



GIT – GITHUB CHO LẬP TRÌNH VIÊN THIẾT KẾ WEBSITE

Tùng Tony



GitHub

TUVO
PISCES KIBO

MỤC LỤC

Bài 1: GIỚI THIỆU VỀ GIT VÀ GITHUB.....	2
Bài 2: TẠO TRANG CHỦ VÀ UPLOAD MIỀN.....	3
Bài 3: SỬA FILE TRÊN GITHUB	5
Bài 4: TẠO TRANG WEB GALLERY	7
Bài 5: TỔNG QUAN VỀ CÁC THẺ HTML.....	9
Bài 6: TỔNG QUAN VỀ CSS	15
Bài 7: ĐỆM, VIỀN VÀ KHOẢNG LỀ	24

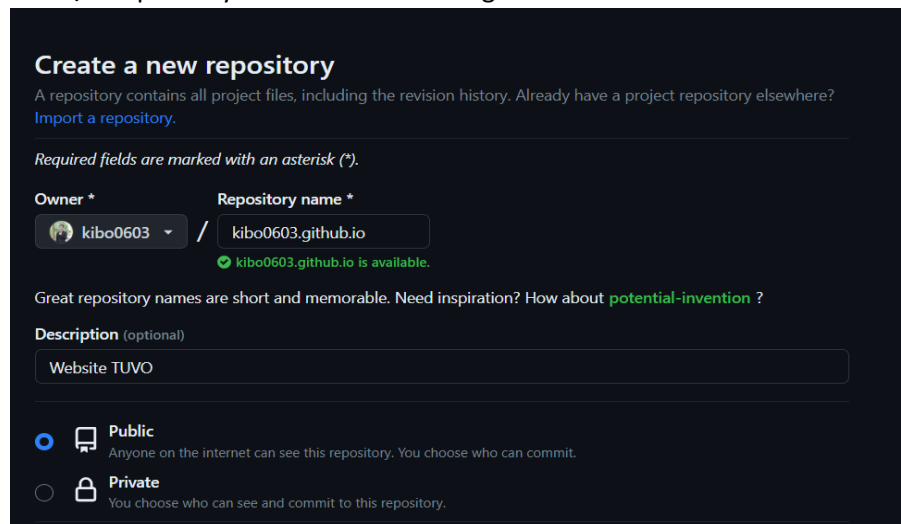
Bài 1: GIỚI THIỆU VỀ GIT VÀ GITHUB

1. Khái niệm:

- GIT:
 - Là một phần mềm cài trên máy tính giúp cho ta có thể quản lý file
 - Có thể biết khi nào một file nào đó trong folder bị thay đổi bằng lệnh GIT status
- GITHUB:
 - Là một website cho phép đăng ký free (giống google drive)
 - Mỗi tài khoản đăng ký có thể lưu được không giới hạn file
 - Cho phép hosting website (không tính phí)
 - Không phải lưu thể nào cũng được, lưu qua phần mềm GIT

2. Tạo tên miền và kho lưu trữ cho Project: (Tạo website)

- B1: Tạo Repository name: OwnerName.github.io → Tên miền




Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * **Repository name ***


 kibo0603 / kibo0603.github.io


✔ kibo0603.github.io is available.

Great repository names are short and memorable. Need inspiration? How about **potential-invention** ?

Description (optional)

Website TUVO

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

- B2: Truy cập vào trang web google gõ: OwnerName.github.io
(vd: kibo0603.github.io)
- B3: Lưu lại dòng: (tên của link dẫn đến web)

```
git remote add origin https://github.com/kibo0603/kibo0603.github.io.git
```

Bài 2: TẠO TRANG CHỦ VÀ UPLOAD MIỀN

1. Cách DEMO trang web:

- Tên miền đã tạo: kibo0603.github.io
- B1: Tạo folder mới cho project WEB
- B2: Kéo folder vào Sublime Text
- B3: Tạo file **index.html** (file trang chủ website) và file CSS (**colorweb.css**)
- B4: Vào file **index.html** gõ html rồi nhấn phímTAB

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> TIÊU ĐỀ HTML </title>
  <link rel="stylesheet" type="text/css" href="colorwebFile.css">
</head>
<body>
  Thực hiện các câu lệnh và phương thức
</body>
</html>
```

- Trong thẻ <body> Nội dung </body>:
 - Thẻ <h1> **Tiêu đề** </h1>
 - Thẻ <div class = "**nameDiv**"> Nội dung của nameDiv </div>
 - Thẻ **Phần in đậm** <i> Phần in nghiêng </i>
 - Thẻ linklienket
 - Thẻ <hr> là dòng kẻ ngang ngăn cách
- B5: Vào file CSS và thực hiện các thao tác: (định dạng phông chữ và vị trí chung cho từng loại thẻ)

```
body {
  font-family: "segoe ui";
}

h1 {
  text-align: center;
}

img {
  width: 20%;
}
```

- B6: Click chuột vào thư mục Project và nhấp Git Bash Here

- B7: Khởi tạo repo trống trên bộ nhớ Stack trên máy: **git init**
- B8: Đẩy file lên repo Stack: **git add "namefile.html"** và **git add "namefile.css"**
(hoặc dùng lệnh **add** để add tất cả các file)
- B9: Kết nối repo trên Github web:
 - **git remote add origin <https://github.com/kibo0603/kibo0603.github.io.git>**
- B10: Kiểm tra trạng thái: **git status**
- B11: Ký tên commit: **git commit -m "Message: upload 2 file trang chu cua website"**
- B12: Đẩy repo máy lên Github web: **git push origin master**
- B13: Có thể truy cập đường link kibo0603.github.io

2. Thực hành DEMO trên GIT:

DEMO GIT

```
Admin@PiscesKibo MINGW64 /d/WEB GITHUB
$ git init
Initialized empty Git repository in D:/WEB GITHUB/.git/

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git add "index.html"

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git add "colorweb.css"

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git remote add origin https://github.com/kibo0603/kibo0603.github.io.git

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   colorweb.css
    new file:   index.html
Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git commit -m "upload 2 file trang chu cua website"
[master (root-commit) 55cf1f8] upload 2 file trang chu cua website
 2 files changed, 36 insertions(+)
 create mode 100644 colorweb.css
 create mode 100644 index.html

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 720 bytes | 720.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kibo0603/kibo0603.github.io.git
 * [new branch]      master -> master
```

Bài 3: SỬA FILE TRÊN GITHUB

1. Câu lệnh dùng SYNC với tài khoản WEB:

- B1: Sửa file cũ thì cần phải kiểm tra lại trạng thái: **git status**
- B2: add lại file sửa trên GIT: **git add** . (add tất cả file lên GIT)
- B3: Xác nhận đã chỉnh sửa: **git commit -m "Comments: đã chỉnh sửa"**
- B4: Đẩy repo máy lên Github web: **git push origin master**
- B5: Refresh lại trang web

2. Minh họa thao tác trên GIT:

DEMO GIT

```
Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git add index.html

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git commit -m "Đổi tiêu đề"
[master d3978ff] Đổi tiêu đề
1 file changed, 2 insertions(+), 2 deletions(-)

Admin@PiscesKibo MINGW64 /d/WEB GITHUB (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 367 bytes | 367.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/kibo0603/kibo0603.github.io.git
e7862f4..d3978ff  master -> master
```

3. Thêm file ảnh trong GIT:

- B1: Tạo thư mục Image chứa ảnh trong folder Project web
- B2: Tạo thẻ ảnh
- B3: Thực hiện lại các thao tác trên để SYNC và load lại trang web

4. Lấy lại File từ Github về máy: (Pull Request)

- B1: Tạo folder mới

- B2: Mở GIT Bash Here nhập: ***git init***
- B3: Kết nối repo trên Github:
 - ***git remote add origin*** <https://github.com/kibo0603/kibo0603.github.io.git>
- B4: Dùng lệnh để kéo dữ liệu GIT về folder mới đó: ***git pull origin master***

Bài 4: TẠO TRANG WEB GALLERY

1. Tạo trang web khác ngoài trang chủ:

- B1: Tạo menu trong HTML
 - B2: Tạo file mới trong ổ của mình
 - B3: SYNC kết nối dữ liệu lại với nhau
- => Mục đích chuyển đổi qua lại giữa hai trang web

2. DEMO chương trình:

<u>index.html</u>	<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title> CV cá nhân của Tạ Quang Tùng </title> <link rel="stylesheet" type="text/css" href="colorweb.css"> </head> <body> Trang chủ HOME
 Xem gallery <h1> Tạ Quang Tùng (Python FullStack) </h1> <h1> </h1> <hr> <div class="thongtin"> Tên: <i> Tạ Quang Tùng </i> </div> <div class="thongtin"> Năm sinh: <i> 06/03/2003 </i> </div> <div class="thongtin"> SĐT: <i> 039846203 </i> </div> <div class="thongtin"> Chuyên ngành: <i> Toán Tin </i> </div> <div class="thongtin"></pre>
-------------------	--

	<pre> Facebook: <i> https://www.facebook.com/pisceskibo </i> </div> </body> </html> </pre>
<u>Gallery.html</u>	<pre> <!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title> Gallery Website </title> </head> <body> <div class="Menu"> Quay lại home
 Xem gallery </div> <h2> Đây là trang Gallery trưng bày sản phẩm </h2> </body> </html> </pre>
<u>Colorweb.css</u>	<pre> body { font-family: "segoe ui"; } h1 { text-align: center; } img { width: 20%; } </pre>

Bài 5: TỔNG QUAN VỀ CÁC THẺ HTML

1. Khái niệm HTML và CSS:

- HTML (Hypertext Markup Language) giúp tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, ... → Khung xương
- CSS (Cascading Style Sheet Language) tạo phong cách cho trang web, định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu HTML → trang trí

2. Các thẻ trong HTML:

TẤT CẢ CÁC THẺ TRONG HTML

Từ khóa	Ý nghĩa
<!DOCTYPE>	Xác định cho trình duyệt biết phiên bản HTML đang sử dụng
<html>	Xác định gốc của một tài liệu HTML (chứa tất cả nội dung website)
<head>	Xác định phần đầu của tài liệu HTML (chứa các thẻ cung cấp thông tin cho trang web)
<title>	Xác định tiêu đề của trang web
<body>	Xác định phần thân của tài liệu HTML
<div>	Chứa các paragraph, images, blockquote, ...
<h1> <h6>	Tạo những đề mục quan trọng trong trang web
<p>	Xác định một đoạn văn bản
 	Chèn ngắt dòng xuống
<hr>	Tạo một đường kẻ ngang phân cách
<!-->	Xác định một đoạn chú thích
<abbr>	Định nghĩa một từ viết tắt
<address>	Xác định thông tin liên hệ của tác giả (chủ sở hữu) trang web
<bdo>	Điều hướng một đoạn văn bản được chỉ định
	Xác định đoạn văn bản in đậm
<blockquote>	Xác định một “đoạn trích dẫn” từ một website khác
<code>	Xác định một đoạn văn bản mang ý nghĩa khác là các mã lệnh
	Xác định văn bản đã bị xóa từ một tài liệu
	Định nghĩa nhấn mạnh văn bản
<big>	Xác định một đoạn văn bản có kích thước chữ to hơn văn bản bình thường
<i>	Xác định một đoạn văn bản in nghiêng
<ins>	Định nghĩa một đoạn văn bản đã được chèn thêm thay thế vào tài liệu
<kbd>	Nhấn mạnh, định nghĩa đầu vào bàn phím
<mark>	Đánh dấu, làm nổi bật cho văn bản
<meter>	Định nghĩa một thước đo lường với một khoảng đã biết
<progress>	Tạo một thanh tiến trình
<pre>	Định nghĩa văn bản định dạng sẵn

<q>	Xác định một câu trích dẫn ngắn
<s>	Định nghĩa văn bản văn bản đó không còn đúng nữa
<small>	Xác định một đoạn văn bản có kích thước chữ nhỏ hơn văn bản bình thường
	Xác định đoạn văn bản được in đậm
<sub>	Định nghĩa văn bản giống như chỉ số dưới
<sup>	Hiển thị văn bản giống số mũ
<time>	Đánh dấu những phần văn bản là: thời gian, ngày tháng, ...
<u>	Tạo đường gạch chân lên văn bản
<wbr>	Ngắt bớt ký tự của từ xuống dòng
<frame>	Xác định một khung trong một bộ khung
<frameset>	Xác định một bộ khung
<noframe>	Xác định một nội dung được hiển thị khi trình duyệt không hỗ trợ thẻ <frame>
<iframe>	Nhúng một trang web khác vào trang web hiện tại
	Chèn hình ảnh vào trang web
<map> & <area>	Tạo một bản đồ ảnh
<figcaption>	Tạo một tiêu đề cho nội dung được đặt bên trong phần tử <figure>
<figure>	Xác định một nội dung cần được tách biệt rõ ràng

3. Comment trong HTML:

- `<!-- Chú thích đoạn code -->` hoặc nhấn tổ hợp phím **Ctrl + /**

4. Thẻ thông dụng trong HTML:

- <h1> <h6>: (heading): đề mục quan trọng cho văn bản (từ lớn đến bé)

```
<h1> Đây là tiêu đề lớn thẻ h1 </h1>
<h2> Đây là tiêu đề lớn thẻ h2 </h2>
<h3> Đây là tiêu đề lớn thẻ h3 </h3>
<h4> Đây là tiêu đề lớn thẻ h4 </h4>
<h5> Đây là tiêu đề lớn thẻ h5 </h5>
<h6> Đây là tiêu đề lớn thẻ h6 </h6>
```

- <p>: (paragraph): văn bản, đoạn văn

```
<p> Đoạn văn bản được viết vào đây </p>
```

- : (image): hiển thị hình ảnh

```

```

- <iframe>: hiển thị video

```
<iframe width="width" height="height" src="linkVideo" frameborder="0"
allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;
picture-in-picture; web-share" allowfullscreen> </iframe>
```

- <i>: thẻ in nghiêng chữ

```
<i> In nghiêng chữ </i>
```

- hoặc : thẻ in đậm chữ, nhấn mạnh chữ

 In đậm chữ
 In đậm chữ mạnh

- <u>: thẻ gạch dưới chữ

<u> Gạch dưới chữ </u>

- a: (anchor): truy cập đến link được gán

 Truy cập đến link được gán
--

-
: tự động ngắt dòng xuống

- : đánh dấu một phần văn bản

- Công thức:

element1element2element3

- Ví dụ:

<pre>4Fe + 30<sub>2</sub> = 2Fe<sub>2</sub>0<sub>3</sub></pre>
4Fe + 3O ₂ = 2Fe ₂ O ₃

- <mark>: thẻ highlight chữ:

<mark> Thẻ highlight chữ </mark>

- ul, li: (unordered list, list item) hiển thị danh sách

<pre> Item 1 Item 2 Item 3 </pre>	<pre> Item 1 Item 2 Item 3 </pre>
---	---

(có thể thay ul thành ol: ul là dấu chấm tròn, ol là đánh số)

- <sup> và <sub>: Thẻ cơ số và số mũ trong toán học

- Công thức:

cơ số^{số mũ}
số_{cơ số phía dưới}

- Ví dụ:

<!-- f(x) = x^2 + 3x - 1 -->

```
f(x) = x<sup>2</sup> + 3x - 1
```

```
<!-- số<sub>ơ số</sub> => Số mũ ở dưới -->
```

```
0101<sub>10</sub>
```

In nghiêng chữ **In đậm chữ** Gạch dưới chữ **In đậm chữ mạnh**

$f(x) = x^2 + 3x - 1$

0101₁₀

Thẻ highlight chữ

- table: tạo bảng dữ liệu

```
<table>
  <!-- Phần đầu của bảng -->
  <thead>
    <th> Table Heading 1 </th>
    <th> Table Heading 2 </th>
    <th> Table Heading 3 </th>
  </thead>

  <!-- Phần thân của bảng -->
  <tbody>
    <!-- Table row: từng dòng -->

    # Dòng 1
    <tr>
      <td> Table Body 1 </td>
      <td> Table Body 2 </td>
      <td> Table Body 3 </td>
    </tr>

    # Dòng 2
    <tr>
      <td> Row 2 Col 1 </td>
      <td> Row 2 Col 2 </td>
      <td> Row 3 Col 3 </td>
    </tr>

  </tbody>
</table>
```

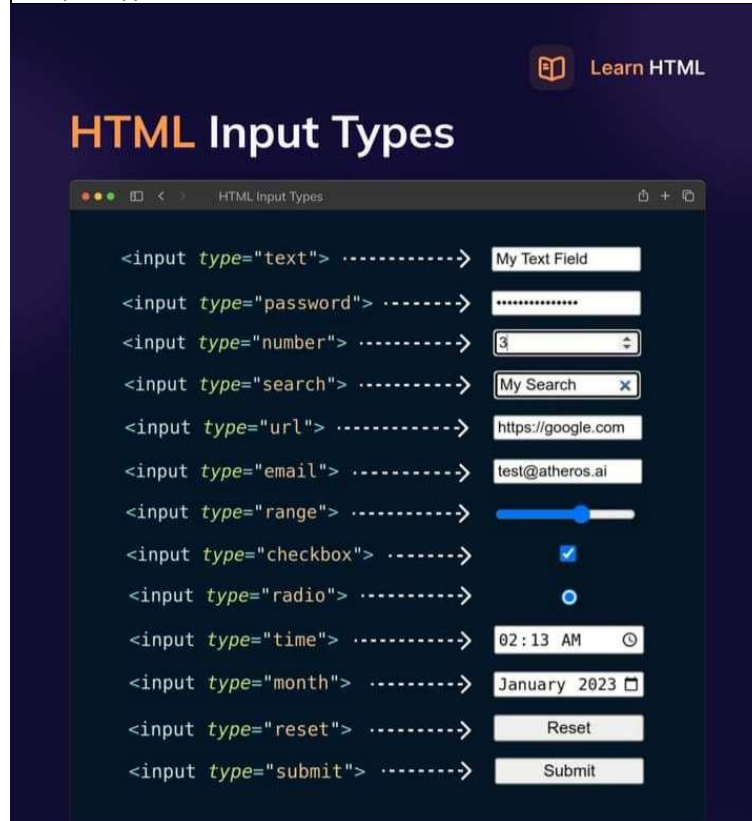
- input: cho phép nhập liệu dữ liệu đầu vào

```
<!-- Ô cho phép nhập liệu
      + type có thể là text, checkbox, radio,...
      + name là class chính của ND đó
-->
<input type="text" name="Ô nhập text">
```

```

<input type="password" name="Nhập mật khẩu ẩn ">
<input type="number" name="Nhập số">
<input type="search" name="Tìm kiếm">
<input type="url" name="Nhập link liên kết">
<input type="email" name="Nhập email">
<input type="range" name="Tạo thanh kéo âm lượng">
<input type="checkbox" name="Ô chọn Checkbox - vuông">
<input type="radio" name="Ô chọn duy nhất Radio - tròn">
<input type="time" name="Nhập ngày giờ">
<input type="month" name="Nhập tháng năm">
<input type="reset" name="Tạo nút reset">
<input type="submit" name="Tạo nút submit">

```



- button: nút bấm click

```
<button> Nút bấm click </button>
```

- div:

```

<!-- Tạo khối bao quanh phần tử khác -->
<div>
    <!-- Nội dung cần phân khối -->
</div>

```

- Attribute: (nằm trong thẻ mở)
→ Ta có thể thêm thuộc tính Attribute của thẻ HTML cho các thẻ (di chuột vào sẽ hiển thị nhỏ nhỏ góc dưới)

```
<thẻ attribute="Nội dung của tiêu đề Attribute"> Attribute </thẻ>
```

→ attribute có thể là title, onclick, ...

- Lorem + TAB: tạo văn bản bất kỳ

5. Thuộc tính target:

- _self – Mặc định → Mở tài liệu trong cùng tab khi được bấm
- _blank → Mở tài liệu trong cửa sổ tab mới
- _top → Mở tài liệu trong toàn bộ cửa sổ

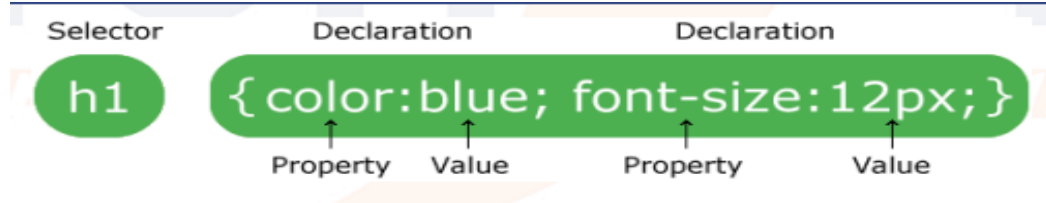
6. Kỹ thuật bố trí Layout:

- <header> – Định nghĩa tiêu đề cho tài liệu hoặc phần
- <nav> - Định nghĩa một tập hợp các liên kết điều hướng
- <section> - Định nghĩa một phần trong tài liệu
- <article> - Định nghĩa nội dung độc lập, độc lập
- <aside> - Định nghĩa nội dung ngoài nội dung (như thanh bên)
- <footer> - Định nghĩa chân trang cho tài liệu hoặc phần
- <details> - Định nghĩa các chi tiết bổ sung mà người dùng có thể mở và đóng theo yêu cầu
- <summary> - Định nghĩa tiêu đề cho phần tử <details>



Bài 6: TỔNG QUAN VỀ CSS

1. Cú pháp câu lệnh:



2. Chú thích trong CSS:

- `/* Chú thích đoạn code */` hoặc nhấn tổ hợp phím Ctrl + /

3. Sử dụng CSS trong HTML:

- Internal: viết ngay CSS ngay bên trong file HTML (sử dụng thẻ `<style>` `</style>` ở phần head)

```
index.html indexcss.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6
7   <!-- Viết CSS vào đây với Internal -->
8   <style>
9     h1 {
10       color: green;
11       font-size: 30px;
12     }
13   </style>
14
15   <title> Sử dụng CSS trong HTML </title>
16 </head>
17 <body>
18   <div>
19     <h1> CSS in HTML </h1>
20     <ol>
21       <li> Internal </li>
22       <li> External </li>
23       <li> Inline </li>
24     </ol>
25   </div>
26
27 </body>
28 </html>
```

- External: viết file CSS riêng ở bên ngoài HTML bằng cách liên kết hai file với nhau

<pre><head> <!-- Viết file CSS vào đây với External --> <link rel="stylesheet" type="text/css" href="color.css"> </head></pre>	<pre>h1 { color: blue; font-size: 20px; }</pre>
--	---

- Inline: sử dụng thẻ mở Attribute Style để viết luôn vào

```
<h3 style="thuộc tính 1: action; thuộc tính 2: action;"> CSS với Inline </h3>
```


4. ID và Class: (CSS selectors)

- CSS selectors là các cách gọi tới các elements trong HTML
 - ID là tên của elements thẻ được chỉ định duy nhất (chỉ thay đổi CSS đối với ID)

<code><h4 id = "nameID"> CSS in HTML </h4></code> → Chỉ đổi màu với ID "first"	<code>#nameID { color: yellow; }</code>
---	---

- Class được sử dụng cho nhiều thuộc tính chung và chức năng giống ID

<code><h4 class = "nameClass"> Class & ID </h4></code> <code><h5 class = "nameClass"> Class & ID </h5></code> <code><h5 class = "nameClass2"> Class & ID </h5></code>	<code>.nameClass { color: gray; }</code>
---	--

- Tính kế thừa trong CSS:
 - Truy xuất đến class chung và class riêng:

<u>HTML file</u>	<code><div class="nameClass nameClass1"> content </div></code> <code><div class="nameClass nameClass2"> content </div></code> <code><div class="nameClass nameClass3"> content </div></code>
<u>CSS file</u>	<code>/* Class chung */</code> <code>.nameClass {</code> Những thay đổi chung cho các elements <code>}</code> <code>.nameClass1 {</code> Những thay đổi riêng cho nameClass1 <code>}</code> <code>.nameClass2 {</code> Những thay đổi riêng cho nameClass2 <code>}</code>

5. CSS selectors cơ bản:

Selector	Ví dụ	Mô tả
.class	.intro	Chọn tất cả các thẻ có class="intro"
.class1.class2	.name1.name2	Chọn tất cả các thẻ có cả name1 và name2 được đặt trong thuộc tính class của nó
.class1 .class2	.name1 .name2	Chọn tất cả các thẻ có class="name2" là con của một phần tử có class="name1"
*	*	Chọn tất cả các thẻ
element	h2	Chọn tất cả các thẻ h2
element.class	div.box	Chọn tất cả thẻ div có class="box"
element, element	div, h2	Chọn tất cả thẻ div và h2

Selector	Ví dụ	Mô tả
element element	div p	Chọn tất cả thẻ p trong thẻ div
element > element	div > p	Chọn tất cả thẻ p là con trực tiếp của thẻ div
element + element	div + p	Chọn thẻ p đứng liền kề sau thẻ div
element ~ element	div ~ p	Chọn tất cả thẻ p đứng sau thẻ div

➤ Ví dụ:

```
<p class="class-1">Đoạn văn 1.1</p>
<p class="class-1">Đoạn văn 1.2</p>

<p class="class-2 class-3">Đoạn văn 2.1</p>
<p class="class-2 class-3">Đoạn văn 2.2</p>

<div class="class-4">
  <p class="class-5">Đoạn văn 3</p>
</div>

<p class="class-6">Đoạn văn 4 - thẻ p</p>
<div class="class-6">Nội dung 5 - thẻ div</div>

<p class="class-7">Đoạn văn 6</p>
<p class="class-8">Đoạn văn 7</p>

<h3>Tiêu đề 8</h3>
<h4>Tiêu đề 9</h4>

<ul class="list">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3
    <ul>
      <li>Item 1 (con Item 3)</li>
      <li>Item 2 (con Item 3)</li>
    </ul>
  </li>
</ul>
```

```
<p class="class-9">Đoạn văn 10</p>
<p>Đoạn văn 11</p>
<p>Đoạn văn 12</p>

<p class="class-10">Đoạn văn 13</p>
<p>Đoạn văn 14</p>
<p>Đoạn văn 15</p>

<div>Vật cản</div>

<p>Đoạn văn 16</p>
<p>Đoạn văn 17</p>
<p>Đoạn văn 18</p>
```

→ HTML

```
/* Lựa chọn toàn bộ các thẻ */
* {
    /* `font-style: italic` là chữ nghiêng */
    font-style: italic;

    /* text-align: căn vị trí của chữ */
    text-align: center;
}

/* Lựa chọn toàn bộ các thẻ p */
p {
    /* `text-decoration: underline` là chữ gạch chân */
    text-decoration: underline;
}

/* Lựa chọn toàn bộ các thẻ có class-1 */
.class-1 {
    color: red;
}

/* Lựa chọn toàn bộ thẻ đồng thời có cả class-2 và class-3 */
.class-2.class-3 {
    color: green;
}
```

```
/* Lựa chọn thẻ có class-5 là con của thẻ class-4 */
.class-4 .class-5 {
    color: violet;
}
→ Class chung là .class-4
→ Class riêng là .class-5

/* Lựa chọn toàn bộ các thẻ p có class-6 */
p.class-6 {
    color: blue;
}

/* Lựa chọn toàn bộ các thẻ h3, h4 */
h3, h4 {
    color: brown;
}

/* Lựa chọn toàn bộ thẻ li là con của class list */
.list li {
    color: deepskyblue;
}

/* Lựa chọn toàn bộ thẻ li là con trực tiếp của thẻ có class list */
.list > li {
    color: fuchsia;
}

/* Lựa chọn 1 thẻ p đứng liền kề sau class-9 */
.class-9 + p {
    color: goldenrod;
}

/* Lựa chọn toàn bộ thẻ p đứng sau class-10 */
.class-10 ~ p {
    color: limegreen;
}
```

→ CSS

6. Độ ưu tiên trong CSS:

- a) Internal, External: giống nhau, cái nào gọi sau thì ưu tiên cái sau
 - a. Inline – 1000
 - b. #id – 100
 - c. .class – 10
 - d. tag - 1
- b) Equal specificity:
 - Cái nào gọi sau thì ưu tiên cái sau
 - Sẽ cộng dồn điểm cái nào cao hơn thì ưu tiên hơn
- c) Universal selector and inherited: ít được ưu tiên nhất
 - Nếu trong CSS thuộc tính nào có !important sẽ được ưu tiên (1000+)

```
tag {  
    color: <màu sắc> !important;  
}
```

7. Sử dụng biến trong CSS:

- Đặt tên biến CSS trong file CSS có sử dụng mã giả:

```
:root {  
    /* Đặt biến toàn cục Global trong :root */  
    --name-variable: value;  
}  
  
/* Lấy từ biến cục bộ */  
Thẻ1 {  
    /* Đặt biến cục bộ Local trong thẻ */  
    --name-variable: value;  
    color: var(--name-variable);  
}  
  
/* Lấy từ biến toàn cục */  
Thẻ2 {  
    color: var(--name-variable);  
}
```

8. Các đơn vị trong CSS:

- Absolute units (đơn vị tuyệt đối):
 - **px: kích thước sẽ cố định (100% = 16px)**
 - pt: points (1pt = 1/72 of 1in)
 - cm: centimeters
 - mm: millimeters
 - inch: 1in = 96px = 2.54cm
 - pc: picas (1pc = 12 pt)
- Relative units (đơn vị tương đối):
 - **%: bằng số phân trăm kích thước thẻ <body> của trình duyệt**

- **rem**: phụ thuộc vào định nghĩa thẻ `<html>`
 - VD: `<html>` có `font-size: 40px` thì `1rem = 40px`
- **em**: phụ thuộc vào thẻ gần nhất chứa nó (cái nào có thuộc tính cùng gần hơn thì lấy theo)
- **vw (viewport width)**: khung chiều ngang ảnh
- **vh (viewport height)**: khung chiều dọc ảnh
- **vmin**: Viewport Minimum (vmin) - Đơn vị này dựa trên kích thước nhỏ hơn của khung nhìn
- **vmax**: Viewport Maximum (vmax) - Đơn vị này dựa trên kích thước lớn hơn của khung nhìn
- Reset margin và padding bằng cách cho **`margin: auto; padding: 0rem`**

9. Một số hàm trong CSS:

- `var(--name-variable)`: hàm lấy giá trị thuộc tính từ biến cục bộ hoặc toàn cục
- `linear-gradient()`
- `rgba(red, green, blue, alpha)`: hàm tạo màu tùy thích
- `rgb(val1, val2, val3)`: hàm tạo màu trong suốt
- `calc(px1 + px2 - px3 * ...)`: tính toán các px của ảnh (cả tuyệt đối và tương đối)
- `attr()`: thường kết hợp với mã giả sửa cùng lúc link và content

```
Thẻ::after {
  content: attr(data-title);
}
```

10. Mã giả Pseudo Classes trong CSS:

- `:root` → tham chiếu tới chính phần tử gốc HTML
- `Thẻ:hover` → sẽ thay đổi khi di chuột vào
- `Thẻ:active` → nếu giữ nút click vào sẽ thay đổi (xuất hiện sau `Thẻ:hover`)
- `Thẻ:first-child` → lấy lớp con đầu tiên của các thẻ giống nhau
- `Thẻ:last-child` → lấy lớp con cuối cùng của các thẻ giống nhau
- `Thẻ:link` → thêm định dạng đặc biệt cho một liên kết chưa được truy cập
- `Thẻ:visited` → thêm định dạng đặc biệt cho liên kết đã truy cập
- `Thẻ:focus` → thêm kiểu đặc biệt cho một phần tử trong khi phần tử đó đang được chọn
- `Thẻ:lang` → chỉ định ngôn ngữ sẽ sử dụng trong phần tử được chỉ định

VD:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Pseudo - Classes </title>
```

```

<!-- File CSS cho pseudo-class -->
<style type="text/css">
h1:hover {
    color: red;
}

h2:active {
    color: yellow;
}

li:first-child {
    color: green;
}

li:last-child {
    color: blue;
}
</style>

</head>
<body>
<h1> CSS pseudo-classes - hover </h1>
<h2> CSS pseudo-classes - active </h2>
<ul>
    <li> :root </li>
    <li> :hover </li>
    <li> :active </li>
    <li> :first-child </li>
    <li> :last-child </li>
</ul>
</body>
</html>

```

11. Mã giả Pseudo elements trong CSS:

- .nameClass::before → thay vì viết các thuộc tính Inline ngay trong thẻ thì tạo mã giả ở ngoài (lấy cái đầu tiên)
- .nameClass::after → thay vì viết các thuộc tính Inline ngay trong thẻ thì tạo mã giả ở ngoài (lấy cái đầu liền kề ngay sau)

HTML	CSS
<div> <div >="" <="" <="" class="box" div="" div>=""> <div> → HTML </div> </div></div>	<div> .box::before { <div> content: ""; display: block; width: 50px; height: 50px; background: #ff3; </div> </div>

	<pre>.box::after { content: ""; display: block; width: 50px; height: 50px; background: #ff9; }</pre> <p>→ CSS</p>
--	--

- ::first-letter → chọn ký tự đầu tiên
- ::first-line → chọn dòng đầu tiên
- ::selection → thay đổi khi bôi chuột vào

VD:

```
/* First-letter Elements */
h1::first-letter {
  font-size: 50px;
}

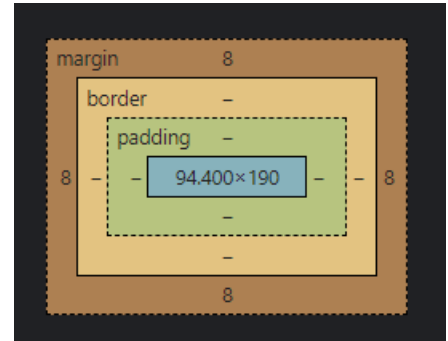
/* First-line Elements */
h2::first-line {
  font-size: 100px;
}

/* Selection Elements */
*::selection {
  background-color: violet;
}
```


Bài 7: ĐỆM, VIỀN VÀ KHOẢNG LỀ

1. Thuộc tính Padding (đệm):

- Đệm thêm viền phía ngoài khung content:
 - Content có size 30x30
 - Padding có size 10px
 - Element = Padding + content có size 40x40
- Đệm viền từng phương hướng
 - Mọi hướng: (1 giá trị)
 - padding: 10px;
 - Cả 4 hướng đều được đệm thêm
 - Cặp hướng: (2 giá trị)
 - padding: 10px 20px;
 - Giá trị đầu là đệm viền trên dưới, giá trị sau là đệm viền trái phải
 - Cặp hướng: (3 giá trị)
 - padding: 10px 20px 30px;
 - Giá trị đầu là đệm phía trên, giá trị giữa là đệm trái phải, giá trị cuối là đệm bên dưới
 - Mọi hướng: (4 giá trị)
 - padding: 10px 20px 30px 40px;
 - Theo thứ tự: trên – phải – dưới – trái
 - Mọi hướng chi tiết:
 - padding-top: 10px;
 - padding-right: 20px;
 - padding-bottom: 30px;
 - padding-left: 40px;
 - (hoặc có thể tùy chỉnh, không nhất thiết phải đủ 4 padding trên)
- Ví dụ:



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> THUỘC TÍNH PADDING (ĐỆM) </title>

  <style type="text/css">
    .box {
      width: 60px;
      height: 60px;
      color: #fff;
      display: flex;
    }
  </style>
</head>
</html>
```

```

        align-items: center;
        align-content: center;
    }
    .box1 {
        background-color: #333;
    }
    .box2 {
        background-color: orange;
    }
    .box3 {
        background-color: firebrick;
        /* Từng hướng --> Theo chiều kim đồng hồ */
        /*padding-top: 10px;
        padding-right: 10px;
        padding-bottom: 10px;
        padding-left: 10px;*/

        /* Cùng lúc mọi hướng --> mở rộng cùng lúc*/
        padding: 10px;

        /* Giá trị đầu là trên và dưới
        Giá trị sau là trái và phải
        */
        padding: 10px 10px;

        /* Giá trị đầu là bên trên
        Giá trị giữa là bên trái và phải
        Giá trị cuối là bên dưới*/
        padding: 10px 12px 8px;
    }
</style>
</head>
<body>
    <div class="box box1"> Box 1 </div>
    <div class="box box2"> Box 2 </div>
    <div class="box box3"> Box 3 </div>
</body>
</html>

```

2. Thuộc tính Border (đường viền):

- Border là lớp viền bao quanh Padding (nét liền)
- Outline cũng giống với Border nhưng không chiếm nhiều không gian và có hình dạng bất kỳ (xuất hiện mọi mặt)
- Các loại bo viền Border:
 - border-width: <giá trị>px → bo tất cả các viền xung quang Padding
 - border-style: <type> → mặc định chiếm 2px (loại viền: nét liền solid)

- border: <kích cỡ> <loại viền> <màu sắc> → Có thể lược bỏ một vài giá trị
 - Các loại bo viền chi tiết:
 - border-top-width: 10px;
 - border-right-width: 10px;
 - border-bottom-width: 10px;
 - border-left-width: 10px;
 - VD:
-

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title> THUỘC TÍNH BORDER (ĐƯỜNG VIỀN) </title>

    <!-- File CSS -->
    <style type="text/css">
        /* Class chung */
        .box {
            width: 60px;
            height: 60px;
            color: #fff;
            display: flex;
            align-items: center;
            align-content: center;
        }

        /* Các class riêng */
        .box1 {
            background-color: #333;
        }
        .box2 {
            background-color: orange;
        }
        .box3 {
            background-color: firebrick;
            padding: 10px;

            /* Bo viền Border */
            border-width: 10px;
            border-style: solid;
            border-color: #333;
            /* solid mặc định 2px và là kiểu nét liền */
```

```

        /* Bo viền từng thành phần */
        border-top-width: 10px;
        border-right-width: 10px;
        border-bottom-width: 10px;
        border-left-width: 10px;

        /* Cách viết tắt border: <kích cỡ> <loại viền> <màu sắc> */
        border: 10px solid #333;
    }
</style>
</head>
<body>
    <div class="box box1"> Box 1 </div>
    <div class="box box2"> Box 2 </div>
    <div class="box box3"> Box 3 </div>
</body>
</html>

```

3. Thuộc tính Margin (khoảng cách lề):

- Không như Padding và Border (phụ thuộc vào chính element đó), thuộc tính Margin có tác dụng dẫn cách lề giữa các elements khác nhau
- Khoảng cách lề từng phương hướng
 - Mọi hướng: (1 giá trị)
 - margin: 10px;
 - Cả 4 hướng đều được dẫn cách
 - Cặp hướng: (2 giá trị)
 - margin: 10px 20px;
 - Giá trị đầu là dẫn cách viền trên dưới, giá trị sau là dẫn cách viền trái phải
 - Cặp hướng: (3 giá trị)
 - margin: 10px 20px 30px;
 - Giá trị đầu là dẫn cách phía trên, giá trị giữa là dẫn cách trái phải, giá trị cuối là dẫn cách bên dưới
 - Mọi hướng: (4 giá trị)
 - padding: 10px 20px 30px 40px;
 - Theo thứ tự: trên – phải – dưới – trái
 - Mọi hướng chi tiết:
 - margin-top: 10px;
 - margin-right: 20px;
 - margin-bottom: 30px;
 - margin-left: 40px;
 - (hoặc có thể tùy chỉnh, không nhất thiết phải đủ 4 padding trên)

- VD:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> THUỘC TÍNH MARGIN (KHOẢNG CÁCH LỀ) </title>

  <!-- File CSS -->
  <style type="text/css">
    /* Class chung */
    .box {
      width: 60px;
      height: 60px;
      color: #fff;
      display: flex;
      align-items: center;
      justify-content: center;
      float: left;
    }

    /* Các class riêng */
    .box1 {
      background-color: #333;
      margin-left: 20px;
      margin-right: 20px;
    }
    .box2 {
      background-color: orange;
      margin-right: 20px;
      margin-bottom: 20px;
    }
    .box3 {
      background-color: firebrick;

      /* Cú pháp viết tắt giống Padding */
      margin: 10px 20px 30px 40px;
    }

  </style>
</head>
<body>
  <div class="box box1"> Box 1 </div>
  <div class="box box2"> Box 2 </div>
  <div class="box box3"> Box 3 </div>
</body>
</html>

```

4. Thuộc tính Boxing-sizing:

- Sử dụng thuộc tính này khi có sự thay đổi hay bất đồng các Padding – Border – Margin

- Nguyên tắc hoạt động: chiều ngang element = border + padding + content
 - **box-sizing**: border-box; (giá trị phù hợp khi có padding và border)
 - **box-sizing**: content-box; (giá trị mặc định như ban đầu)
- VD:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> THUỘC TÍNH BOX-SIZING </title>
  <style type="text/css">
    html, body {
      margin: 0;
    }

    .box {
      width: 100px;
      height: 100px;
      color: #fff;
      background-color: darkorchid;

      /* Tạo khoảng cách lề với chữ */
      padding: 16px;
      border: 2px solid #333;

      box-sizing: border-box;
    }
  </style>
</head>
<body>
  <div class="box">
    Đây là nội dung về Box-sizing
  </div>
</body>
</html>
```

Bài 8: THUỘC TÍNH TẠO NỀN

1. Thuộc tính Background-image:

- Đặt hình nền background là một bức ảnh được gán url():
 - **background-image:** url(linkImage)
- background-image có tính chất lặp lại từng ảnh nên ta sử dụng thêm thuộc tính khác để kiểm soát số lần lặp lại là **background-repeat**:
 - **background-repeat: no-repeat;** → Không lặp lại ảnh
 - **background-repeat: repeat-x;** → Lặp lại theo chiều ngang
 - **background-repeat: repeat-y;** → Lặp lại theo chiều dọc
- Có thể đặt nhiều ảnh làm background và chúng cách nhau bởi dấu phẩy, ảnh nào gọi trước thì ưu tiên ở phía trên hơn
 - **background-image:** url(linkImage1), url(linkImage2);
- Chuyển background dài màu bằng hàm linear-gradient():
 - **background-image:** linear-gradient(<chiều xoay>, màu 1, màu 2)
- VD:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> THUỘC TÍNH BACKGROUND-IMAGE </title>
  <style type="text/css">
    html, body {
      margin: 0;
    }

    .header {
      width: 100%;
      height: 100vh;

      /* Đặt hình nền Background là ảnh */
      background-image: url(link1), url();
      background-size: 100%;
      background-repeat: no-repeat;

      /* Dải màu background giảm dần */
```

```

background-image: linear-gradient(0, rgba(255, 255, 0, 0.5), rgba(0, 0, 255, 255));
    }
</style>
</head>
<body>
    <div class = "header"> </div>
</body>
</html>

```

2. Thuộc tính Background-size với cover, contain:

- Nếu **background-size: contain;** → lấy background to nhất có thể của trang web (hoàn chỉnh, không bị che khuất)
- Nếu **background-size: cover;** → chọn theo chiều dài nhất của bức ảnh nhưng sẽ chịu sự che khuất khi mở rộng hoặc thu nhỏ trang web (không cho phép có khoảng trắng dư thừa trên web → Toàn background)

3. Các thuộc tính background khác:

- background-origin: được sử dụng khi đi kèm với background-image, căn chỉnh ảnh từ lớp A (content-box, padding-box, border-box, ...) → lấy ảnh đồ về từ lớp
- background-position: vị trí hình ảnh nền trong CSS (khi gọi một giá trị thì giá trị còn lại mặc định là center)
 - Truyền hai giá trị tuyệt đối:
 - top right: ảnh ở trên cùng bên phải
 - top left: ảnh ở trên cùng bên trái
 - bottom right: ảnh ở dưới cùng bên phải
 - bottom left: ảnh ở dưới cùng bên trái
 - center: căn giữa
 - Truyền hai giá trị tương đối:
 - <value1> <value2>: dẫn cách chiều ngang và dẫn cách chiều dọc
 - Truyền bốn giá trị:
 - <vị trí tuyệt đối 1> <value1> <giá trị tuyệt đối 2> <value2>
 - background-position: top 10px right 10px → ưu tiên lấy gốc từ giá trị tuyệt đối
- Cách viết tắt shorthand Background:
 - Cách 1: Nếu viết từng thuộc tính có thể ghi luôn background: <thuộc tính>; chứ không cần ghi cụ thể background-origin, background-position, ...
 - Cách 2: Nếu viết gộp các thuộc tính với nhau:
 - **background: <background-image> <background-repeat> <background-position> / <background-size>;**
 - VD: **background: url(link) no-repeat center / contain;**

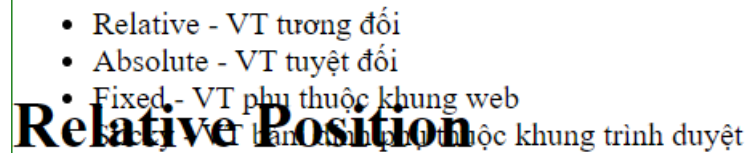
4. Mở rộng các hàm giá trị màu sắc của CSS:

- `rgb(red, green, blue)`: mỗi màu chạy từ giá trị 0-255 (8 bit)
 - Nếu ba màu có giá trị bằng nhau sẽ là màu trắng
 - Nếu cả ba màu là 0 thì là màu đen
 - Mã màu sắc luôn có dạng thập lục phân (0123456789abcdef)
- `rgb(red green blue / alpha)`
- `rbg(red, green, blue, alpha)`: alpha có thể chứa màu dạng số nguyên hoặc số thập phân
 - Cách quy đổi (làm tròn $[\alpha * 255]$) sang thập lục phân sẽ có dạng #....

Bài 9: THUỘC TÍNH VỊ TRÍ POSITION

1. CSS position relative:

- Tạo giao diện có vị trí tương đối khác đề lên giao diện hiện tại (thông báo, tin nhắn, trang chủ, ...)
- Relative là vị trí tương đối không bị phụ thuộc vào đối tượng nào khác cả (lấy chính vị trí mà nó đang đứng để làm gốc tọa độ):
 - **position: relative;**
- Có thể di chuyển tự do vị trí element có Relative Position mà không phụ thuộc bất kỳ vào đối tượng nào (có thể đề lên các element khác bằng top – right – bottom – left)



• Relative - VT tương đối
• Absolute - VT tuyệt đối
• Fixed - VT phụ thuộc khung web
• Sticky - VT bám dính phụ thuộc khung trình duyệt

Relative Position

2. CSS position absolute:

- Tạo giao diện có vị trí tuyệt đối khác phủ đề lên giao diện hiện tại (thông báo, tin nhắn, trang chủ, ...)
- Absolute là vị trí tương đối bị phụ thuộc vào thẻ cha gần nhất chứa Relative Position:
 - **position: absolute;**
- Sau khi dùng Position thì ở dưới xác định vị trí sẽ ưu tiên Top hơn Bottom, ưu tiên Left hơn Right (mặc định)

3. CSS position fixed:

- Giống cái phần header trong các trang web (đứng yên, không bị thay đổi khi kéo lên kéo xuống thanh trượt)
- Khác với Absolute thì Fixed Position phụ thuộc chính vào cửa sổ trình duyệt

4. CSS position sticky:

- Tạo menu, header bám dính (na ná với Fixed Position)
- Header vẫn chuyển động theo thanh trượt lên xuống nhưng sẽ bám dính vào trang web khi Element chứa Sticky Position chạm đến điểm dừng phía trên