Logo, company name

Description automatically generatedA black and white logo

Description automatically generated with medium confidencetrường đại học kiến trúc hà nôi
 J?NM.

**Giảng viên hướng dẫn: Nguyễn Thị Hạnh**

Đề tài

Thiết kế giao diện cho nhà hàng

**Môn học**

**Công nghệ Wed**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

Thành viên nhóm:

Nguyễn Việt Hoàng (Nhóm trưởng)

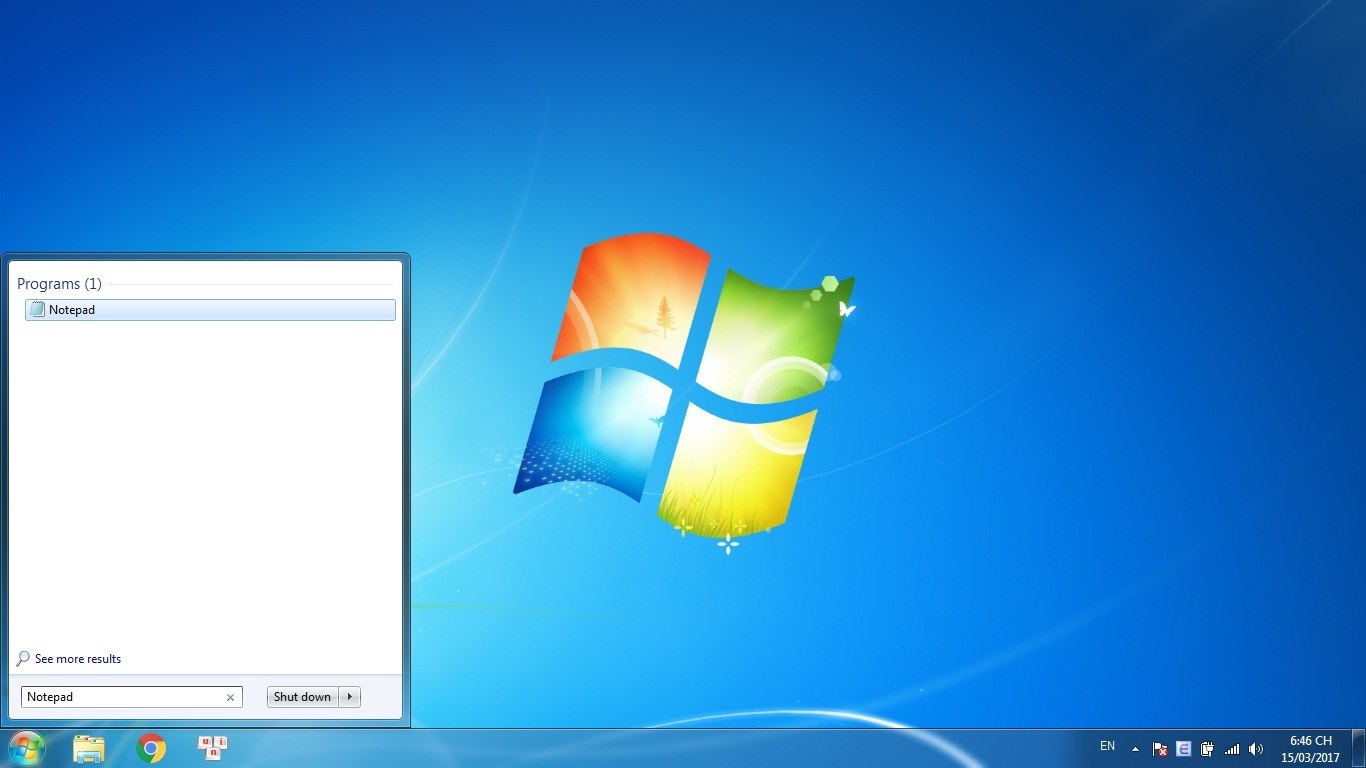
Nguyễn Quang Hải

Phạm Đình Trường Sơn

**1 Giới thiệu công cụ soạn thảo mã HTML**

- Ngày nay có rất nhiều công cụ hiện đại hỗ trợ việc soạn thảo mã HTML, ví dụ như: Adobe Dreamweaver, CoffeeCup, TextWrangler, Sublime Text, . . . . Tuy nhiên, các công cụ này thường tương đối phức tạp, chúng chỉ thích hợp với những người đã thông thạo về lập trình web, còn đối với những người mới học HTML giống như chúng ta thì tốt nhất là nên dùng các công cụ soạn thảo thuần văn bản, một mặt nó sẽ giúp ta rèn luyện kỹ năng gõ mã lệnh, mặt khác chúng ta có thể ghi nhớ được tên của từng loại thẻ.

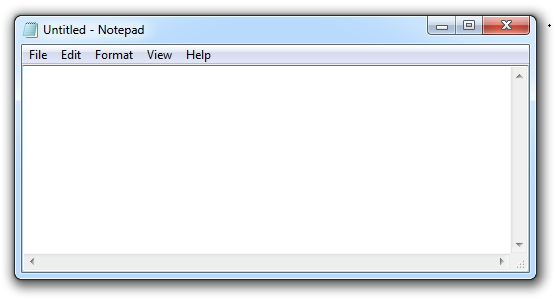
- Mà nói đến trình soạn thảo thuần văn bản thì tôi xin giới thiệu đến các bạn một công cụ có sẵn trên máy tính cài hệ điều hành Windows, đó chính là Notepad (tùy vào phiên bản mà cách mở sẽ khác nhau, nhưng nếu các bạn đang dùng Windows 7 thì các bạn có thể mở bằng cách vào Start rồi gõ từ khóa Notepad)

[](https://webcoban.vn/image/cach-tao-va-chay-mot-tap-tin-html.jpg)

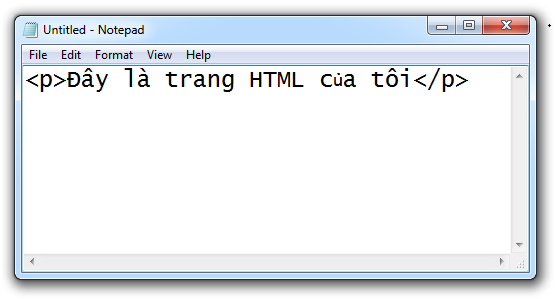
**2) Cách tạo một tập tin HTML**

- Để tạo một tập tin HTML thì chúng ta thực hiện lần lượt các bước như sau:

- Bước 1: Mở trình soạn thảo thuần văn bản Notepad lên.

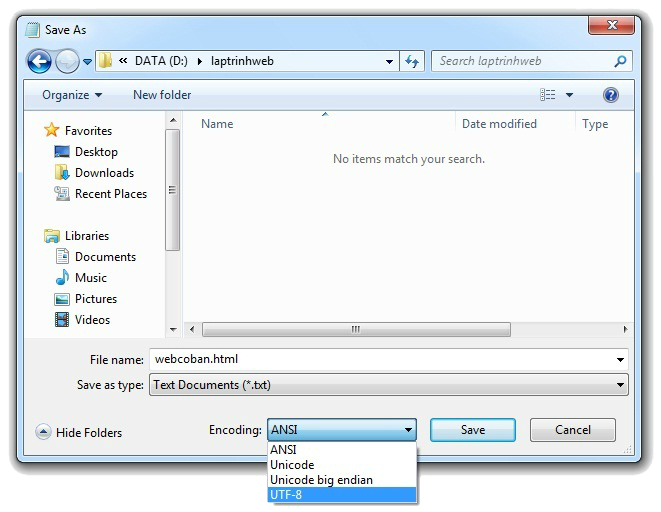


- Bước 2: Nhập những nội dung mà các bạn muốn chúng được hiển thị trên trang web.



- Bước 3: Lưu tập tin.

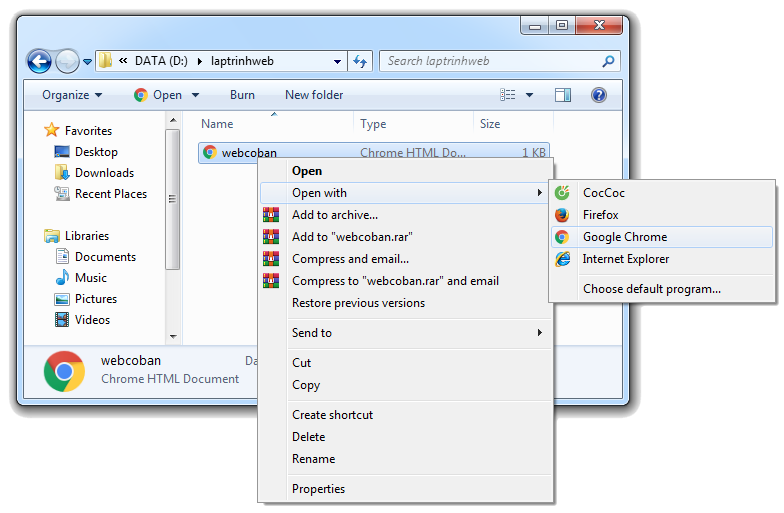
1. Bấm vào nút **File** rồi chọn **Save As**.
2. Chọn thư mục muốn lưu tập tin.
3. Đặt tên cho tập tin.
4. Chỗ Encoding chọn **UTF-8**.
5. Cuối cùng, bấm nút **Save** để lưu lại.



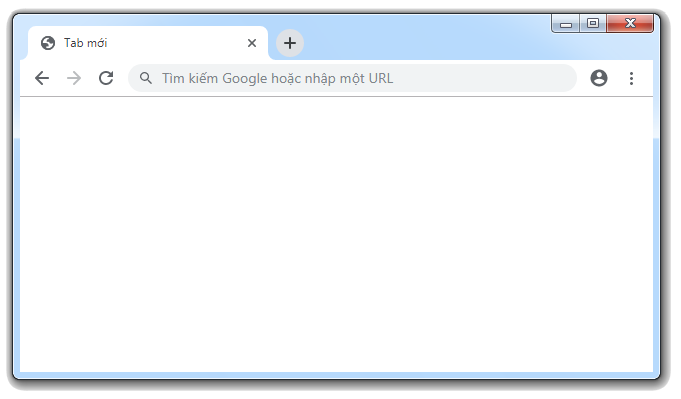
|  |  |
| --- | --- |
|  | - MỘT SỐ ĐIỀU CẦN LƯU Ý:   * Tên của tập tin tuyệt đối không được chứa dấu khoảng trắng & các ký tự đặc biệt. * Tên của tập tin phải có phần đuôi là .html hoặc .htm. Tuy nhiên, phần đuôi .htm có một số hạn chế, nên chúng tôi khuyến khích các bạn sử dụng phần đuôi .html * Chỗ Encoding chọn UTF-8 là một bước vô cùng quan trọng, kiểu mã hóa UTF-8 sẽ giúp cho trang web hiển thị ngôn ngữ tiếng Việt được chính xác. |

**3) Cách chạy (thực thi) một tập tin HTML**

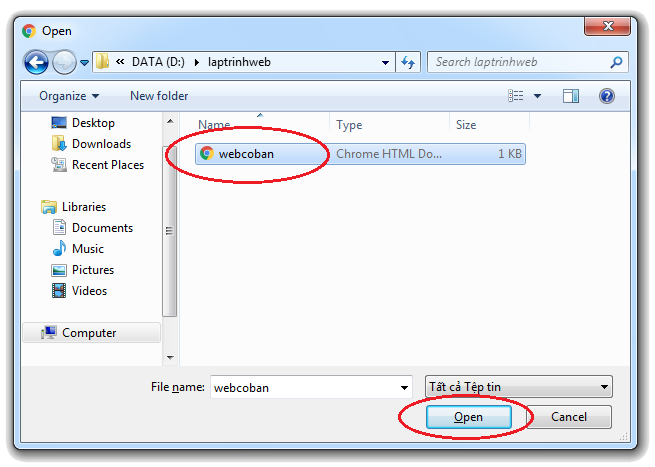
- Thực ra thì cách chạy một tập tin HTML khá là đơn giản, chúng ta có thể di chuyển thẳng đến thư mục chứa tập tin, sau đó nhấp chuột phải vào nó rồi chọn một trình duyệt để mở.



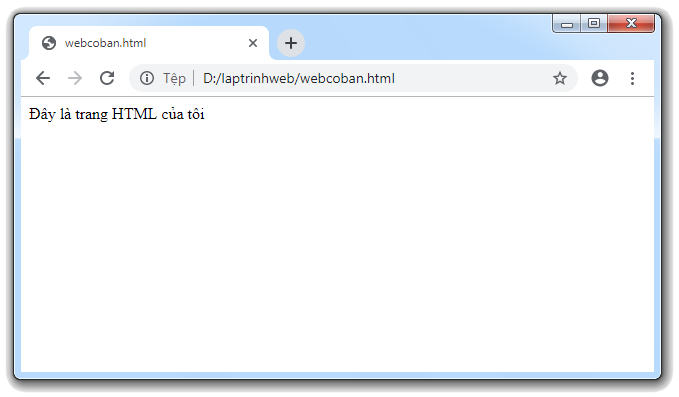
- Hoặc nhấp vào thanh địa chỉ của trình duyệt rồi bấm tổ hợp hợp phím Ctrl + O



- Sau đó, chọn tập tin HTML mà mình muốn mở rồi bấm "Open"



- Nếu trình duyệt hiển thị giống như hình bên dưới thì tức là các bạn đã làm thành công.



#### Cấu trúc của file HTML cơ bản

**Cú pháp và các phần tử HTML cơ bản**

**<!DOCTYPE html>**

<html>

**<head>**

<title>Howkteam.com</title>

**</head>**

**<body>**

**</body>**

</html>

Cấu trúc chung của html gồm có thẻ **<!DOCTYPE html>** dùng để xác định rằng code bạn đang dùng là html

* Cặp thẻ đóng mở **<html> </html>** chứa toàn bộ các dòng lệnh html bên trong (sau này còn là code của Javascript và CSS khi chèn vào)
* Tiếp đến là 2 cặp thẻ **<head></head>**chứa thông tin của webpage và đặc trưng bởi thẻ**<title></title>**, mà mặc định sẽ không được hiển thị trên web
* Thẻ **<body></body>** chứa toàn bộ nội dung html sẽ hiển thị lên trên trang web của chúng ta.

#### Các thẻ thường gặp trong HTML

Chúng ta tiếp tục với các thẻ thường gặp trong html

##### Thẻ tiêu đề

**<h1>** Nội dung tiêu đề **</h1>**

Tương tự với các thẻ **h2**,**h3**,**h4**,**h5,h6** với cấp độ tiêu đề giảm dần

**Ví dụ:**

<**h1**>Howkteam.com</**h1**>

<**h2**>Kteam </**h2**>

<**h3**>Free Education</**h3**>

<**h4**>Share To Be Better</**h4**>

<**h5**>Khóa học lập trình Front End</**h5**>

<**h6**>Project Landing Page</**h6**>

##### Thẻ định dạng đoạn văn bản

**<p>** Nội dung đoạn văn bản chứa bên trong **</p>**

**Ví dụ:**

<**p**> Với mong muốn mang đến kiến thức chất lượng, miễn phí cho mọi người, với tâm huyết phá bỏ rào cản kiến thức từ việc giáo dục thu phí. Chúng tôi - Kteam đã lập nên trang website này để thế giới phẳng hơn.

Bất cứ ai có mong muốn khai phá thế giới. Phá bỏ mọi thứ ngăn cản sự phát triển tất yếu bền vững của xã hội đều là Kter (Thành viên của Kteam).

##### Các thẻ mà tạo ra hiệu ứng với chữ

Ngoài các thẻ trên, HTML còn có các thẻ tạo hiệu ứng với chữ, đoạn văn bản như **<strong></strong>**, **<em></em>**, **<b></b>**, **<u> </u>**, **<strike></strike**>,….

Và một số thẻ khác như **<br>** để cách xuống dòng, **<hr>** để kẻ đường ngăn cách ngang,…

Cùng tham khảo ví dụ bên dưới để hiểu hơn về cách sử dụng các thẻ trên

**Ví dụ:**

<**h1**>HOWKTEAM BLOG</**h1**>

<**h2**>HOWKTEAM</**h2**>

<!--heading tieu de cua cac phan trong website-->

<**p**>

<**strike**>Hello Kter</**strike**> <!--tao chu co gach ngang-->

<**br**> <!--break co tac dung giup xuong dong-->

<**hr**>

<**b**>HOWKTEAM</**b**> <!--bold tao chu in dam-->

<**em**>HOWKTEAM</**em**> <!-- Emphasize tao chu in nghieng-->

<**u**>HOWKTEAM</**u**> <!--underline tao chu co gach chan-->

Lorem, ipsum dolor sit amet consectetur adipisicing elit.

Rem debitis eos dolorem itaque tempore libero ipsa quibusdam fuga, vel,

molestiae maxime harum consequatur architecto repellat dolores cupiditate iure nisi fugiat?

Lorem ipsum, dolor sit amet consect <**br**>etur adipisicing elit. Nulla, culpa.

</**p**>

<!-- Noi ban chua van ban-->

##### **Thẻ hình ảnh & liên kết**

Thẻ **<img>** thẻ chứa nội dung là file ảnh có các thuộc tính:

* src=”” (dùng làm đường dẫn/địa chỉ của ảnh )
* alt=”” (văn bản thay thế hiển thị khi ảnh không tải lên được trên trang web )

**<img src**=”đường dẫn hình ảnh” **alt**=”nội dung thay thế”**>**

Thẻ**<a></a>** dẫn đến các trang web khác.

**<a href**=”đường dẫn/ liên kết”**>** nội dung gắn đường dẫn **</a>**

Thẻ **<form></form>** là nơi chứa các thẻ như **input**, **textarea**, **select** để làm mẫu điền các thông tin trong trang web

Thẻ **<input></input>** là chỗ điền thông tin theo dạng chữ, hoặc dạng email, chúng ta sẽ nói kĩ về input ở các bài sau

Thẻ **<textarea></textarea>**điền thông tin dạng văn bản

**Ví dụ:**

<**img** src="https://www.howkteam.vn/Content/areacontent/images/HowKteamLogo.png" alt="PHoto">

<!-- src source alt alternative -->

<**a** href="https://www.howkteam.vn/">Link dan den HOWKTEAM</**a**>

<**form**>

<**input** type="text" placeholder="dien ten cac ban vao day">

<**input** type="submit" value="gui tin nhan">

<**textarea** cols="100" rows="10" placeholder="hello anh em"></**textarea**>

</**form**>

**2.Các các để CSS đế trang wed**

Trước khi tiến hành viết CSS và để CSS có thể thực thi trên website hoặc tài liệu HTML của bạn thì bạn phải tiến hành nhúng CSS vào website. Nói về nhúng CSS vào website thì chúng ta có hai cách là:

* **Inline Styles** – Nhúng trực tiếp vào tài liệu HTML thông qua cặp thẻ <style> </style>.
* **External Styles** – Tạo một tập tin .css riêng và nhúng vào tài liệu HTML thông qua cặp thẻ <link>.

Mỗi cách nhúng đều có ưu và nhược điểm khác nhau, cũng như tùy theo trường hợp mà ta sử dụng nó.

* **Inline Styles**

Thích hợp với việc chèn một vài đoạn CSS ngắn.

Trình duyệt không mất thời gian tải tập tin CSS.

**External Styles**

* Thích hợp với việc chèn nhiều đoạn CSS, dễ quản lý.
* Nhưng trình duyệt sẽ mất thêm thời gian để tải tập tin CSS.

Và trong các bài học CSS trong serie này bạn có thể sử dụng kiểu Inline Styles để nhúng CSS vào website vì sẽ không mất thời gian tạo tập tin, do chúng ta chỉ viết vài đoạn CSS mà thôi. Anyway, bạn có thể sử dụng kiểu bạn thích.

### Cách nhúng CSS với Inline Styles

Để nhúng CSS vào website thông qua kiểu Inline Styles, bạn sẽ khai báo cặp thẻ <style> vào vị trí bất kỳ của website (tốt nhất là bên trong cặp thẻ <head>) như sau.

[html]

<style type="text/css">

</style>

[/html]

Và trong cặp thẻ <style> đó bạn có thể viết CSS vào, ví dụ đơn giản:

### Cách nhúng CSS với External Styles

Khi sử dụng cách này, việc đầu tiên là bạn cần tạo ra một tập tin .css với tên bất kỳ, bạn có thể dùng bất cứ chương trình soạn thảo văn bản nào để tạo. Sau đó dán một đoạn CSS đơn giản vào như thế này:

p {

color: blue;

font-family: Arial;

}

Và cuối cùng là chèn vào tập tin HTML bằng thẻ <link> và thẻ này phải đặt bên trong cặp thẻ <head>. Ví dụ:

[html]<link rel="stylesheet" href="name.css" />[/html]

Trong đó, thuộc tính rel là khai báo loại tập tin nhúng và href là đường dẫn khai báo tên tập tin .css cần nhúng vào.

Đơn giản phải không nào?

#### Nhúng tập tin CSS vào bên trong một tập tin CSS

Chẳng hạn bây giờ bạn có 3 tập tin CSS mà bạn không muốn thêm tất cả tụi nó vào website mà chỉ muốn thêm một tập tin CSS thôi, thì bạn có thể sử dụng cách nhúng các tập tin CSS vào bên trong một tập tin CSS với từ khóa @import, và các từ khóa @import này phải được đặt ở đầu tập tin .css (không bao gồm các đoạn comment).

## 3.Cú pháp Javascript cơ bản

Theo các hiểu thông thường, một chương trình máy tính là một tập danh sách các hướng dẫn (có thể gọi là statements) để máy tính thực thi. Và Javascript cũng vậy, chỉ khác một điều là các tập lệnh này cho  trình duyệt thực hiện thay vì máy tính.

Các câu lệnh Javascript không cần phải đặt trong dấu ngoặc đơn, hay ngoặc nhọn.

Thậm chí, cuối mỗi câu lệnh cũng không cần dấu chấm phẩy.

* var a = 7;
* var b = 5;
* var c = a + b //Không cần dấu chấm phẩy cũng được

### **3.1. Cách đặt tên biến, hàm**

Trong Javascript, định danh (identifiers) được sử dụng để đặt tên biến, từ khóa, hàm và labels.

Việc định danh nói chung là theo quy tắc sau:

* Ký tự đầu tiên của định danh phải là chữ cái ( a-z, A-Z) hoặc ký tự $ hoặc dấu gạch dưới (\_)
* Không được sử dụng chữ số làm ký tự đầu tiên của định danh.

Nói tóm lại, việc đặt tên biến, hàm… phải tuân theo quy tắc trên.

* var person; // Đúng
* var \_person; // Đúng
* var \_\_person; // Đúng
* var person1; // Đúng
* var 10persion; // Sai

**Lưu ý:**Không được sử dụng chữ số làm ký tự đầu tiên khi đặt tên biến, hàm…

Ngoài ra, các định danh có phân biệt chữ hoa với chữ thường nhé.

* // Đây là 2 biến hoàn toàn khác nhau.
* var Name = "Joe";
* var name = "John";

Về quy tắc đặc tên các đinh danh, thì có nhiều trường phái khác nhau. Điển hình có 3 loại:

* Dùng dấu gạch ngang: Ví dụ: first-name
* Dùng dấu gạch dưới. Ví dụ: first\_name
* Và trường phái **camel case**, tức là viết chữ in hoa với mỗi từ đơn. Ví dụ: FirstName

[Camel case](https://en.wikipedia.org/wiki/Camel_case) là trường phái được sử dụng phổ biến nhất, không chỉ Javascript mà cả các ngôn ngữ khác nữa. Và nhớ rằng, trong Javascript thì định danh thường bắt đầu với chữ thường (Ví dụ: firstName)

### **3.2. Cách khai báo biến**

Javascript hỗ trợ 3 cách để khai báo một biến, đó là sử dụng các từ khóa: let, var và const

Trong đó:

* Const: được dùng để khai báo hằng số, giá trị của nó không thay đổi trong suốt chương trình. Nó giống với từ khóa final của Java vậy.
* var: biến được khai báo với từ khóa var sẽ được truy cập ở bất kể đâu trong chương trình.
* let: từ khóa này giúp cho biến chỉ có thể được truy xuất bên trong phạm vi block quanh nó (block được định nghĩa bởi dấu ngoặc nhọn {} ).

### **3. 3. Kiểu dữ liệu của biến**

Khi bạn đọc đến phần khai báo biến, bạn sẽ thấy ngạc nhiên rằng: Không thấy định nghĩa kiểu dữ liệu của biến khi khai báo, vậy sao biết biến đó có kiểu dữ liệu là gì? Về vấn đề này thì Javascript không có chặt chẽ như Java.

Với Javascript, kiểu dữ liệu sẽ được xác định một cách tự động khi chương trình thực thi. Điều này có nghĩa là một biến có thể có nhiều kiểu dữ liệu ở những thời điểm khác nhau.

* var myTest = 123 ; // myTest là một số
* var myTest = "Xin chào các bạn độc giả VNTALKING "; //myTest là một string/text
* var myTest = false; // myTest là một boolean

Theo tiêu chuẩn [ES7](https://en.wikipedia.org/wiki/ECMAScript), có 7 kiểu dữ liệu:

* Kiểu boolean
* Kiểu null
* Kiểu undefined
* Kiểu số (int)
* Kiểu chuỗ**i** (String)
* Kiểu **Symbol** (mới trong ECMAScript 6)
* Kiểu đối tượ**ng** (Object)

### **3.4. Toán tử trong Javascript**

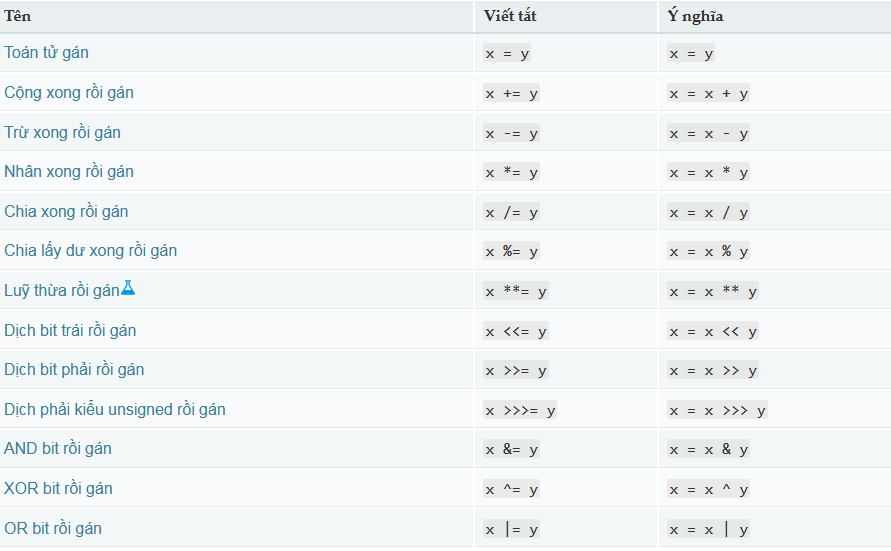
Để định nghĩa toán tử là gì thì thật là khó. Để mình lấy một ví dụ cụ thể cho dễ hiểu nhé:

* // Ta có phép tính như sau:
* const a = 5
* const b = 6
* var c = a + b
* console.log("c = " + c) //màn hình console sẽ in ra: c = 11

Trong ví dụ trên, ta thấy phép tính a + b, thì phép tính cộng ấy được gọi là toán tử (operator).

Javascript hỗ trợ đầy đủ các loại toán tử tính toán và kết hợp.

Dưới đây là bảng tất cả các toán tử được hỗ trợ:

Các loại toán tử trong Javascript

### **3.5. Comment trong Javascript**

Comment chứ không phải “Comment”

Trong bất kể ngôn ngữ thì comment là một phần không thể thiếu. Comment không chỉ giúp cho người đọc mã nguồn nhanh chóng hiểu được ý nghĩa của hàm, biến đó làm gì. Comment còn giúp bạn xuất ra tài liệu hướng dẫn, mô tả về code (thường áp dụng cho các thư viện muốn làm tài liệu mô tả APIs).

Comment không được thực thi bởi trình duyệt. vì vậy, bạn thoải mái mà viết chú thích mà không sợ code bị lỗi hay in ra màn hình.

Javascript hỗ trợ 2 kiểu comment: **Single Line Comments** (comment một dòng) và **Multi-line Comments** (comment nhiều dòng). Bạn có thể xem minh họa bên dưới để hiểu rõ hơn.

* var x = 5; // Khai báo biến x và cung cấp giá trị cho nó là 5
* // Khai báo biến y và cung cấp giá trị cho nó là x+ 2
* var y = x + 2;
* /\* Đây là ví dụ về comment trên nhiều dòng trong JavaScript
* code \*/
* document.getElementById("test\_el").innerHTML = "Some text";
* document.getElementById("test\_el2").innerHTML = "More text";

**Lưu ý:**Dạng comment nhiều dòng (hay còn gọi là block comment) thường được dùng để xuất ra các tài liệu mô tả.

Một vài lời khuyên của mình về comment:

* Nên Comment để giải thích “**Tại sao**” lại viết code đó thay vì **đoạn code làm như thế nào**. Vì người đọc code là đủ hiểu code làm như nào thế nào rồi.
* Không comment những đoạn code mà đọc code đó đã quá rõ ràng rồi.
* Không comment để lấp liếm vấn đề của đoạn code. Tức là người code sau cần phải đọc comment sử dụng đúng đoạn code nếu không sẽ bị lỗi. Thay vì đó, bạn nên viết code lại cho chặt chẽ hơn.
* Xóa ngay comment khi đoạn code đi kèm đã bị xóa để tránh dư thừa comment.

**4.c**

Biểu mẫu HTML được sử dụng để thu thập thông tin đầu vào của người dùng thường được gửi đến máy chủ để xử lý.

Biểu mẫu hay còn gọi là Form trong HTML được tạo ra bằng thẻ <form>, bên trong form đó nó chứa một hoặc nhiều phần tử để nhập liệu gọi là các điều khiển (control), có nhiều loại điều khiển như:

-Điều khiển nhập một dòng text (TextBox)

-Điều khiển cho nhập nhiều dòng text (textarea)

-Điều khiển là các nút bấm (button)

-Các phần tử checkbox (hộp chọn)

-Các phần tử radio (chọn một):

-Danh sách đổ xuống

Hầu hết các control để người dùng nhập thông tin, dữ liệu được tạo ra bằng thẻ HTML <input> và đi cùng nó thường là phần tử <label> để tạo ra nhãn (tiêu đề) cho control.

Để tạo ra HTML Form thì dùng đến thẻ <form>, sau đó nội dung trong thẻ trình bày các HTML và các phần tử là điều khiển (control) có trong form.

<**form** action="http://nhom21.net/" method="post">

<!--Các mã HTML, các phần tử trong form -->

</**form**>

Thẻ <form> cơ bản có hai thuộc tính cần lưu tâm là action và method:

Thuộc tính action trong form: thuộc tính để thiết lập URL sẽ nhận dữ liệu, là địa chỉ mà dữ liệu của form gửi đến (submit đến, post đến). Thiếu tham số này thì action bằng URL đang truy cập (tức gửi thông tin form đến server theo địa chỉ đang truy cập). Web server nhận được dữ liệu, xử lý và trả về nội dung.

Thuộc tính method trong form: thuộc tính để thiết lập HTTP Method, xem thêm HTTP Request Message thường có giá trị bằng get hoặc post. Nếu không viết thuộc tính này thì method mặc định của form là get

Lưu ý : Sử dụng method="get" thì khi submit dữ liệu được biểu diễn (encoding) thông qua URL (người dùng thấy rõ dữ liệu trên thanh địa chỉ trình duyệt khi nó gửi đi). Sử dụng method="post" thì khi submit biểu diễn trong nội dung của Http Request gửi đến Server và là ẩn với người dùng. Sử dụng post an toàn hơn.

Cách tạo phần tử nhập dữ liệu :

-Ô văn bản : <input type="text"> thẻ input xác định trường nhập một dòng để nhập văn bản

Ví dụ : <form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname">  
</form>

Sau khi nhập xong dữ liệu chúng ta sẽ được kết quả như sau :

A white rectangle with black lines

Description automatically generated

-Ô chọn : <input type="checkbox"> dòng lệnh xác định ô chọn

Các ô chọn cho phép người dùng lựa chọn các tùy chọn theo ý mình có thể không chọn và chọn nhiều lựa chọn cùng lúc có giới hạn.

Ví dụ :

<**form** action="https://httpbin.org/anything" method="post">

<**input** type="checkbox" name="love1" id="love1" value="LovePHP">

<**label** for="love1">Tôi thích PHP</**label**><**br**>

<**input** type="checkbox" name="love1" id="love2" value="LoveHTML">

<**label** for="love2">Tôi thích HTML</**label**><**br**>

<**input** type="checkbox" name="love1" id="love3" value="LoveCSS">

<**label** for="love3">Tôi thích CSS</**label**><**br**>

<**button** type="submit">Gửi</**button**>

</**form**>

Kết quả :

A white rectangular object with a black border

Description automatically generated

-Nút gửi : <input type="submit"> dòng lệnh xác định nút bấm để gửi dữ liệu biểu mẫu cho trình xử lý biểu mẫu.

Trình xử lý form là một tệp trên máy chủ có tập lệnh để xử lý đầu vào.

Trình xử lý form mang thuộc tính hành động <action> của form.

Ví dụ về tạo nút gửi trong form :

<form action="/action\_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>

Kết quả :

A screenshot of a login form

Description automatically generated

**5.Quy trình phát triển trang wed:**

Để phát triển một trang wed ta cần phải xác định được các yêu cầu :

1. Wed làm về để tài gì
2. Cần những thông tin gì để làm về đề tài đó
3. Trang wed làm theo form giao diện ,màu sắc như thế nào
4. Cần hiển thị những thông tin gì trên trang wed

**Phân tích các bước quan trọng làm một trang wed**

### **Mục đích thiết kế website là gì?**

Xác định mục đích thiết kế website giúp doanh nghiệp định hướng loại hình trang web ngay từ khi bắt đầu thiết kế. Điều này rất quan trọng bởi trang web bán hàng có yêu cầu khác với trang web giới thiệu hay tin tức, truyền thông. Chỉ khi xác định được mục tiêu chính xác, doanh nghiệp mới có thể xây dựng chiến lược phát triển đúng đắn cho website.

### **Nội dung website**

Nội dung website ảnh hưởng đến cả trải nghiệm khách hàng lẫn thứ hạng Seo của web. Chính vì vậy mà nó trở thành yếu tố quan trọng không thể bỏ qua khi phân tích website. Phân tích website sẽ chỉ ra chất lượng của nội dung các bài đăng, những lỗi thường gặp, đâu là những bài đăng thu hút traffic, tăng tỉ lệ chuyển đổi,...

Dựa trên kết quả phân tích, doanh nghiệp sẽ quyết định xem nên tập chung content theo hướng nào, bỏ hay sửa các nội dung nào để đáp ứng nhu cầu của người truy cập.

**Thiết kế giao diện**

### **Đơn giản hóa các tác vụ người dùng**

Mọi người thường nghĩ một website với nhiều hiệu ứng chuyển cảnh, nhiều tác vụ đa dạng sẽ làm khách hàng ấn tượng bởi sự đầu tư.

Nhưng đây là một nhận định sai lầm. Ngay cả chính chúng ta cũng không muốn dùng những sản phẩm, thiết bị có quá nhiều nút bấm, thao tác. Vì vậy, nhà thiết kế cần đơn giản hóa các tác vụ website để nâng cao trải nghiệm người dùng.

### **Mọi thứ cần thực hiện rõ ràng và đầu tư**

Một website với những tính năng rõ ràng, dễ dàng sử dụng nhận được niềm yêu thích của khách hàng. Không ai có đủ thời gian và kiên nhẫn để tìm hiểu cách dùng một website.

Người thiết kế giao diện website cần ghim những chức năng lên thanh menu chính để dù ở bất kỳ nơi nào, khách hàng của bạn vẫn có thể truy cập một cách dễ dàng.

### **Sắp xếp bố cục hài hòa, hợp lý**

Khi thiết kế giao diện người dùng website, cần hình dung quá trình mua hàng của khách hàng để sắp xếp bố cục hài hòa. Một website với những tính năng được sắp xếp khoa học, theo đúng trình tự sẽ tăng tỷ lệ chuyển đổi cho doanh nghiệp.

**Kiểm thử**

Kiểm thử web là tên gọi được đặt cho một quá trình kiểm thử phần mềm chủ yếu vào việc kiểm tra các vận dụng web. Ứng dụng web cần được kiểm tra hoàn toàn trước khi đi vào hoạt động, điều này có thể giúp giải quyết các vấn đề trong vận dụng website trước khi tiếp xúc với người dùng như các vấn đề về tính năng, bảo mật, các vấn đề dịch vụ web, các vấn đề tích hợp và khả năng xử lý lưu lượng truy cập, trong quá trình kiểm thử website, cần cố gắng phát hiện ra lỗi có thể xảy ra trong nền tảng nhằm giải quyết kịp thời.

**Vai trò của thành viên trong nhóm**

Chịu trách nghiệm chính :

**Nguyễn Việt Hoàng**:Lên ý tưởng triên khai thiết kết ,làm về giao diện các thông tin hiện thị trên trang wed,làm báo cáo,sửa lỗi quá trình làm của bản thân và các thành viên trong nhóm

**Nguyễn Quang Hải**:Làm về phần tìm kiếm hình ảnh ,báo cáo,các thành phần giao diện ở phần đầu trang

**Phạm Đình Trường Sơn**: làm về phần tìm kiếm hính ảnh và các phần giao diện ở phần đầu trang