

Phần 1: Hyper Text Markup Language

1. Định dạng cho văn bản HTML

- ➔ Định dạng nổi bật của chữ: sử dụng cặp thẻ từ <h1></h1> đến <h6></h6>. Cặp thẻ h có các thuộc tính align: left, right, center, justify
- ➔ Định dạng chữ:
 - Thuộc tính face: arial, time new roman,...
 - Thuộc tính size: (-1,-6), (+1,+6), (1,7).
 - Thuộc tính color: Tên màu hoặc mã màu (#...).
 - Cặp thẻ căn giữa vị trí: <center></center>
- ➔ Định nghĩa một đoạn văn: <p>paragraph</p>
 - Thẻ p tự động xuống dòng và thêm margin
- ➔ Ngắt hai nội dung HTML (ngang): <hr>
- ➔ Ngắt dòng (line breaks/new line):

- ➔ Chỉ định các văn bản được định dạng trước: <pre> </pre>
 - Bảo toàn khoảng trắng và ngắt dòng trong văn bản
- ➔ Text formatting:

HTML Tag	Description
	Chữ in đậm (bold)
	Văn bản quan trọng (important)
<i>	Chữ in nghiêng (italic)
	Emphasized text: nhấn mạnh
<small>	Small text
<mark>	Mark/highlight text
	Delete text: chữ bị gạch
<ins>	Insert txt: chữ gạch dưới

<sub>	Subscript text, like H ₂ O
<sup>	Superscript text, like www ^[1]

➔ HTML quotation: trích dẫn

Thẻ	Mô tả
<abbr>	Định nghĩa chữ viết tắt hoặc từ viết tắt.
<address>	Sử dụng để xác định thông tin liên lạc của tác giả tài liệu.
<bdo>	Định nghĩa hướng của văn bản.
<blockquote>	Được sử dụng để xác định một phần được trích dẫn từ một nguồn khác.
<q>	Được sử dụng để đặt các trích dẫn nhỏ vào website.
<cite>	Định nghĩa tiêu đề nguồn từ nơi trích dẫn hoặc hoạt động được đưa ra.
<define>	Được sử dụng để định nghĩa các điều khoản.

2. HTML Links – Hyperlinks

- ➔ Syntax: ` link text` //href attribute to define the link address
- ➔ Target attribute:
 - _self : default, mở tài liệu trong cùng cửa sổ/ tab khi clicked
 - _blank : mở trong cửa sổ/tab mới
 - _parent : mở trong khung cha
 - _top: mở trong toàn bộ cửa sổ
- ➔ Đường dẫn tuyệt đối vs đường dẫn tương đối:
 - Đường dẫn tuyệt đối: đường dẫn đầy đủ tới trang web (<https://www.google.com/>)
 - Đường dẫn tương đối: liên kết các trang trong cùng một trang web, không có phần ["https://www"](https://www) (/asset/header.php).
- ➔ Sử dụng image như một link: chỉ cần đặt thẻ trong thẻ <a>
- ➔ Link tới một địa chỉ email:
 - Sử dụng **mailto:** trong thuộc tính href để tạo liên kết mở chương trình email của người dùng.
 - Eg: `Send email`

- ➔ Button như một liên kết: sử dụng javascript events trong HTML button
Eg: `<button onclick="document.location='default.asp'">HTML Tutorial</button>`
- ➔ Link title attribute: title = “”;
Chỉ định thêm thông tin về một yếu tố, thông tin thường hiển thị dưới dạng văn bản chú giải khi di chuyển chuột qua phần tử.
- ➔ HTML links - create a bookmark
Sử dụng thuộc tính id = “value” tạo bookmark
`<h2 id="C4">Chapter 4</h2>`
Sau đó thêm link đến bookmark (href = “#value”)
`Jump to Chapter 4`

3. HTML images

- ➔ Syntax: ``
- ➔ Attribute:
 - Thuộc tính src: đường dẫn ảnh
 - Thuộc tính width: độ dài của ảnh
 - Thuộc tính height: độ cao của ảnh
 - Thuộc tính alt: mô tả của ảnh, khi tải ảnh lỗi
 - Thuộc tính border: đường viền của ảnh
 - Thuộc tính align: vị trí của ảnh (left, right)
 - Thuộc tính hspace: Khoảng cách margin theo chiều ngang
 - Thuộc tính vspace: Khoảng cách margin theo chiều dọc
 - usemap...
- ➔ Image maps: sử dụng usemap attribute
Xác định một số khu vực có thể click trong bản đồ hình ảnh ược định nghĩa bởi thẻ `<area>` với các attribute shape(hình dạng), coords(tọa độ), alt, href.
``


```

<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>

```
- ➔ Image background: (CSS)
- ➔ Image picture element: cho phép hiển thị hình ảnh khác nhau cho các thiết bị/màn hình khác nhau.
Eg: `<picture>`

```

<source media="(min-width: 650px)" srcset="img_food.jpg">
<source media="(min-width: 465px)" srcset="img_car.jpg">

</picture>

```

`<source>` tag: đề cập tới các hình ảnh khác nhau thông qua thuộc tính srcset, trình duyệt sẽ chọn hình ảnh đầu tiên phù hợp nhất với chế độ xem và thiết bị hiện tại. Media attribute xác định khi hình ảnh là phù hợp nhất.

Luôn để thẻ cuối cùng (default) trong cặp thẻ <picture> </picture> tránh trường hợp không có <source> nào phù hợp.

Khi nào sử dụng picture:

- Bảng thông: khi thiết bị nhỏ, không cần load các ảnh lớn
- Format sport: một số trình duyệt/ thiết bị không hỗ trợ tất cả image format

4. HTML tables

→ Syntax:

```
<table>
  <tr><th></th></tr>
  <tr><td></td></tr>
  ...
</table>
```

- Cặp thẻ <table></table> : Khai báo bảng
- Cặp thẻ <th></th> : Khai báo tiêu đề của cột trong bảng
- Cặp thẻ <tr></tr> : Khai báo một dòng/ hàng (row) trong bảng
- Cặp thẻ <td></td> : Khai báo một cột của một dòng trong bảng

→ Note:

Phải có đủ 3 thành phần

Phải có nội dung trong bảng (text, images, lists, other tables, etc)

→ Thuộc tính của thẻ <table></table>: (kết hợp CSS)

Width, height, border, bordercolor, align, bgcolor, background, cellpadding, cellspacing, border-collapse(viên còn 1 đường)

→ Thuộc tính thẻ <tr></tr>:

Height, bgcolor, align, valign,

→ Thuộc tính thẻ <td></td>

Width, height, bgcolor, background, align, valign.

→ Tùy biến thành bảng không đối xứng

→ Thuộc tính colspan: Gộp các cột trên cùng một hàng của một bảng

Thuộc tính colspan có giá trị là số cột muốn gộp và nó được đặt tại thẻ td bắt đầu gộp. Sau đó xóa bỏ các cột đã gộp.

→ Thuộc tính rowspan: Gộp các cột trên nhiều hàng khác nhau trong một bảng

Thuộc tính rowspan có giá trị là số hàng muốn gộp và nó được đặt tại thẻ td bắt đầu gộp. Sau đó xóa bỏ các hàng đã gộp

→ Thêm phụ đề cho bảng:

Sử dụng <caption> tag ngay sau <table> tag

→

5. HTML lists

→ Unordered list: (danh sách không có thứ tự)

syntax:

```
<ul>
  <li>list1</li>
  <li>list2</li>
  <li>...</li>
</ul>
```

→ Nested HTML list: list inside list

➔ Ordered list: (danh sách có thứ tự)

syntax:

```
<ol>
  <li>list1</li>
  <li>list2</li>
  <li>...</li>
</ol>
```

➔ Description list: danh sách mô tả

syntax:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

dl: khai báo một danh sách mô tả

dt: khai báo thuật ngữ (term) cần mô tả

dd: mô tả cho thuật ngữ

6. Block & inline HTML

➔ Block: Luôn bắt đầu trên một dòng mới và chiếm toàn bộ chiều rộng có sẵn.

Một số thẻ block HTML: <address> <article> <aside> <blockquote> <canvas>
<dd> <div> <dl> <dt> <fieldset> <figcaption> <figure> <footer> <form>
<h1>-<h6> <header> <hr> <main> <nav> <noscript> <p> <pre>
<section> <table> <tfoot> <video>

➔ Inline: Không bắt đầu trên dòng mới và chỉ chiếm chiều rộng cần thiết. Một thẻ inline HTML không thể chứa thẻ block HTML bên trong.

Một số thẻ inline HTML: <a> <abbr> <acronym> <bdo> <big>

<button> <cite> <code> <dfn> <i> <input> <kbd> <label> <map>
<object> <output> <q> <samp> <script> <select> <small>
<sub> <sup> <textarea> <time> <tt> <var>

- ➔ Thẻ div (block-level) thường được sử dụng làm vùng chứa cho các phần tử HTML khác. Thẻ span (inline) được sử dụng để đánh dấu một phần của văn bản hoặc một phần của tài liệu.

7. HTML classes

- ➔ HTML class attribute được sử dụng để chỉ định một lớp cho một phần tử HTML, các phần tử HTML có thể chia sẻ cùng class.
Sử dụng: thuộc tính class thường được sử dụng để trỏ đến một tên lớp trong CSS. Sử dụng dấu chấm để truy cập tới lớp muốn style.
Ex:

```
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
}
</style>
</head>
<body>

<div class="city">
<h2>London</h2>
<p>London is the capital of England.</p>
</div>

</body>
</html>
```

- ➔ Multiple classes: một phần tử HTML có thể thuộc về nhiều hơn một lớp
`<h2 class="city main">London</h2>`
- ➔ Sử dụng thuộc tính class trong Javascript: tên lớp cũng có thể được sử dụng bởi Javascript để thực hiện các tác vụ nhất định cho các thành phần cụ thể.
Javascript có thể truy cập tới các phần tử HTML với một tên lớp cụ thể bằng phương thức `getElementsByClassName("nameclass")`.

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
```

8. HTML id

- Được sử dụng để chỉ định một id duy nhất cho một phần tử HTML, không thể có nhiều hơn một thành phần có cùng id trong một tài liệu HTML.

Sử dụng: thuộc tính id thường được sử dụng để trỏ đến một tên id trong CSS. Sử dụng dấu # để truy cập tới lớp muốn style.

Ex:

```
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h2>The id Attribute</h2>
<p>Use CSS to style an element with the id "myHeader":</p>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

- Sử dụng thuộc tính id trong Javascript để thực hiện một số tác vụ cụ thể: Javascript có thể truy cập một phần tử HTML với một id cụ thể bằng phương thức `getElementById("nameId")`.

```
<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

9. HTML iframes

Một HTML iframe được sử dụng để hiển thị một trang web vào trong một trang web.

Syntax: `<iframe src="url" title="description">`
`//defines an inline frame`

➔ Iframe - set Height & Width

Sử dụng thuộc tính width, height attributes HTML

Ex: `<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>`

Hoặc có thể thêm style attribute và CSS width & height

Ex: `<iframe src="demo_iframe.htm" style="height:200px;width:300px;" title="Iframe Example"></iframe>`

➔ Remove the border: theo mặc định iframe có được viền xung quanh nó (có thể CSS cho border.

Xóa bỏ đường biên:

`<iframe src="demo_iframe.htm" style="border:none;" title="Iframe Example"></iframe>`

➔ Iframe - target for a link

Một iframe có thể làm khung đích cho một liên kết, thuộc tính target của link phải tham chiếu đến thuộc tính name của của iframe.

Ex:

`<iframe src="demo_iframe.htm" name="iframe_a" title="Iframe Example"> </iframe>`

`<p> W3Schools.com </p>`

10. HTML header

➔ Phần tử HTML <head> là một thùng chứa cho các thành phần (elements) sau: <title>, <style>, <meta>, <link>, <script>, and <base>.

◆ <title>: xác định tiêu đề của tài liệu

◆ <style>: dùng để css cho trang web

◆ <link>: dùng để xác định mối quan hệ giữa tài liệu hiện tại và tài nguyên bên ngoài. Thẻ link thường được sử dụng để liên kết CSS bên ngoài vào.

`<link rel="stylesheet" href="mystyle.css">`

◆ <script>: được sử dụng để xác định Javascript phía máy khách.

◆ <base>: chỉ định URL cơ sở và/hoặc target cho tất cả các URL tương đối cho một trang.

`<base href="https://www.w3schools.com/" target="_blank">`

➔ HTML <meta> elements:

Thường được sử dụng để chỉ định bộ ký tự, mô tả trang, từ khóa, tác giả của tài liệu và cài đặt chế độ xem.

Ex:

Define the character set used:


```
<meta charset="UTF-8">
```

Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Define a description of your web page:

```
<meta name="description" content="Free Web  
tutorials">
```

Define the author of a page:

```
<meta name="author" content="John Doe">
```

Refresh document every 30 seconds:

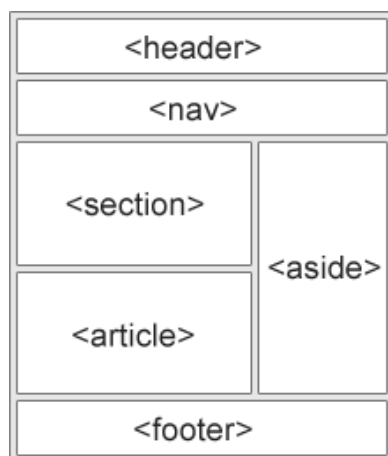
```
<meta http-equiv="refresh" content="30">
```

Setting the viewport to make your website look good on all devices:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

11. HTML layout

➔ **HTML layout elements:**



- `<header>` - Defines a header for a document or a section
- `<nav>` - Defines a set of navigation links
- `<section>` - Defines a section in a document
- `<article>` - Defines an independent, self-contained content
- `<aside>` - Defines content aside from the content (like a sidebar)
- `<footer>` - Defines a footer for a document or a section
- `<details>` - Defines additional details that the user can open and close on demand
- `<summary>` - Defines a heading for the `<details>` element

➔ **HTML layout techniques:** 4 kỹ thuật

CSS framework (bootstrap,...)
CSS float property (left, right, clear,...)
CSS flexbox
CSS grid

12. HTML responsive web design

→ Setting the viewport:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

→ Responsive images: hình ảnh có tỉ lệ độ phóng đại để phù hợp với mọi kích thước trình duyệt.

set **max-width: 100%**

```

```

→ Hiển thị các hình ảnh khác nhau khi thay đổi kích thước trình duyệt: <picture>

→ Responsive text size: text size có thể được set với một vw (viewport width), kích thước text sẽ được thay đổi theo kích thước của trình duyệt.

1vw = 1% viewport width,

→ Media queries

→ Responsive – framework: bootstrap, w3.css

13. HTML form

→ Tạo form: <form></form>

Thuộc tính của thẻ form:

+ Name, action (hướng xử lý dữ liệu cho form), method (Get/Post).

Các phần tử trong form:

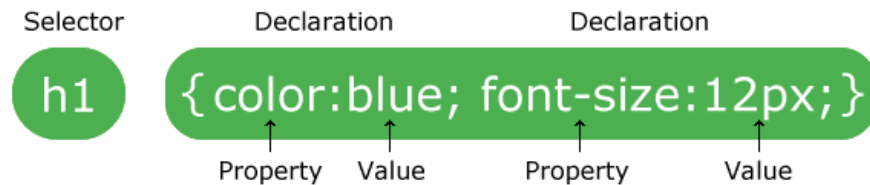
+ Khai báo sử dụng với thẻ <input/>, đưa vào type.

+ Các phần tử: textbox, password, checkbox, radio, file, submit, reset, textarea, selectbox, button.

Các thuộc tính của các phần tử: type, name, value, checked, required, disabled, placeholder, ...

Cascading Style Sheets

1. CSS syntax



Example:

```
p{  
    color:red;  
    text-align:center;  
}  
→ Comment: /* this is a comment */
```

2. CSS selectors

Selector	Example	Example description
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element,..</u>	div, p	Selects all <div> elements and all <p> elements

3. How to add CSS

- Có 3 cách để chèn CSS
 - + External CSS: xây dựng CSS ra một file riêng (.css), triệu gọi CSS vào file HTML
`<link rel="stylesheet" type="text/css" href="mystyle.css">`
 - + Internal CSS: khai báo CSS trong phần HEAD của văn bản HTML, sử dụng thẻ `<style>`
 - + Inline CSS: xây dựng các câu lệnh CSS ngay trong các thẻ HTML
- Thứ tự: CSS inline > external > internal > browser default

4. Bố cục trong CSS (width, height, float, clear)

- Width (độ rộng này không bao gồm border, padding, margin)
- Height (độ cao này không bao gồm border, padding, margin)

➔ Float:

+ Thuộc tính float dùng để xác định vị trí của một thành phần nằm về bên trái hoặc bên phải so với thành phần bao ngoài của nó và tạo ra khoảng không gian trống ở bên còn lại (nếu có)

+ Giá trị của thuộc tính float: left (nằm về bên trái) , right (nằm về bên phải) và none (default)

+ Chú ý: Khi áp dụng Float cho một phần tử HTML bất kỳ thì phần nội dung phía dưới sẽ được tự động tràn lên lấp đầy khoảng trống mà float tạo ra (nếu có)

➔ Clear: thuộc tính clear có tác dụng ngăn cản việc tràn lên ở bên trái (left), bên phải (right) hay ở cả 2 bên (both) do thuộc tính float tạo ra.

5. CSS backgrounds

➔ Thuộc tính Background-color: xác định màu nền

Example:

```
body{  
    background-color: lightblue;  
}
```

➔ Màu thường được chỉ định bởi:

+ Tên màu

+ Mã màu HEX (vd: #ff0000)

+ Mã màu RGB (vd: rgb(255,0,0))

➔ Thuộc tính opacity/transparency (độ mờ/ độ trong)

+ Có giá trị từ 0.0 đến 1.0, giá trị càng nhỏ càng mờ

+ Default: 1

+ Tranparency sử dụng RGBA:

```
div{  
    background: rgba(0, 128, 0, 0.3)  
    /* Green background with 30% opacity */  
}
```

➔ Thuộc tính Background-image: xác định ảnh nền, mặc định hình ảnh được lặp lại (repeat) để bao phủ toàn bộ thành phần.

Example:

```
body{  
    background-image: url("paper.gif");  
}
```

➔ Thuộc tính Background-repeat: xác định chiều lặp lại

+ Value: repeat-x, repeat-y, repeat (cả x,y), no-repeat.

➔ Thuộc tính Background-attachment:

+ Dùng để thiết lập ảnh nền được cố định hay cuộn theo thanh cuộn trang.

+ Giá trị:

Fixed: Cố định ảnh nền tại một vị trí xác định

Scroll: Ảnh nền tự động bị cuộn theo thanh cuộn

➔ Thuộc tính Background-position giá_trị_1 giá_trị_2;

+ Thiết lập vị trí ảnh nền cho thành phần

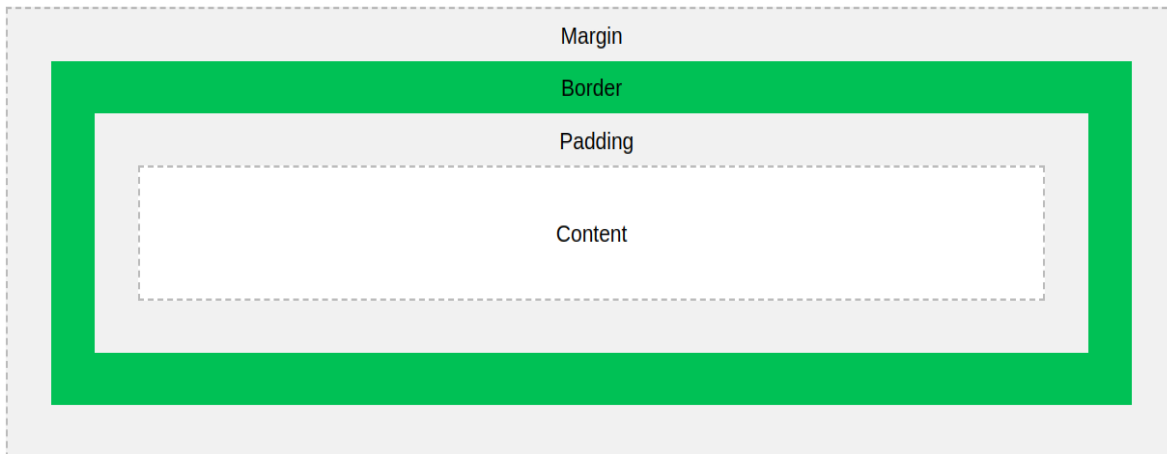
+ value: giá trị độ dài hoặc left/right/top/bottom/center

→ CSS background shorthand

Background: <background-color> | <background-image> |
<background-repeat> | <background-attachment> | <background-position>;

6. Mô hình hộp BOXMODEL

→ Gồm: padding (vùng đệm), border (viền), margin (canh lề) và các tùy chọn.



6.1 Thuộc tính Padding

→ Vùng đệm padding chính là khoảng cách giữa phần nội dung bên trong và đường viền (border)

→ Value: giá trị độ dài

→ Syntax: padding-left/right/top/bottom: giá_trị;

→ Shorthand:

+ padding: trên phải dưới trái;

+ padding: trên_dưới trái_phải;

+ padding: trên_dưới_trái_phải;

6.2 Thuộc tính BORDER

→ Border-width: giá_trị;

+ Thiết lập độ dày đường viền

+ Giá trị: giá trị độ dài

→ Border-color: giá_trị;

+ Thiết lập màu cho đường viền

+ Giá trị: tên màu/ mã màu

→ Border-style: giá_trị;

+ Thiết lập kiểu đường viền

+ Giá trị: solid, double, inset, none,...

→ Shorthand:

+ border: weight color style;

6.3 Thuộc tính MARGIN

→ Thiết lập khoảng cách giữa đường viền (border) với các nội dung bên ngoài

→ Giá trị: giá trị độ dài

→ Syntax:

- + Margin-top: giá_trị
- + Margin-right: giá_trị
- + Margin-bottom: giá_trị
- + Margin-left: giá_trị

→ Shorthand:

- + Margin: trên phải dưới trái;
- + Margin: trên_dưới trái_phải;
- + Margin: trên_phải_dưới_trái;

7. CSS outline

➔ Một đường được vẽ xung quanh các phần tử (elements), bên ngoài border để làm cho các phần tử nổi bật.

➔ Outline properties:

- + outline-style
- + outline-color
- + outline-width
- + outline-offset
- + outline

7.1 Outline style

→ value:

- dotted - Defines a dotted outline
- dashed - Defines a dashed outline
- solid - Defines a solid outline
- double - Defines a double outline
- groove - Defines a 3D grooved outline
- ridge - Defines a 3D ridged outline
- inset - Defines a 3D inset outline
- outset - Defines a 3D outset outline
- none - Defines no outline
- hidden - Defines a hidden outline

7.2 Outline width

→ Value:

- thin (typically 1px)
- medium (typically 3px)
- thick (typically 5px)
- A specific size (in px, pt, cm, em, etc)

7.3 Outline color

→ Value:

- name - specify a color name, like "red"
- HEX - specify a hex value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"

HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
invert - thực hiện đảo ngược màu (đảm bảo nhìn thấy outline, bất kể màu nền (color background)).

7.4 Shorthand: outline

→ **outline**: **outline-width** | **outline-style** | **outline-color**;

(note: outline style (required)).

→ Example:

```
p.ex1 {outline: dashed;}
p.ex2 {outline: dotted red;}
p.ex3 {outline: 5px solid yellow;}
p.ex4 {outline: thick ridge pink;}
```

7.5 Outline offset

→ Chỉ định khoảng trắng giữa outline và border của một element.

→ Value: đơn vị độ dài.

→ Example: **outline-offset**: 15px;

8. CSS text

- color: giá_trị;
 - + Thiết lập màu chữ cho đoạn text
 - + Giá trị: tên màu/ mã màu
- text-indent: giá_trị;
 - + Thiết lập khoảng thụt đầu dòng cho đoạn text
 - + Giá trị: giá trị độ dài
- text-align: giá_trị;
 - + Thiết lập vị trí cho đoạn text so với thành phần bao ngoài của nó
 - + Giá trị: left, right, center, justify(căn đều 2 bên)
- letter-spacing: giá_trị;
 - + Thiết lập khoảng cách giữa các ký tự trong văn bản
 - + Giá trị: giá trị độ dài
- word-spacing: giá_trị;
 - + Thiết lập khoảng cách giữa các từ trong văn bản
 - + Giá trị: giá trị độ dài
- line-height: giá_trị;
 - + Thiết lập khoảng cách giữa các dòng trong văn bản
 - + Giá trị: giá trị độ dài
- text-decoration: giá_trị;
 - + Sử dụng để định dạng kiểu chữ
 - + Giá trị:
 - Underline: gạch chân
 - Line-through: gạch ngang
 - Overline: gạch trên
 - None
- text-transform: giá_trị;

- + Dùng để thiết lập kiểu chữ
- + Giá trị:
 - Uppercase: chữ in hoa
 - Lowercase: chữ in thường
 - Capitalize: viết hoa chữ cái đầu tiên của mỗi từ
 - None: không áp dụng hiệu ứng (mặc định)
- ➔ text-shadow: thêm shadow (bóng) cho text.

9. CSS fonts

- ➔ font-family: giá_trị;
 - + Thiết lập font chữ cho thành phần
 - + Giá trị: Times new roman, arial, courier new,...
 - + Example:


```
font-family: "Times New Roman", Times, serif;
```
- ➔ font-style: giá_trị;
 - + Thiết lập kiểu chữ cho thành phần
 - + Giá trị:
 - Italic : Kiểu chữ in nghiêng
 - Normal: Kiểu chữ thường
- ➔ font-variant: small-caps;
 - + Thiết lập kiểu chữ in hoa nhỏ
- ➔ font-weight: giá_trị;
 - + Thiết lập độ đậm cho kí tự
 - + Giá trị:
 - Normal : Chữ thường
 - Bold: Chữ in đậm
 - 100 - 900: Tăng dần độ đậm cho chữ
- ➔ font-size: giá_trị;
 - + Thiết lập cỡ chữ
 - + Giá trị:
 - Giá trị độ dài
 - Default: 16
 - + responsive font size:


```
<h1 style="font-size:10vw">Hello World</h1> /*10% viewport*/
```
- ➔ Shorthand:
 - + syntax:


```
font: font-style|font-variant|font-weight|font-size/line-height|font-family;
```
 - + Example:


```
p.b {
  font: italic small-caps bold 12px/30px Georgia, serif;
}
```

10. CSS link

- ➔ Sự kiện link bắt trạng thái liên kết khi chưa có click


```
a:link { thuộc_tính_1: giá_trị; ... thuộc_tính_N: giá_trị;}
```

➔ Sự kiện hover bắt trạng thái liên kết đang được hơ chuột

```
a:hover { thuộc_tính_1: giá_trị; ... thuộc_tính_N: giá_trị; }
```

➔ Sự kiện active bắt trạng thái liên kết đang được giữ click

```
a:active { thuộc_tính_1: giá_trị; ... thuộc_tính_N: giá_trị; }
```

➔ Sự kiện visited bắt trạng thái liên kết đã được click

```
a:visited { thuộc_tính_1: giá_trị; ... thuộc_tính_N: giá_trị; }
```

➔ Điều khiển cho thẻ HTML(tag), sự kiện hover bắt trạng thái khi thẻ HTML được hơ chuột qua

```
+ HTML tag:hover {  
    thuộc_tính_1: giá_trị;  
    ...  
    thuộc_tính_N: giá_trị;  
}
```

+

```
a:hover p{...}
```

 : Khi a được hơ chuột thì p (...)

11. CSS display

➔ Display: chỉ định phần tử được hiển thị, giúp kiểm soát bố cục. Mỗi thành phần elements có một kiểu hiển thị mặc định khác nhau (block, inline).

➔ Override lại giá trị mặc định:

Example 1:

```
li { //default: block  
    display: inline;  
}
```

Example 2:

```
span { //default: inline  
display: block;  
}
```

➔ Display none: ẩn phần tử, cho các phần tử khác hiển thị tại vị trí của mình.

```
h1.hidden {  
    display: none;  
}
```

➔ Visibility hidden: ẩn phần tử nhưng vẫn chiếm vị trí không cho phần tử khác hiển thị vào vị trí của mình.

```
h1.hidden {  
    visibility: hidden;  
}
```

➔ from

(Chú ý kiến thức ngoài lề: **Overflow: hidden**)



CSS3:

+ thuộc tính di chuyển:

Transform: translateX(200px);

Transform: translateY(300px);

Transform: translateZ(300px); (cần thuộc tính kích hoạt perspective: 300px

ở div bố làm điểm đặt mắt)

+ thuộc tính zoom

Transform: scale(3); (zoom lên 300%)

Transform-origin: x y (tâm zoom)

+ Position trong CSS: position: static | relative | fixed | absolute

- Static: tĩnh

- Relative: di chuyển căn chỉnh theo vị trí tương đối của nó

- Fixed : di chuyển căn chỉnh theo vị trí trình duyệt
 - Absolute: di chuyển căn chỉnh theo vị trí của thẻ tag bố đầu tiên chưa thuộc tính relative
- + Transition-delay: 0.4s; => delay 0.4s/
 - + Hiệu ứng RotateY, Backface...
 - + Box-shadow: 0 0 0 0 back; => tạo 2 đường viền.

