Cơ bản về CSS3(1 tuần)

Nội dung kiến thức đạt được:

- CSS3 là gì?

- Biết cách đặt tên cho selector(class, id, name) cho HTML để sử dụng trong CSS3

- Biết cách sử dụng các thuộc tính cơ bản trong CSS3

- Biết cách sử dụng các thuộc tính phức tạp

- Biết cách viết animation

- Cách sử dụng DevTool.

1. **CSS3 là gì?**

Nếu HTML là 1 bộ khung xương người thì CSS là phần màu da, tóc,… của người đó. Như vậy CSS(Cascade Style Sheet) được định nghĩa là một ngôn ngữ để định dạng lại HTML sao cho phần hiển thi lên trang web trở lên sinh động và hấp dẫn hơn.

Vậy CSS3 chỉ là một chuẩn mới của CSS với một sự cải tiến hơn so với CSS. Nó có thể kết hợp jQuery thì website của bạn sẽ trở nên sinh động, điển hình là các hiệu ứng Paralax, Landing Page, ...

1. **Cách sử dụng.**
2. Ba kiểu dùng CSS trong trang

-Inline Style Sheet(CSS cục bộ): là cách viết CSS trên chính thẻ chúng ta cần định dạng. Bắt đầu bằng từ khoá style.

VD:

<p style="color:sienna;margin-left:20px;">This is a paragraph.</p>

-Internal Style Sheet(CSS nội tuyến): là cách viết CSS trong cặp thẻ style thường đặt trong phần head của trang.

VD:

<head>

<style>

hr {color:red}

p{margin-left:20px;}

body {background-image: url("images/background.gif");}

</style>

</head>

-External Style Sheet(CSS ngoại tuyến): là cách dùng CSS từ file ở ngoài. Người dùng có thể tái sử dụng file CSS ngoài đó cho các trang khác.

VD:

<head>

<link rel=”stylesheet” type=”text/css” href=”css/style.css”>

</head>

1. Sử dụng Selector
   1. Định nghĩa Selector:

Selector là định nghĩa các cách truy vấn các thẻ HTML. Dùng trong Internal Style Sheet và External Style Sheet

* 1. Các dạng selector:

-Normal:

+ Selector tất cả: loại selector này sẽ tác động đến tất cả các thẻ html trong trang.

Cú pháp:

\*{

property: value;

}

+Selector theo thẻ xác định: loại selector này sẽ tác động lên những thẻ có tên đó.

Cú pháp:

tagName{

property: value;

}

VD: p{

color:red;

}

+Selector theo class: loại selector này sẽ chỉ tác động lên class.

Cú pháp:

.ClassName{

property: value;

}

Vd: ta có thẻ

<div class="banner">Thẻ div</div> có class banner.

Giờ ta muốn style nó chỉ cần:

.banner{

color: red;

}

+Selector theo ID:

Cú pháp:

#id{

property: value;

}

VD:

ta có thẻ

<div id="banner">Thẻ div</div> có id banner.

Giờ ta muốn style nó chỉ cần:

#banner{

color: red;

}

+Selector phân cấp: loại CSS này sẽ cho chúng ta truy cập vào con của thẻ cha.

Cú pháp:

cha con chau... n{

property: value;

}

VD: ta muốn style thẻ p trong thẻ div

<div>

<p>Thẻ P bên trong thẻ div</p>

</div>

Thì ta sẽ cần khai báo như sau:

div p{

color: red;

}

+Selector nhiều thẻ: cho phép định dạng nhiều thẻ cùng một lúc.

Cú pháp:

tag1, tag2 ,..., tagn{

/\*\*/

}

VD:

.banner, #banner,p{

Color:red;

}

* Pseudo:

Pseudo Element (có thể gọi là các phần tử giả) được sử dụng để thêm các hiệu ứng đặc biệt cho một số Selector mà không cần sử dụng JavaScript hoặc bất cứ ngôn ngữ Script nào để sử dụng các hiệu ứng này.

Cú pháp:

selector:pseudo-element {property: value}

Các Class trong CSS cũng có thể được sử dụng với Pseudo Element như sau:

selector.class:pseudo-element {property: value}

Dưới đây là một số cách sử dụng phổ biến của Pseudo Element:

|  |  |
| --- | --- |
| **Giá trị** | **Miêu tả** |
| :first-line | Sử dụng phần tử này để thêm Style đặc biệt cho dòng đầu tiên của đoạn văn trong một Selector |
| :first-letter | Sử dụng phần tử này để thêm Style đặc biệt cho chữ cái đầu tiên của đoạn văn trong một Selector |
| :before | Sử dụng phần tử này để chèn nội dung nào đó vào trước một phần tử |
| :after | Sử dụng phần tử này để chèn nội dung nào đó vào sau một phần tử |

VD: style cho dòng đầu tiên của đoạn văn trong thẻ p

p:first-line { text-decoration: underline; }

-Attribute: là loại selector theo thuộc tính của thẻ.

+ Theo thuộc tính:

tag[attribute] {

background-color: yellow;

}

VD:

HTML:

<a href="http://www.disney.com" target="\_blank">disney.com</a>

<a href="http://www.wikipedia.org" target="\_top">wikipedia.org</a>

CSS:

a[target] {

background-color: yellow;

}

+ Theo thuộc tính có giá trị xác định:

Cú pháp:

tag[attribute="value "] {

background-color: yellow;

}

VD:

HTML:

<a href="http://www.disney.com" target="\_blank">disney.com</a>

< a href="http://www.wikipedia.org" target="\_top">wikipedia.org</a>

CSS:

a[target=”\_blank”] {

background-color: yellow;

}

+ Theo thuộc tính bắt đầu bằng giá trị hoặc bắt đầu bằng giá trị được tách ra bởi dấu cách:

Cú pháp:

tag[attribute~="value "] {

background-color: yellow;

}

VD:

HTML:

<img src="klematis.jpg" title="klematis flower" width="150" height="113">

<img src="img\_flwr.gif" title="flower" width="224" height="162">

CSS:

[title~=flower] {

border: 5px solid yellow;

}

+ Thuộc tính bắt đầu bằng giá trị hoặc bắt đầu bằng giá trị theo sau là dấu ‘-‘ :

Cú pháp:

[attribute|=value] {

background: yellow;

}

VD:

HTML:

<h1 class="top-header">Welcome</h1>

<p class="toptext">Hello world!</p>

CSS:

[class|=top] {

background: yellow;

}

+ Thuộc tính bắt đầu bằng giá trị:

Cú pháp:

[attribute^=value] {

background: yellow;

}

VD:

HTML:

<h1 class="top-header">Welcome</h1>

<p class="toptext">Hello world!</p>

CSS:

[class^=top] {

background: yellow;

}

+ Thuộc tính kết thúc bằng giá trị:

Cú pháp:

[attribute$=value] {

background: yellow;

}

VD:

HTML:

<h1 class="header-top">Welcome</h1>

<p class="texttop">Hello world!</p>

CSS:

[class$=top] {

background: yellow;

}

+ Thuộc tính có giá trị ở bất cứ chỗ nào:

Cú pháp:

[attribute\*=value] {

background: yellow;

}

VD:

HTML:

<h1 class="header-tops">Welcome</h1>

<p class="texttoptttt">Hello world!</p>

CSS:

[class\*=top] {

background: yellow;

}

1. Sử dụng các thuộc tính cơ bản trong CSS

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Ý nghĩa | Cách sử dụng |
| font-family | Chỉnh font văn bản | Cú pháp:  font-family: tên-font, tên-font-backup, tên-font-backup,...;  VD: font-family: 'Times New Roman', Times, serif; |
| font-size | Chỉnh cỡ chữ văn bản | Cú pháp:  font-size: đơn vị;  VD: font-size: 16px; |
| font-style | Chỉnh kiểu chữ | Cú pháp:  font-style: kiểu chữ;  VD: font-style: italic; |
| font-weight | Chỉnh sửa độ dày chữ | Cú pháp:  font-weight: độ dày;  VD: font-weight: bold; |
| width | Điều chỉnh độ rộng của thẻ | Cú pháp:  width: đơn vị kích thước;  VD: width: 100px; |
| height | Điều chỉnh chiều cao của thẻ | Cú pháp:  height: đơn vị kích thước;  VD: height: 100px; |
| letter-spacing | Giãn khoảng cách các kí tự trong văn bản | Cú pháp:  letter-spacing: đơn vị;  VD: letter-spacing:100px; |
| line-height | Xác định chiều cao của 1 dòng | Cú pháp:  line-height: đơn vị;  VD: line-height:100px; |
| text-tranform | Sửa đổi định dạng chữ | Cú pháp:  text-tranform: kiểu chữ;  VD: text-tranform:uppercase; |
| text-decoration | Trang trí cho văn bản | Cú pháp:  text-decoration:cách trang trí;  VD: text-decoration: underline; |
| word-spacing | Giãn khoảng cách các từ trong văn bản | Cú pháp:  word-spacing: đơn vị;  VD: word-spacing:100px; |
| background | Thay nền cho thẻ | Cú pháp:  background:màu;  VD: background:blue; |
| color | Chỉnh sửa màu chữ thẻ | Cú pháp:  color:màu;  VD: color:blue; |
| margin | Cách các thành phần bên ngoài thẻ | Cú pháp:  margin:trên dưới trái phải;  VD: margin:10px 10px 10px 10px; |
| padding | Các thành phần bên trong thẻ cách thẻ chứa chúng | Cú pháp:  padding:trên dưới trái phải;  VD: padding:10px 10px 10px 10px; |
| border | Tạo đường viền trong cho thẻ | Cú pháp:  border:kiểuviền kích thướcviền;  VD: border:solid 10px; |
| outline | Tạo đường viền ngoài cho thẻ | Cú pháp:  outline:kiểuviền kích thướcviền;  VD: outline:solid 10px; |
| float | Làm nổi thẻ sang trái hoặc sang phải | Cú pháp:  float:hướng;  VD: float:right; |
| list-style | Xác định kiểu cho 1 danh sách(ul,ol) | Cú pháp:  list-style:kiểu;  VD: list-style: upper-latin; |
| opacity | Mức độ trong suốt của thẻ | Cú pháp:  opacity:giá trị;  Giá trị từ float từ 0->1  Càng gần 0 thì càng mờ  và ngược lại  VD: opacity:0.5; |
| visibility | Ẩn thẻ hoặc hiện thẻ | Cú pháp:  visibility:trạng thái;  Trạng thái:visible là hiện  Trạng thái:hidden là ẩn  VD: visibility:hidden; |

1. Các thuộc tính nâng cao trong CSS

|  |  |
| --- | --- |
| Thuộc tính | Ý nghĩa |
| position: fixed | Định vị trí tương đối cho thành phần theo cửa sổ trình duyệt. |
| position: relative; | Định vị trí tuyệt đối cho thành phần. |
| position: absolute; | Định vị trí tuyệt đối cho thành phần theo thành phần bao ngoài (thành phần định vị trí tương đối position: relative;) hoặc theo cửa sổ trình duyệt. |
| display:block | Thành phần hiển thị như một khối, khi sử dụng giá trị block thành phần sẽ đứng một hàng độc lập so với thành phần trước và sau nó. |
| display:inline | Thành phần sẽ hiển thị như một nội tuyến (inline, không ngắt dòng) |
| display:inline-block | Thành phần sẽ hiển thị như một khối, nhưng là một khối nội tuyến. |
| display:flex | Hiển thị dàn hàng cho một danh sách các phần tử nằm trong một container. |
| flex-direction | Chuyển trục hiển thị cho flex  Có 4 kiểu thường dùng:  -flex-direction :row Chuyển trục main axis thành chiều ngang, nghĩa là hiển thị theo hàng.  -flex-direction :colum Chuyển trục main axis thành chiều dọc, nghĩa là hiển thị theo cột.  -flex-direction :row-reverse Hiển thị theo hàng nhưng đảo ngược vị trí các item.  - flex-direction :column-reverse Hiển thị theo cột nhưng đảo ngược vị trí các item. |
| flex-wrap:wrap | Thuộc tính này cho phép container có thể bọc lại các item kể cả khi kích thước của item bị thay đổi, mặc định là nowrap. Thuộc tính này có thể áp dụng với cả chiều dọc của container và item |
| flex-flow: flex-direction|| flex-wrap | Thuộc tính này là một dạng viết tắt (shorthand) cho hai thuộc tính flex-direction và flex-wrap.  VD: flex-flow:column ( column trong flex-direction) |
| justify-content trong flex | Thuộc tính justify-content sắp xếp các mục linh hoạt theo trục chính của dòng hiện tại trong khung linh hoạt. Nó giúp bổ sung không gian còn thừa ngay cả khi các mục linh hoạt trên một dòng không thể co giãn hoặc đã đạt đến kích thước tối đa.  5 loại:  -justify-content:flex-start; Các mục linh hoạt được sắp xếp ở bên trái của khung linh hoạt theo dạng ltr  -justify-content:flex-end; Các mục linh hoạt được sắp xếp ở bên phải của khung linh hoạt theo dạng ltr  -justify-content:center; Các mục linh hoạt được sắp xếp ở giữa khung linh hoạt  -justify-content:space-between; Các mục linh hoạt sẽ được hiển thị với khoảng cách tương đương nhau ở giữa chúng, mục linh hoạt đầu tiên và cuối cùng được xếp vào 2 bên của khung linh hoạt  -justify-content:space-around; Các mục linh hoạt được hiển thị với khoảng cách bằng nhau xung quanh mọi mục linh hoạt, kể cả mục linh hoạt đầu tiên và cuối cùng |
| background gradient | tạo màu sắc cho background theo biên độ giảm dần. |
| text-shadow | Xác định bóng đổ cho chữ. |
| box-shadow | Xác định bóng đổ cho một khung. |
| transform | chuyển đổi 2 chiều, 3 chiều, có thể là xoay, tỷ lệ, di chuyển, nghiêng, ... cho thành phần. |
| !important | Xác định quyền ưu tiên cho các thuộc tính css.  Nếu thuộc tính có !important thì nó sẽ được dùng khi mà cũng với thuộc tính đó ở một cách dùng css khác(inline,internal,external) thì những cách này sẽ không được dùng.  VD: ta dùng css inline có !important thì các cách như internal và external chúng ta sử dụng sẽ không được dùng. |
| @keyframes | dùng để điều khiển diễn biến một hoạt động của thành phần, được dùng kèm với thuộc tính animation. |
| animation-bounce | Tạo hiệu ứng nhảy lên xuống cho thành phần. Sử dụng tranform translate |
| animation-fade | Tạo hiệu ứng làm mờ. Sử dụng opacity |
| animation-slide | Tạo hiệu ứng trượt. Sử dụng position |
| animation-zoom | Tạo hiệu ứng phóng to thu nhỏ thành phần. Sử dụng scale |
|  |  |
|  |  |