Html:

|  |  |
| --- | --- |
| Index.html |  |
| <!DOCTYPE html> | Thẻ (tag) |
| <html> |
| <head> |
| <body> |
| <h1> | Heading, tiêu đề |
| <p> | Paragrap |
| <b> | In đậm |
| <i> | In nghiêng |
| <img>  Src = “ “  Alt = “ “ (Không bắt buộc)  Width = “ 100%“ | Hình ảnh  Link nguồn  Đánh dấu tên ảnh để sửa lỗi  Độ rộng |
| <a href = “ (Link) “ > (Nội dung) </a> | Neo liên kết |
| <ul>  <li> \* 4  <ol> | Danh sách dấu .  Danh sách theo số |
| <table>  <thead>  <th>  <tbody>  <td>  (<th> và <td> số lượng bằng nhau) | Bảng |
| <input>  Type = “ text “  = “ checkbox “  = “ radio “  Name = “\_“ Type = “ radio “  (Phải cùng tên name) | Để cho người dùng nhập  Chữ  Dấu tích vuông  Dấu tích tròn  Dấu tích tròn nhưng chỉ 1 cái |
| <button> | Nút |
| <div>  (Nội dung code) | Khối bao quanh nội dung |
| <hr/> | Gạch ngang dài xuyên trang |
| Attribute  <h1 title = “\_”> \_</h1> | Thuộc tính (Attribute)  Vị trí thuộc tính |
|  |  |
| CSS trong HTML | |
| <style>  h1 { color: red;  font-size: 20px; } | Internal |
| <link rel=”stylesheet” href = “ (Tên file).css ”  (Tạo file css internal ở ngoài) | External |
| <h1 style =” color: red; font-size: 16px> | Inline |
| <h1 id = “first-heading”>  #first-heading { color: red; } | ID  (Chỉ 1 id cùng tên, không nên trùng nhau, dễ lỗi) |
| <h1 class = “second-heading”>  .second-heading { color: green; }  .Tên1 .Tên2 | Class  (Khắc phục lỗi trên id, cho phép cùng tên)  Chọn class Tên2 là con của Tên1 |
| Thứ tự ưu tiên trong CSS   |  |  |  |  | | --- | --- | --- | --- | | 0. !important - ∞ |  | | | | 1. Internal, External |  | | | | 2. Inline - 1000 |  | | | | 3. #id - 100 |  | | | | 4. .class - 10 |  | | | | 5. tag - 1 |  | | | | 6..Equal specificity | Mới hơn | | | | 7. Universal selecter and inherited - 0 | Universal selecter | \* {\_} | Dùng cho tất cả | | inherited | li {\_} | Vd chỉ riêng phần list | | |
| -- Tên: red  Color: var(--Tên);  Global: Thường nằm trên cùng, dùng chung  Local: Nằm trong 1 thẻ, sử dụng trong 1 thẻ | Biến (Variable) |
| Đơn vị trong CSS | |
|  | px |
|  | pt |
|  | cm |
|  | mm |
|  | inch |
|  | pc |
|  |  |
| Phụ thuộc vào thẻ chứa nó | % |
| Phụ thuộc vào html {font-size: 40px;}  Mặc định font-size html là 16px  Font-size 62,5% = 10px | rem |
| Phụ thuộc vào thẻ gần nhất từ trong ra ngoài | em |
| Phụ thuộc vào kích thước trình duyệt | vw |
| vh |
|  | vmin |
|  | vmax |
|  | ex |
|  | ch |
| \* {  Margin: 0;  Padding: 0;  } | margin (khoảng cách giữa các thẻ) |
| padding |
| Một số hàm trong CSS ( CSS funtions) | |
| var() |  |
| linear-gradient() |  |
| rgba() |  |
| rgb()  Rgb( 255,255,255) |  |
| calc() |  |
| attr()  .box: :after { content: attr(data); }  <div class=”box” data=“Đây là box hồng” | clor |
| Lớp giả trong CSS (CSS pseudo-classes) | |
| :root | Tham chiếu tới thẻ <html> |
| :hover  .bix:hover{}  h1:hover{} | Đổi thuộc tính khi di chuột vào |
| :active | Đổi thuộc tính khi bấm chuột vào |
| :first-child | Chọn thẻ con đầu |
| :last-child | Chọn thẻ con cuối |
| Phần tử giả trong CSS (CSS pseudo-elements) | |
| ::before  .box::before{  content: “ “;  display: block;  width: 50px;  height: 5-px;  background-color: #fff; | before và after chỉ có thể có một và luôn luôn tồn tại đầu cuối. |
| ::after |
| ::first-letter |  |
| ::first-line |  |
| ::selection |  |
|  | |
| padding-top  padding-left  padding-right  padding-bottom  🡪 padding: trên phải dưới trái | padding |
| border-width  border-style  border-color  🡪 border: 10px solid #333 ?  (bị nhầm với border gạch chân) | border |
| margin-top  margin -left  margin -right  margin -bottom | margin |
| float | Box trình bày theo bên trái |
| border-radius | Bo tròn góc |
| text-de coration: none | Bỏ gạch chân măc định thẻ a (thẻ link)b |
| box-sizing: border-box  box-sizing: content-box  box-sizing: unset | Tổng kích thước content, padding, border.  mặc định  huỷ |
| background | |
| background-image:  url(\_) , url(\_)  linear-gradient( góc độ, màu 1, màu2)  background-size: 100%, (hoặc auto)  background-repeat: no-repeat  repeat-x : chiều ngang  repeat-y : chiều dọc  background-size: contant  background-size: cover  box-sizing: border-box  background-origin: padding-box  background-position:  top top  10px 10px  top 10px left -20px | lấy vừa màn hình  không lấy khoảng trắng |
| background:  màu  center  url(\_) no-repeat  /contain  gõ lệnh, Chọn mục MDN Reference để chuyển tới web tìm hiều về thuộc tính. | Rút gọn |
| position | |
| relative | Lấy thẻ đó làm gốc toạ độ |
| absolute | Lấy thẻ cha gần nhất có thuộc tính position làm gốc toạ độ |
| fixed | Cố định khi kéo web lên xuống  Dùng margin để những thẻ khác không bị che mất |
| sticky (ít dùng do trình duyệt ít hỗ trợ) |  |
| top: 10  bottom: 10  left: 10  right: 10 | phủ nhanh bề mặt thẻ cách viền 10 px |
| Chữ chính giữa | |
| text-align: center  (có thể dùng cho thẻ cha để áp dụng tất cả thẻ con)  line-height: 100px |  |
| display: flex  (align-itiems: center & justify-content: center)  hoặc margin: auto |  |
| position: absolute  top: 50%  transform: translateY(-50%)  top: 50%  left: 50%  transform: translate(-50%, -50%) |  |
|  |  |
| Fallback image (ảnh dự phòng khi lỗi) | |
| Placeholder image  placehold.it/500x500 | Nguồn |
| onerror = ” this.scr ‘/img/\_‘ | Nên dùng vì có liên quan tới javascript |
| background-image: url(‘\_’), url(‘\_’), url(‘\_’), |  |
| Những thành phần thường gặp trên giao diện website | |
| 1. Header (đầu trang)      1. Navigation (điều hướng)      1. Breadcrumb      1. Sidebar      1. Slider      1. Content (nội dung chính) 2. Banner (ảnh bìa, ảnh quáng cáo) 3. Footer (chân trang) | |
| Một số thủ thuật bấm nhanh | |
| ! tab | Tạo dàn bài nhanh |
| viết tắt tab | Gõ một số lệnh nhanh |
| .tên  #tên | Tạo class nhanh  Tạo id nhanh |
| bôi đậm, crtl-D | Chọn nhanh nhiều mục giống nhau |
| ctrl-/? | Tạo ghi chú nhanh  (chọn cái tiếp theo) |
| ctrl-L | Tạo ghi chú nhanh  (chọn hết) |
| chọn in=”  ctr-L, mũi tên phải  alt-shift-mũi tên phải  ctr-c | Copy tất cả các tên id, class khác nhau |
| chọn trỏ chuột trc main  giữ alt chọn các dòng khác  gõ #  bấm end  gõ { } | đồng điệu cùng một lúc các thẻ |
| ul.subnav>li\*3>a | Tạo nhanh nhiều mục cùng lúc |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| JavaScript | |
| setTimeout (funtion(){}, Thời gian- mili giây ) | Thực thi lệnh sau 1s |
| setInterval (funtion(){}, Thời gian- mili giây ) | Thực thi lệnh sau 1s nhưng lặp lại vô số lần |
| var myObject = {  name: ‘Men’,  age: ‘24’,  myFuntion: funtion () {}  }  console.log(‘myObject’, myObject); | In ra thông tin danh sách |
| var myArray = [  Men  24  ] | In ra thông tin danh sách có đánh số |
| 0  ‘ ‘  null  underfined  NaN  false |  |
| var result = ‘A’ && ‘B’  var result = ‘A’ || ‘B’ | result B  result A |
| target.addEventListener(event, function, useCapture);  btn.addEventListener("click", xinChao); | Cho phép thực thi nhiều event cùng lúc mà ko bị ghi đè funtion |
| document.getElementById("id").innerHTML = target  let target = document.getElementById("id") | Thay thế chữ trong id |
| onclick="inputValue(this.value) | Điền ngay vào ô input giá trị bấm nút |
| var x = Math.round(Math.random()\*window.innerWidth);  var y = Math.round(Math.random()\*window.innerHeight); | Toạ độ các vị trí ngẫu nhiên |
|  |  |
| Trả về kiểu dữ liệu số | |
| + |  |
| Math.ceil | Làm tròn 1 số lên |
| ceil  -------  real  -------  floor | 5  -------  4.5  -------  4 |
| Math.floor | Làm tròn 1 số xuống |
| Math.round | Làm tròn gần nhất |
| Math.trunc | Làm tròn lấy phần thân |
| Math.pow | Luỹ thừa |
| Math.sqrt | Căn bậc 2 |
| parseInt | Trả về số nguyên (chỉ lấy số trước dấu , ) |
|  |  |
| Mảng ( Array ) | |
| .push() | Thêm pt cuối mảng |
| .unshift() | Thêm pt đầu mảng |
| .pop() | Xoá 1 pt cuối mảng |
| .shift() | Thêm pt đầu mảng |
| .sort() | Sắp xếp mảng theo 1 trật tự |
| .reverse() | Đảo ngược hàm |
| replace()  string.replace(“Ký tự cần thay thế ”, “Ký tự thay vào)  string.replace(/[Ký tự cần thay thế]/g, “Ký tự thay vào) | Thay thế ký tự trong chuỗi  Thay thế toàn bộ 1 ký tự |
| .splice(vị trí bắt đầu xoá, số pt muốn xoá)  .splice(vị trí bắt đầu xoá, 0, ‘phần tử mới’) | Xoá pt  Thêm pt |
| concat()  mảng mới = mảng 1.concat(mảng 2,mảng 3,…mảng n) | Nối n mảng |
| indexOf(‘phần tử’ , vị trí bắt đầu tìm kiếm);  nếu không có thì trả về -1 |  |
| let a = “codegym.vn”  let b = a.indexOf(“g”);  //4 | Lấy ra vị trí của ký tự trong chuỗi |
| let a = [1, 4, 5, 2, 3]  let b = a.indexOf(“5”);  //3 | Lấy ra vị trí của pt trong mảng |
| .map() {} | Thay đổi giá trị các thuộc tính của đối tượng trong mảng |
| .join() | Chuyển tất cả các pt trong mảng thành chuỗi và nối chúng lại với nhau |
| .toString() | Chuyển toàn bộ các phần tử trong mảng sang dạng chuỗi |
| var matrix = new Array(10).fill(new Array(10));  for (var row = 0; row < matrix.length; row++) {  for (var column = 0; column < matrix[row].length; column++) {  matrix[row][column] = Math.floor((Math.random() \* 100) + 1);  }  } | Khởi tạo giá trị ngẫu nhiên của mảng |
| for (var row = 0; row < matrix.length; row++) {  for (var column = 0; column < matrix[row].length; column++) {  console.log(matrix[row][column] + " ");  }  } | Hiển thị các phần tử của mảng |
| var total = 0; for (var row = 0; row < matrix.length; row++) {  for (var column = 0; column < matrix[row].length; column++) {  total += matrix[row][column];  }  } | Tính tổng các phần tử số |
| for (var column = 0; column < matrix[0].length; column++) {  var total = 0;  for (var row = 0; row < matrix.length; row++) ;  total += matrix[row][column];  console.log("Sum for column " + column + " is " + total);  } | Tính tổng các phần tử số theo từng cột |
| var maxRow = 0;  var indexOfMaxRow = 0;  for (var column = 0; column < matrix[0].length; column++) { maxRow += matrix[0][column];  }  for (var row = 1; row < matrix.length; row++) {  var totalOfThisRow = 0;  for (var column = 0; column < matrix[row].length; column++) totalOfThisRow += matrix[row][column];  if (totalOfThisRow > maxRow) {  maxRow = totalOfThisRow; indexOfMaxRow = row;  }  }  console.log("Row " + indexOfMaxRow + " has the maximum sum of " + maxRow) | Tìm hàng có tổng lớn nhất |
| for (var i = 0; i < matrix.length; i++) {  for (var j = 0; j < matrix[i].length; j++) {  var i1 = (int)(Math.random() \* matrix.length);  var j1 = (int)(Math.random() \* matrix[i].length); var temp = matrix[i][j]; matrix[i][j] = matrix[i1][j1]; matrix[i1][j1] = temp;  }  } | Trộn ngẫu nhiên các phần tử |
| .fill(value, start, end) | Thay đổi giá trị của tất cả các phần tử trong mảng thành một giá trị mới. |
| .charAt(vị trí) | Lấy ra ký tự ở vị trí |
| string.replace(searchvalue, newvalue) | Thay thế ký tự 1 thành ký tự 2 |
|  |  |