạo một line từ điểm được chỉ định cuối cùng của Path đến điểm đó
* *closePath()*: Phương thức tạo một Path từ điểm hiện hành đến điểm bắt đầu của Path (tạo thành Path đóng)
* *stroke()*: Phương thức vẽ Path trên canvas
* *arc()*: Phương thức vẽ đường tròn hoặc một cung

### Line

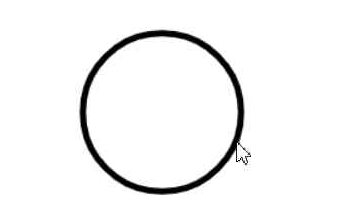
* Khi MouseDown được kích hoạt, ta lưu lại tọa độ đó vào 1 mảng.
* Tiếp tục khi MouseMove được kích hoạt, ta lưu lại tọa độ đó vào 1 mảng khác đồng thời render 1 line trên canvas với tọa độ đầu đã lưu và tọa độ điểm sau là điểm vừa được lưu bằng tổ hợp các phương thức *beginPath()*, *moveTo()*, *lineTo()*, *ctx.stroke().*



Hình ‑ Vẽ đường thẳng trên canvas

### Circle

Tương tự Line nhưng từ những tọa độ đã lưu, ta xác định được bán kính và dùng phương thức *arc()* để tạo Path và vẽ trên canvas.



Hình ‑ Vẽ hình tròn trên canvas

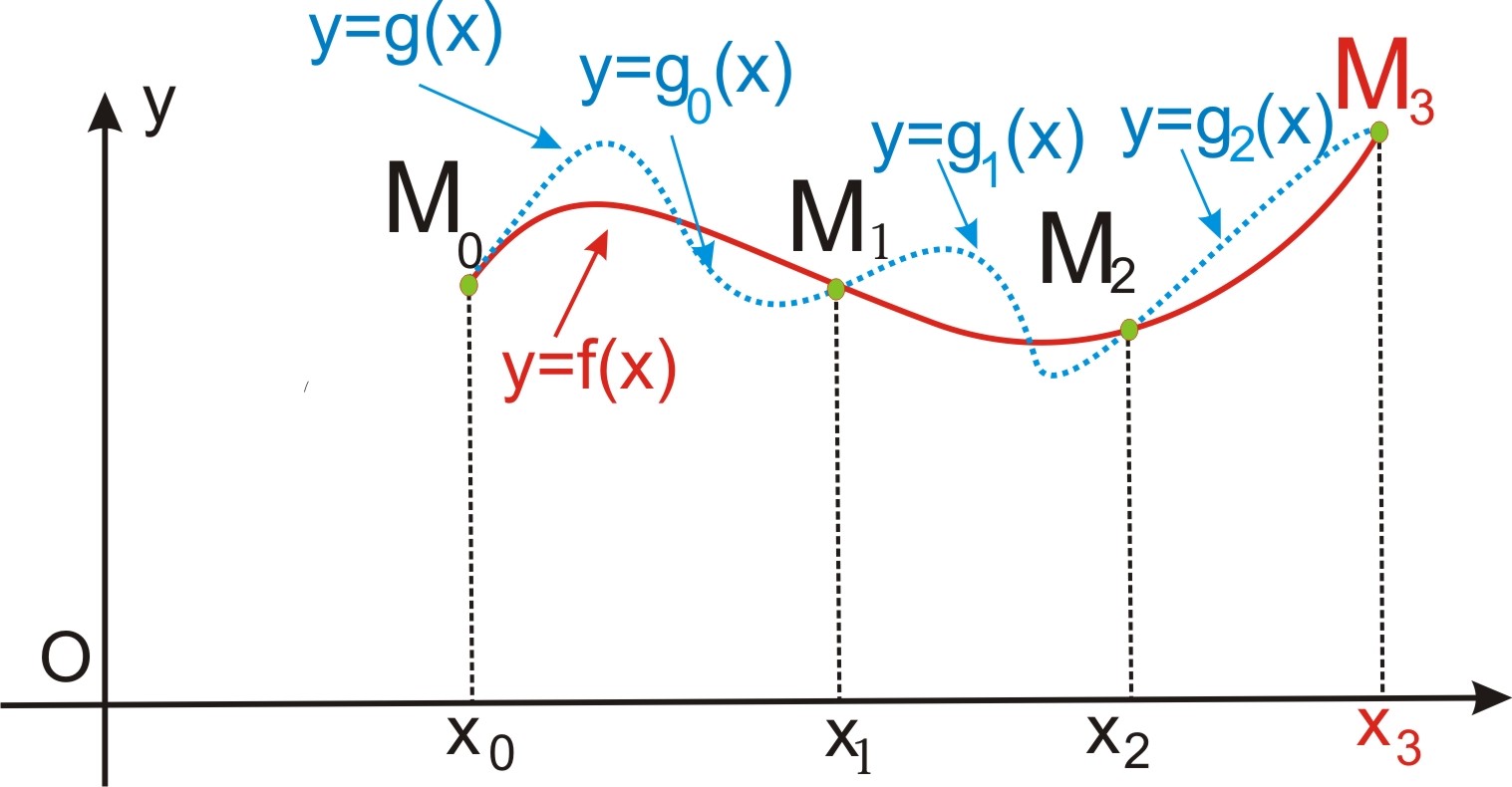
### Spline

Sử dụng các lý thuyết toán học để vẽ đường spline:

* ***Nội suy spline***Việc xây dựng một đa thức đi qua các điểm nội suy cho trước trong trường hợp n lớn là rất khó khăn. Biện pháp khắc phục là trên từng đoạn liên tiếp của các cặp điểm nút nội suy ta nối chúng bởi các đường cong đơn giản như đoạn thẳng. Tuy nhiên, khi đó tại các điểm nút hàm sẽ mất tính khả vi. Do đó, phải xây dựng đường cong bằng cách nối các đoạn cong nhỏ lại với nhau sao cho vẫn bảo toàn tính khả vi của hàm. Đường cong như vậy được gọi là đường spline (đường ghép trơn). Các hàm trên các đoạn nhỏ này thường là các đa thức và bậc cao nhất của các đa thức đó gọi là bậc của spline.  
  **Định nghĩa:**Cho  xác định trên đoạn  và một phép phân hoạch của nó: . Đặt , , . Một spline bậc ba nội suy hàm trên  là hàm  thỏa các điều kiện sau:  
   có đạo hàm đến cấp 2 liên tục trên 

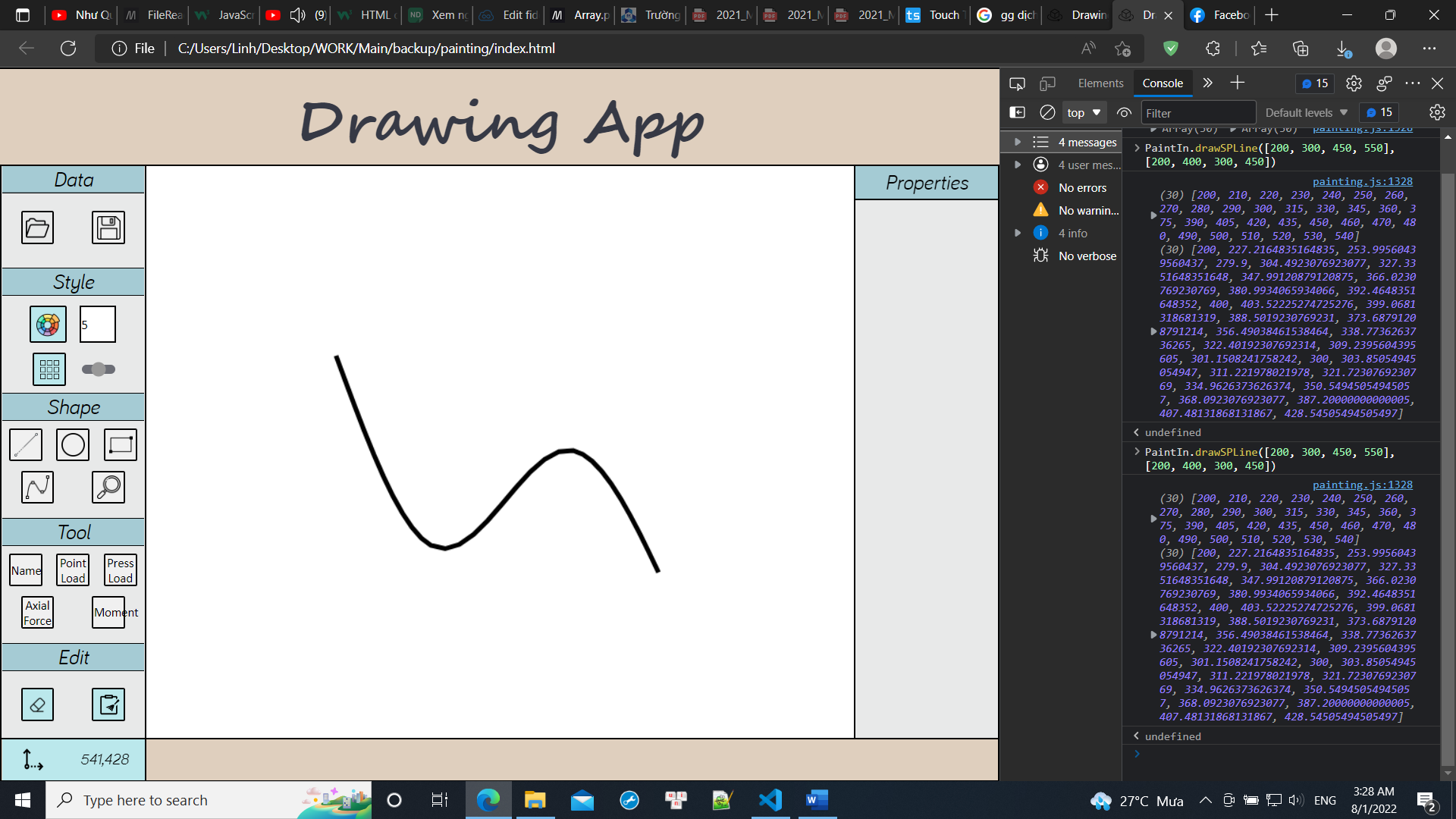
 ở đây ,  là các đa thực bậc 3

, ,   
**Định nghĩa:** Cho  xác định trên đoạn  và một phép phân hoạch của nó: . Đặt . Một spline bậc ba nội suy hàm trên  là hàm  thỏa các điều kiện sau:  
 có đạo hàm đến cấp 2 liên tục trên   
Trên mỗi đoạn ,  là các đa thực bậc 3



Hình ‑ Minh họa spline bậc 3

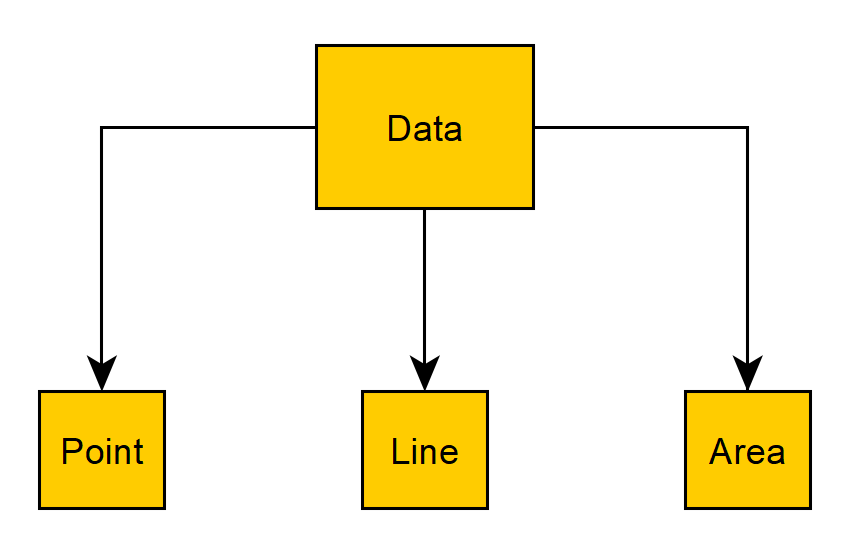
* **Spline bậc 3 tự nhiên**Điều kiện để xác định spline tự nhiên là  
  Giải hệ  tìm  với  và,  
  Sau khi tìm được thì các hệ số  được xác định bởi
* Từ n điểm rời rạc, viết phương trình của spline bậc 3 trên n-1 đoạn
* Sau đó, từ hàm của *spline* ta tính một số lượng đủ lớn các điểm thuộc *spline*
* Nối các điểm rời rạc bằng các đường thẳng bằng phương thức như vẽ Line



Hình ‑ Spline tạo từ những đoạn thẳng nhỏ

## Lưu những dữ liệu thành đối tượng

Dựa theo cấu trúc bài toán phần tử hữu hạn, tạo 3 loại đối tượng bao gồm Point, Line,Area.



Hình ‑ Cấu trúc dữ liệu

### Point

Tạo một *class* Point gồm các *thuộc tính* cơ bản sau :

* *className*: tên class của đối tượng (mặc định ’Point‘ cho class Point)
* *point*: là mảng chưa tọa độ của Point
* *x*: là tọa độ x của đối tượng
* *y*: là tọa độ y của đối tượng
* *name*: tên của đối tượng

### Line

Tạo một *class* Line gồm các *thuộc tính* cơ bản sau :

* *className*: Tên class của đối tượng (mặc định ’Line‘ cho class Line)
* *Point*: Là mảng chứa các đối tượng Point xác định Line
* *name*: Tên của đối tượng
* *length :* Độ dài của đối tượng

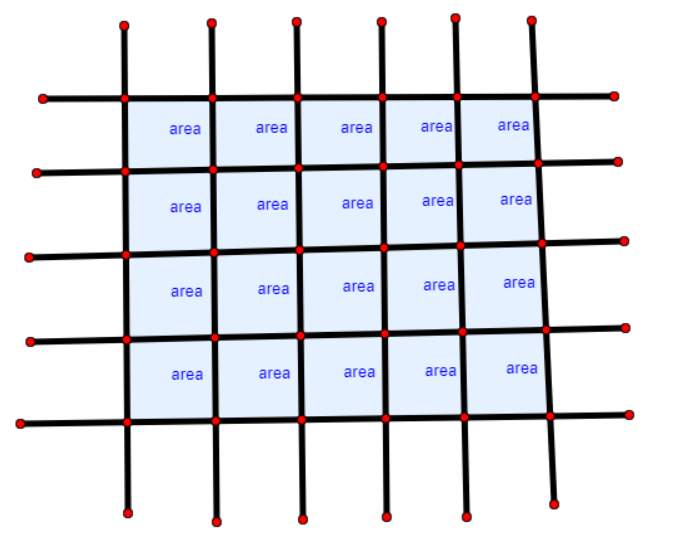
### Area

Tạo một *class* Area gồm các *thuộc tính* cơ bản sau :

* *className*: Tên class của đối tượng (mặc định ’Area‘ cho class Area)
* *Line*: Là mảng chứa các đối tượng cạnh của Area
* *name*: Tên của đối tượng
* *perimeter :* Chu vi của đối tượng
* *area :* Diện tích của đối tượng
* *center*: Là mảng chứa tọa độ tâm của đối tượng

## Tạo tính năng phát hiện diện tích

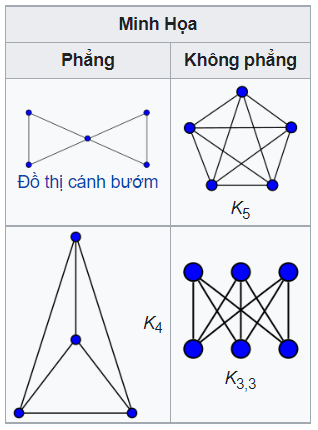
Cho những đối tượng là đoạn thẳng ***xác định***cắt nhau một cách ngẫu nhiên như hình dưới.Yêu cầu hãy ***xác định*** các đối tượng diện tích được tạo ra (ví dụ như các diện tích phía dưới)



Hình ‑ Minh họa phát hiện diện tích

Sử dụng các lý thuyết về phương trình và đồ thị

* Phương trình tham số  
  Qua 2 điểm, bất kỳ ta viết đường phương trình tham số của đoạn thẳng  nối 2 điểm đó  *với*  là vector chỉ phương, biến đổi một chút về dạng ma trận ta được 
* Một đồ thị được gọi là một đồ thị phẳng nếu ta có thể vẽ nó trên mặt phẳng sao cho các cạnh của nó không cắt nhau ngoài ở đỉnh. Cách vẽ như vậy sẽ được gọi là biểu diễn phẳng của đồ thị. Ví dụ  là một đồ thị phẳng.Đồ thị là phẳng khi và chỉ khi nó không chứa đồ thị con đồng phôi với  hoặc 

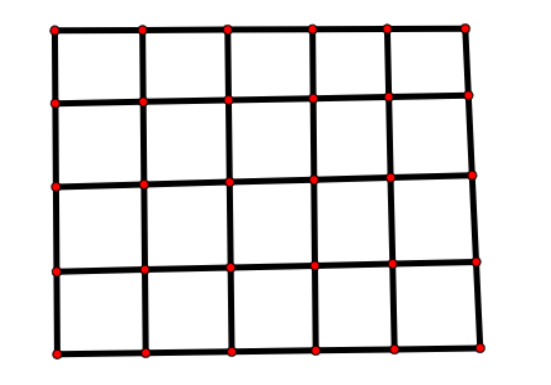


Hình ‑ Các loại đồ thị phẳng

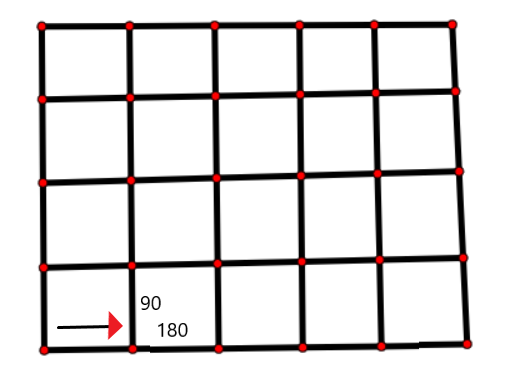
Euler đã chứng minh được rằng các biểu diễn phẳng khác nhau của một đồ thị đều chia mặt phẳng ra thành cùng một số miền qua định lý sau (thường được gọi là **công thức Euler**):  
*Giả sử một đồ thị phẳng liên thông có n đỉnh, m cạnh, r miền. Khi đó r=m-n+2.*Chứng minh: Cho G là đồ thị phẳng liên thông có n đỉnh, m cạnh và r miền. Ta bỏ một số cạnh của G để được một cây khung của G. Mỗi lần ta bỏ một cạnh (m giảm 1) thì số miền của G cũng giảm 1 (r giảm 1), còn số đỉnh của G không thay đổi (n không đổi). Như vậy, giá trị của biểu thức n - m + r không thay đổi trong suốt quá trình ta bỏ bớt cạnh của G để được một cây. Cây này có n đỉnh, do đó có n - 1 cạnh và cây chỉ có một miền, vì vậy: n - m + r = n - (n -1) + 1 = 2.  
Hệ thức n - m + r = 2 thường gọi là hệ thức Euler cho hình đa diện, vì được Euler chứng minh đầu tiên cho hình đa diện có n đỉnh, m cạnh và r mặt.Một hệ quả đơn giản là:  
*Trong một đồ thị phẳng liên thông luôn tồn tại ít nhất một đỉnh có bậc không vượt quá 5.*Chứng minh: Trong đồ thị phẳng mỗi miền được bao bằng ít nhất 3 cạnh. Mặt khác, mỗi cạnh có thể nằm trên biên của tối đa hai miền. Gọi d là số miền, thì 3d ≤ 2p. Nếu trong đồ thị phẳng mà tất cả các đỉnh đều có bậc không nhỏ hơn 6 thì do mỗi đỉnh của đồ thị phải là đầu mút của ít nhất 6 cạnh mà mỗi cạnh lại có hai đầu mút nên ta có 6n ≤ 2p hay 3n ≤ p. Từ đó suy ra 3d+3n ≤ 2p+p hay d+n ≤ p, trái với hệ thức Euler d+n = p+2.

**Giải thuật**

* Xác định giao điểm của những đoạn thẳng  
  Mỗi đoạn thẳng ta viết được 1 phương trình tham số.Ta có 2 đoạn thẳng có phương trình tham số lần lượt là :  
   và   
  Giao điểm thuộc cả 2 đường thẳng nên ta có:  
    
  Giải hệ phương trình ta tìm được , thế vào 1 trong 2 phương trình của 2 đường thẳng.
* Bỏ những đoạn thẳng không là cạnh của diện tích => thiết lập được một *planar graph*

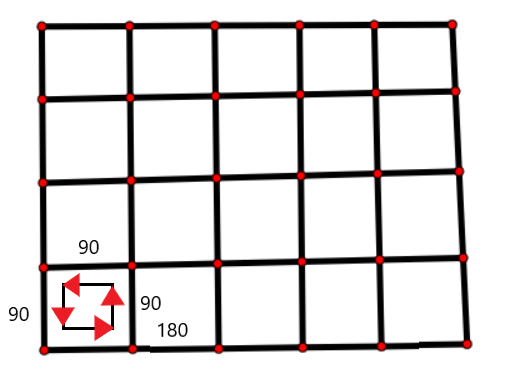


Hình ‑ Planar grahp

* Chọn một cạnh ngẫu nhiên, chọn 1 định hướng(CW, CCW)=> sắp xếp các cạnh tiếp theo theo giá trị góc  
  

Hình ‑ Định hướng trái sang phải

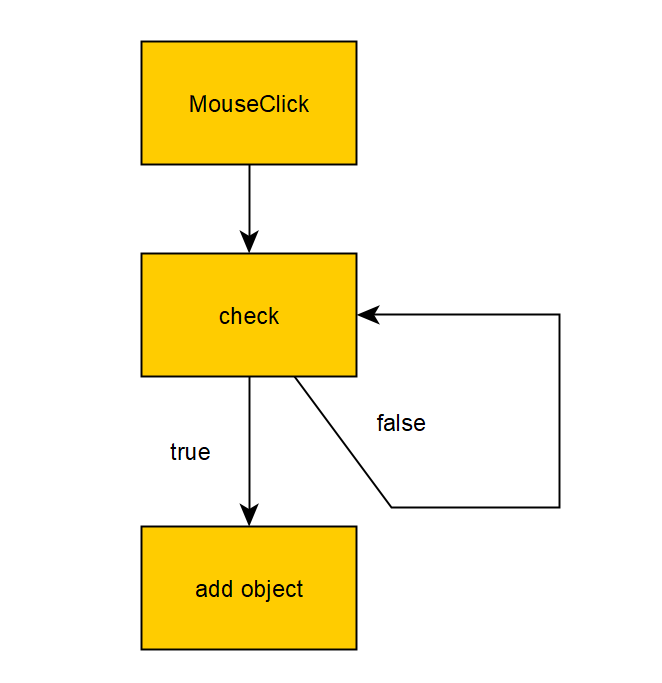
* Chọn cạnh tiếp theo theo góc (góc lớn nhất hay bé nhất tùy vào định hướng)
* Tiếp tục đến khi cạnh tiếp theo là cạnh đầu tiên. Ta sẽ nhận được một diện tích



Hình ‑ Xác định diện tích

## Tạo tính năng tương tác với đối tượng

Muốn tương tác với đối tượng, ta phải phát hiện được đối tượng bằng con trỏ chuột.Lấy vị trí của trỏ chuột bằng event và kiểm tra từng đối tượng có đang tại vị trí đó.



Hình ‑ quá trình kiểm tra đối tượng

### Click chọn

Kiểm tra bằng phương thức *isIn()*

* *isIn()* cho class Point : Nhận tham số là tọa độ chuột, kiểm tra có ở gần đối tượng trong một bán kính cho phép, nếu có trả về *true*
* *isIn()* cho class Line : Nhận tham số là tọa độ chuột, tính khoảng cách từ đó đến 2 điểm đầu và cuối của đối tượng, nếu độ dài đối tượng bằng tổng độ dài 2 đoạn thì trả về *true*
* *isIn()* cho class Area : Nhận tham số là tọa độ chuột, sử dụng thuật toán *raycasting* tạo một tia từ vị trí chuột, đếm số lần cắt các cạnh của đối tượng, nếu là lẻ thì trả về *true*

### Quét chọn

Kiểm trà bằng phương thức *isInBox()*

* *isInBox()* cho class Point : Nhận 2 tham số là tọa độ của 2 góc vùng chọn, kiểm tra điểm có nằm trong vùng quét chọn đó, nếu có trả về *true*
* *isInBox()* cho class Line : Nhận 2 tham số là tọa độ của 2 góc vùng chọn, kiểm tra 2 điểm đầu và cuối của đối tượng có nằm trong vùng quét chọn đó, nếu có trả về *true*
* *isInBox()* cho class Line : Nhận 2 tham số là tọa độ của 2 góc vùng chọn, kiểm tra tất cả điểm của đối tượng có nằm trong vùng quét chọn đó, nếu có trả về *true*

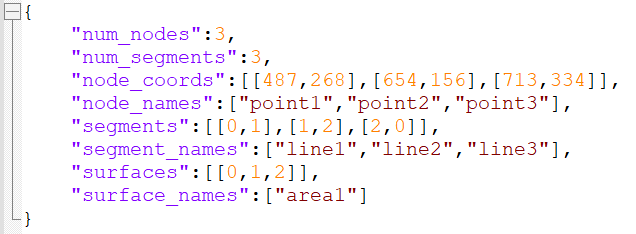
## Import & export dữ liệu

Các đối tượng có nhiều phương thức và thuộc tính phụ, trong quá trình xuất và nhập dữ liệu, ta chỉ đưa về những dữ liệu cơ bản nhất để *xác định* được đối tượng

* *Point*: chỉ lưu thuộc tính tọa độ, tên
* *Line*: chỉ lưu tọa độ điểm đầu, cuối, tên
* *Area*: chỉ lưu tọa điểm các điểm góc và tên

Sau đó đưa vào thành một đối tượng và lưu vào file *json* có các thuộc tính :

* *Num\_nodes*: số điểm của toàn model
* *Num\_segments*: số đoạn thẳng của toàn model
* *Node\_coords*: chứa tọa độ các điểm
* *Node\_names*: chưa tên của các điểm
* *Segments*: chứa thông tin xác định của các đoạn thẳng
* *Segment\_names*: chứa tên của các đoạn thẳng
* *Surfaces*: chứa thông tin xác định của các diện tích
* *Surface\_names*: chưa tên của các diện tích



Hình ‑ cấu trúc file export

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| 1. | https://en.wikipedia.org/wiki/Planar\_graph |
| 2. | https://tmarketing.vn/tong-quan-ve-html/ |
| 3. | https://hanhtranglaptrinh.net/css-la-gi-tong-quan-ve-css/ |
| 4. | Slide bài giảng môn *Phương Pháp Tính* HCMUT |