**HTML – CSS – JAVASCRIPT**

**HTML**

* Các thẻ thông dụng trong HTML:
  + h1->h6 (heading): Tiêu đề
  + p (paragraph): Đoạn văn
  + img (image): ảnh có 2 thuộc tính hiển thị là src = ‘đường link’, alt=’chú thích khi ảnh không hiển thị’.
  + a (anchor): link tới, có thuộc tính href=”link liên kết”, nội dung
  + ul – li (unordered list – danh sách): danh sách A black background with white text

    Description automatically generatedA close-up of a text

    Description automatically generated
  + table: bảng , bao gồm 2 thẻ chính bênh trong là thead và tbodyA screen shot of a computer program

    Description automatically generatedA close up of black text

    Description automatically generated
  + input: có type có nhiều loại như (text, checkbox, radio, ...) A white rectangle with black lines

    Description automatically generatedA black background with blue and white text

    Description automatically generatedA group of icons with check marks

    Description automatically generatedA screen shot of text

    Description automatically generatedA close up of a screen

    Description automatically generated
  + button: nút bấm
  + div: khối, giống bố cục view

**CSS**

* Có 3 loại: Internal, External, Inline
  + Internal: sử dụng một cặp thẻ style đặt trong file html và viết css trong cặp thẻ style đó.

A screen shot of a computer program

Description automatically generated

* + External: Tạo ra 1 file.css bên ngoài rồi link vào bằng thẻ link A screen shot of a computer program

    Description automatically generated
  + Inline: Viết trong dòng luôn
* ID và class trong CSS: đặt id và class để định danh cái mà chúng ta muốn css, với id chúng ta css truy xuất External với dấu #, còn class là dấu . A black background with text

  Description automatically generatedA screen shot of a computer program

  Description automatically generatedA white background with green text

  Description automatically generated

A screenshot of a computer

Description automatically generated

* Độ ưu tiên trong CSS:

A white background with red text

Description automatically generated

* + Đối với số 1 thì cái nào đc viết mới hơn, viết trước thì đc ưu tiên
  + Đối với 2 3 4 5 thì dựa vào số điểm cái nào lớn hơn thì ưu tiên
  + Đối với 6 thì nếu 2 thg giống nhau thì cái nào mới sẽ ưu tiên
  + Đối với số 7 là 0 điểm, nếu ko có cái nào ưu tiên hơn thì sẽ là nó
* Đặt biến trong CSS: Có 2 dạng cục bộ và toàn cục A computer screen shot of a program code

  Description automatically generated
* Đơn vị trong CSS: đơn vị tương đối và đơn vị tuyệt đối, 100% = 16px
  + Tuyệt đối (Absolute): px, pt, cm, mm, inch, pc nó là cứng và cố định, không bị thay đổi theo tỷ lệ màn hình khác nhau.
  + Tương đối (Relative): %, rem, em, vw, vh ..... đơn vị tương đối thì linh động theo tỷ lệ đối với đon vị này thì cần có 1 cái để nó phụ thuộc vào để link động.
    - %: phụ thuộc zo cái thẻ chứa nó
    - Rem: phụ thuộc vào thuộc tính đc định nghĩa ở thẻ html, vd h1 phụ thuộc zo html thì 1 rem = 40px.A screen shot of a computer code

      Description automatically generated
    - Em: phụ thuộc vào thể gần nhất chứa nó và có thuộc tính mà nó cần gán đơn vị em.
    - Vw-vh: A black background with green text

      Description automatically generated lần lượt là % chiều rộng và chiều cao của trình duyệt. Khác với % là không phụ thuộc zo thẻ chứa nó mà phụ thuộc zo kích thước của trình duyệt.
* Một số hàm trong CSS như: var(), rgb(), rgba() có thể nhìn xuyên qua, calc(), linear-gradient(), attr().

A white background with black text

Description automatically generated

* Thẻ <br/> để xuống dòng: 
* Border và padding đều làm tăng kích thước tổng (trên – phải- dưới - trái)A screenshot of a computer

  Description automatically generated
* Margin được đẩy ra từ cái element, nó không làm tăng kích thước của element mà ta margin.
* Thuộc tính box-sizing: sẽ giúp ta khi thêm padding, border vào mà không làm thay đổi kích thước, nó sẽ tự động tính toán để đảm bảo kích thức ban đầu.

A screen shot of a computer code

Description automatically generatedA purple square with white text

Description automatically generated

* Background-image: ảnh nền, có thể có nhiều ảnh nề thì thêm url cách nhau dấu phẩy, nó tự động repeat nên phải thêm thuộc tính background-repeat: no-repeat, background-size thì đầu tiên là chiều rộng, chiều cao thì auto. Ngoài ra cũng có thể thêm background gradientt với màu trong suốt như hình bên dưới.A computer screen with text

  Description automatically generated A screenshot of a phone

  Description automatically generated
* Thuộc tính background-size với cover và contant: cover thì kéo to thu nhỏ trình duyệt vẫn đảm báo cái ảnh co giãn. Contain thì chỉ lấy cạnh dài nhất.
* Thuộc tính background-origin: sử dụng đi kèm với background-image, và nó có ba giá trị là paading-box, border-box, content-box .... nó sẽ đổ hình ảnh từ vị trí padding, border hay content tương ứng.
* Thuộc tính background-position: dùng để cố định vị trí của hình ảnh, top right, top left, top center .... có thể để 1 cái thì cái còn lại tự hiểu là center vd top = top center, hoặc có thể top 20px right 20px thì cách trên và phải 20px hoặc ghi 30px 40px thì theo trục x y nó sẽ cách trục x 20 và y 40.
* Cú pháp ngắn cho background:
  + A blue screen with text

    Description automatically generated sẽ tương đương với background: url(), no-repeat, center, contain.
* Thuộc tính position: thiết lập vị trí giữa các element, vd có thể đè lên nhau bla bla. Gồm 4 cái (relative, absolute, Fixed, Sticky).
  + Relative (Tương đối): là vị trí tương đối, nó sẽ lấy chính mình làm gốc tòa độ và không phụ thuộc vào bất cứ đối tượng nào bên ngoài cả.
  + Absolute (Tuyệt đối): là vị trí tuyệt đối và nó phụ thuộc vào thẻ cha gần nhất có thuộc tính positon để làm gốc tòa độ.
  + Fiexed: phụ thuộc vào cửa sổ trình duyệt, nó sẽ cố định và không bị thay đổi khi ta trượt, kéo, lăn, cuộn cửa sổ lên xuống qua lại.
* Căn giữ các item trong CSS:
  + Căn giữ theo chiều ngang ta dùng (text-align: center) cho thẻ cha thì tất cả thẻ con bên trong sẽ center, còn để trong thẻ con thì thẻ căn giữa theo thẻ cha chứa nó.
  + Căn giữa theo chiều dọc thì dùng (line-height: chiều cao của thẻ chứa nó) thì sẽ căn giữa, để ở thể cha hay con đều được.

A screen shot of a computer code

Description automatically generated

* + Cách khác đưa vào thể cha (display: flex) và thể con đưa vào (margin: auto) như hình, cái này có thể căn giữa nhiều iteam con.

A computer screen shot of text

Description automatically generated

* + Cách khác nữa là dùng (align-items: center) căn giữa theo chiều cao và thêm (jusstify-content: center) căn giữa theo chiều ngang. Cách này linh động hơn, có thể thay đổi giá trị ở justify-content.

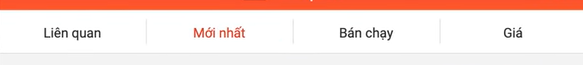
A screen shot of a computer program

Description automatically generated

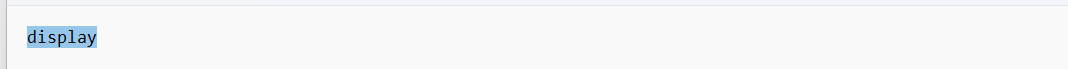
* + Cách khác nữa:

A screen shot of a computer program

Description automatically generated

* + Ví dụ làm dấu gạch đứng ngăn cách như thế này 
  + A screen shot of a computer screen

    Description automatically generated
* Thuộc tính display: Có 1 số giá trị none, inline, block, inline-block, flex.
  + inline: Nó sẽ chiếm kích thước vừa với nội dung và mình không thể đặt width, height.



* + block: Nó sẽ chiếm kích thước hết một khối, hay một dòng. Không thể đặt width, height.



* + Inline-block: Bth nó sẽ chiếm kích thước vừa với nội dung nhưng khác ở chỗ có thể đặt width, height.
  + Flex:
* Flexbox: bí kíp flexbox (https://codepen.io/enxaneta/full/adLPwv/).A screenshot of a computer program

  Description automatically generated
* Khi display: flex|flex-inline thì mặc định thuộc tính flex-direction: row nên các phần tử sẽ nằm cùng nhau trên một hàng.
* Justify-content: thường gán cho thẻ cha, giúp căn theo chiều ngang, có thể căn giữa, trái, phải, 2 bên.

**Lưu ý với javascript:**

a = 1, b = ‘1’ thì c = a ==b thì c mang giá trị true vì == nó chỉ cho sánh giá trị trả về đều là 1 chứ nó không quan tâm đến kiểu dữ liệu. c = a===b thì c sẽ mang giá trị là false vì === se vừa so sánh giá trị trả về và cả kiểu dữ liệu.

!= và !== cũng tương tự.

3 loại hàm

dele: function ham1() {}

express: var ham1 = function() {}

arrow: tutu nha

chuỗi: để in 1 chuỗi Toi la ‘siêu nhân’ và Day la dau / thì ta phải

var nameString = ‘Toi la /’siêu nhân/’’ và var nameString = ‘Day la dau //’

Mảng: var arr = [‘a’, ‘b’, ‘c’];

arr.join(x): chèn vào giữa các phần tử trong mảng ký tự x, nếu x là ‘’ thì sẽ biến mảng thành string.

arr.pop(): xóa phần tử ở cuối mảng và trả về phần tử đã xóa ở cuối

arr.push(x, y): thêm một hoặc nhiều phần tử x, y ở cuối mảng và trả về độ dài mới của mảng

arr.shift(): xóa phần tử ở đầu mảng và trả về phần tử đã xóa ở đầu

arr.unshift(x, y): thêm một hoặc nhiều phần tử x, y ở đầu mảng và trả về độ dài mới của mảng

arr.splice(): có thể dùng để xóa, thêm, thay thế phần tử

xóa: splice(index bắt đầu, số lượng phần tử muốn xóa)

thêm: splice(index bắt đầu, 0 , phần tử muốn thêm ví dụ ‘Dart’)

thay thế: splice(index bắt đầu, số lượng phần tử muốn xóa, elemenent1 thay thế, element2, element3)

arr1.concat(arr2): nối 2 mảng lại

arr.slice(index bắt đầu cắt mảng, index end), có thể sao chép mảng bằng arr.slice(0), lấy phần tử đầu mảng arr.slice(0, 1), lấy phần tử cuối mảng arr.slice(-1);

Object: thêm một cặp key: value vào object A computer screen with text and numbers

Description automatically generated

Xóa 1 key

A computer screen with text and numbers

Description automatically generated

Object constructor:

A computer screen with text

Description automatically generated

Thêm thuộc tính bên ngoài hàm tạo thông qua từ khóa prototype A computer screen with many colorful text

Description automatically generated

Đối tượng date trong javascript thì getMonth() phải + thêm 1, vì trả về từ 0 🡪11

A screen shot of a computer

Description automatically generated

Toán tử giải 3 chấm ‘...’ chuyển object thành mảng

A screenshot of a computer

Description automatically generated

Đệ quy: phải biết 2 thứ: Điểm dừng, logic xử lý để tạo ra điểm dừng.

Mảng2: chứa 1 số phương thức có tham số truyền vào là một hàm, hàm đc truyền vào dưới dạng tham số thì gọi là CALLBACK. Tất cả phương thức này đều lặp qua từng phần tử bên trong mảng, lặp qua 1 phần tử thì sẽ gọi cái hàm callback truyền vào 1 lần.

A screen shot of a computer program

Description automatically generated

forEach(): để duyệt

A black background with blue text

Description automatically generated

Every(): lặp qua từng phần tử của mảng và gọi hàm tham số truyền vào để kiểm tra cái chúng ta muốn, trả về kiểu boolean.

A black background with colorful text

Description automatically generated

Some(): Giống every nhưng nó chỉ kiểm tra 1 lần gặp là sẽ dừng lại, trả về boolean.

Find(): trả về 1 phần tử trong mảng, nếu tìm ra điều kiện đúng.

Filter(): Giống find() nhưng trả về tất cả phần tử tìm thấy chứ k chỉ 1 thg đầu tiên.

Map(): trả về một mảng và hàm callback được truyền vào ngoài việc trả về tử trong mảng thì có thể trả về thêm index và mảng ban đầu, chỉ cần truyền thêm 2 tham số index và originArr như hình.

A screen shot of a computer program

Description automatically generated

Reduce(callback(arr, value, index, originalArray), initValue): 4 tham số trong hàm callback lần lượt là initValue(biến lưu trữ), curentValue(giá trị hiện tại nó lặp qua), index, originArray.

InitValue phụ thuộc vào kq chúngta mog muốn, nếu kq là số thì InitValue là 0, nếu là 1 mảng thì là mảng rỗng[], đối tượng thì là {}.

**HTML DOM**

* Khi lấy element thì ta dùng: getElementByID, querySelector.
* Khi lấy html collection ta dùng: getElementByClassName, getElementByTagName.
* Khi lấy nodeList: thì ta dùng querySelectorAll.

A screen shot of a computer

Description automatically generated

* Đối với PUT: cũng là cập nhật nhưng là cập nhật ghi đè lên tất cả, còn đối với PATCH chỉ thay đổi những cái thuộc tính cần thay đổi chứ k có ghi đè.

**RESPONSIVE**

* Responsive: là một kỹ thuật giúp tăng trải nghiệm người dùng, giúp cho giao diện người dùng tương thích với nhiều kích cỡ màn hình khác nhau.
  + Đối với mobile là < 724px ~ 45.25 rem
  + Đối với tablet là từ (724px – 1024px) ~ (45.25 rem – 64 rem)
  + Đối với PC, laptop là > 1024px ~ 64rem
* Sử dụng @media và breakpoint để xử lý: A screen shot of a computer program

  Description automatically generated
* Grid system: tự động thay đổi chiều rộng của "grid" với 3 breakpoints., cái wide để tạo ra margin ở 2 bên.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Javascript nâng cao**

* IIFE (Immediately Invoked Function Expression): Hàm được gọi ngay lập tức:

A screenshot of a computer

Description automatically generatedA computer screen shot of text

Description automatically generated

* Scope (Phạm vi): A screenshot of a computer

  Description automatically generated
* Closure (Bao đóng): A computer screen with white text

  Description automatically generatedA screen shot of a computer program

  Description automatically generated
* Hosting (Đưa lên đầu): Output là: “I have $NaN”A screen shot of a computer program

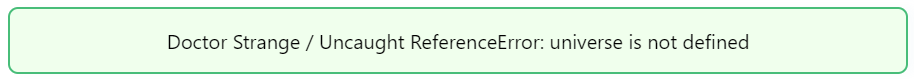
  Description automatically generated A screenshot of a phone

  Description automatically generated
  + Chú ý một biến đã khai báo rồi nhưng nếu chúng ta cố tình khai báo lại trùng tên thì biến đó sẽ mang giá trị undefind. Undefind mà chúng ta đem đi tính toán sẽ mang giá trị NaN.
* Strict mode: Sử dụng với chuỗi này “use strict” là sử dụng nghiêm ngặt. A screenshot of a phone

  Description automatically generatedA black background with white text

  Description automatically generated
  + Ví dụ:

A screen shot of a computer program

Description automatically generated

* Value type (Tham trị) – Reference type (Tham chiếu): A screenshot of a computer

  Description automatically generated
  + Số 1 là tập hợp ác kiểu dữ liệu nguyên thủy, nhóm tham trị.
  + Số 2 là tập hợp các kiểu dữ liệu phức tạp hơn, nhóm tham chiếu.
* Từ khóa “this” : A screen shot of a black and white text

  Description automatically generated
  + This chính là thằng đang gọi thực hiện phương thức.
* Phương thức bind() của một function giúp ta ràng buộc từ khóa this cho một phương thức khi mà phạm vi của từ khóa this không giống như ta tưởng tượng.

A black background with white text

Description automatically generated

* Phương thức call() là phương thức gọi hàm với phương thức bind bên trong. Đối với bind() thì nó sẽ trả về hàm mới chứ k có gọi hàm. Còn call thì vừa bind() và gọi hàm thực thi luôn. A screenshot of a computer

  Description automatically generated
* Phương thức apply(): Nó y chang như call() nhưng chỉ khác đối số là mảng thay vì đối số bình thường.

A black screen with white text

Description automatically generated

* + Khác nhau giữa bind(), call() và apply():

A screenshot of a computer program

Description automatically generated

* Thư viện redux: Gồm các thành phần như trong hình, tôi sẽ ví dụ về một chức năng THÊM trong ứng dụng mà chúng ta xây dựng sẽ hoạt động theo redux như thế nào.
  + Đầu tiên Action ở đây là hành động THÊM(action), dữ liệu ban đầu chúng ta muốn thêm gọi là **state (dữ liệu trong store là state)** , cả **action** và **state** sẽ đều là tham số đầu vào của hàm reducer(state = initValue, action), action là một callback trả về một object.
  + Trong reducer sẽ thực hiện thêm dữ liệu và xử lý logic code bên trong và trả về state mới sau khi xử lý thêm(dữ liệu mới sau khi thêm hay gọi là Trạng thái mới sau khi thêm). Và nó sẽ đc làm tham số của Store.
  + Và Store sẽ tạo ra các phương thức để gọi để render ra giao diện cho người dùng thông qua phương thức subscribe(tham số đầu vào là một callback chứa hàm render bên trong) sau khi thực hiện action THÊM. Từ giao diện ng dùng có thể thực hiện action tiếp theo thông qua phương thức dispatch(tham số đầu vào là một callback chứa hàm muốn thực hiện hành động tiếp theo).

A screenshot of a computer screen

Description automatically generated