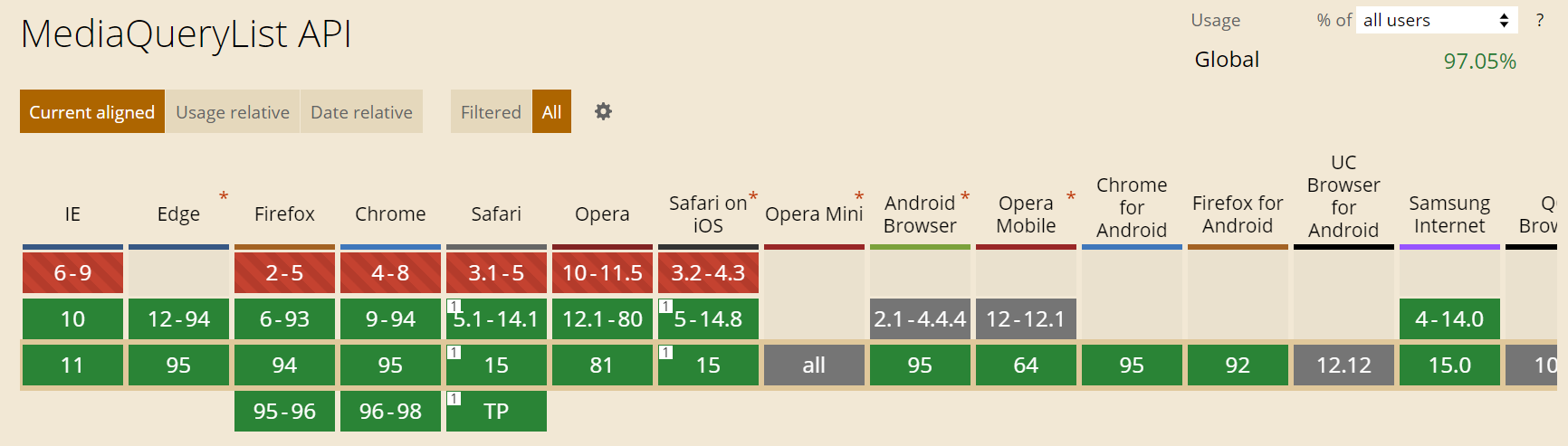
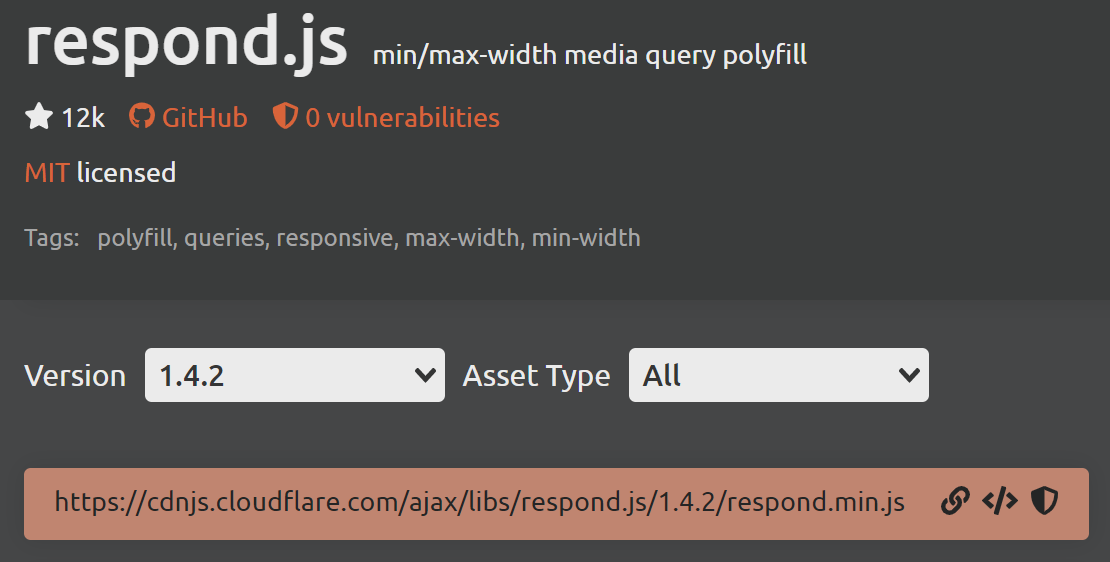
Bản chất của Responsive là gì? Là kỹ thuật giúp website hiển thị tương thích với nhiều kích thước màn hình khác nhau (Mobile, Tablet, PC,…)

<https://caniuse.com/?search=media%20query>



<https://cdnjs.com/libraries/respond.js>



<!--[if lt IE 9]>

<script src="https://cdnjs.cloudflare.com/ajax/libs/respond.js/1.4.2/respond.min.js""></script>

<![endif]-->

/\*

@media not|only mediatype and (mediafeature and|or|not mediafeature){}

1. Keywords:

+ not

+ only

+ and

+ or

2. Mediatypes:

+ print

+ screen

+ speech

+ all- default

3. Media Features:

+ min-width

+ max-width

4. Polyfill?

\*/

/\* Mobile: width <= 740px \*/

/\* 1em = 16px => 739px = 46.1875em \*/

/\* Tablet: width >= 740px and width <= 1024px \*/

/\* 1em = 16px => 740px = 46.25em, 1023px = 63.9375em \*/

/\* PC: width >= 1024px \*/

/\* 1em = 16px => 1024px = 64em \*/

Grid system?

1) Xuất hiện đầu thế kỷ 20

2) Tạo nên các khung nền, hỗ tợ việc sắp xếp bố cục theo trật tự / thống nhất / cân bằng

3) Hệ thống lưới thường gặp:

+ Lưới nhiều cột (Mutilcolumn grid)

+ Lưới một cột (Single column grid)

+ Lưới module (Modular grid)

+ Lưới đường cơ sở (Baseline grid)

4) Vai trò

+ Tổ chức: Có các đường căn giống tiện lợi, dễ dàng sắp xếp các thành phần được ngăn nắp

+ Cân bằng: Dù đối xứng / bất đối xứng, mang lại cái nhìn trực quan, đảm bảo sự cân bằng

+ Tách biệt thành phần: Phân chia nội dung, tạo khoảng cách các thành phần hiệu quả

5) Ứng dụng

+ Lưới trong thiết kế UI UX: Vai trò đặc biệt quan trọng trong Responsive web design

+ Lưới trong in ấn: Google “Grid system”

6) Ứng dụng trong thiết kế web

I Lý thuyết:

1) Cột (Column): Độ rộng sử dụng đơn vị % (tương đối) giúp linh động, dễ dàng tương thích với độ rộng khác nhau của các thiết bị. Số lượng cột trong grid system được xác định trước.

VD: PC 12|16 cột, Table 8 cột, Mobile 4 cột

2) Gutter (Đường ngăn cách) (rãnh ngăn)

Là khoảng cách 2 bên trái / phải của bố cục chính của website. Độ rộng phần lề hay thay đổi để phù hợp với các kích thước màn hình.

VD : PC khoảng cách giữa 2 cột là 12px, Table khoảng cách giữa 2 cột là 8px, Mobile khoảng cách giữa 2 cột là 4px

3) Phần lề (Margin)

Là khoảng cách 2 bên trái/phải của bố cục chính của website. Độ rộng phần lề thay đổi để phù hợp với các kích thước màn hình

II Thành phần chính (làm việc với CSS)

1) Grid – Lưới (Thường là phần cha chứa Row và Column)

- Tạo class

+ grid: full-width, chiếm hết chiều ngang đối tượng chứa (là thằng cha)

+ wide : chiều ngang tối đa 1200px

- Đặt lại chiều rộng trên các thiết bị

/\* Tablet - PC low resolution \*/

@media (min-width: 740px) and (max-width: 1023px) {

  .wide {

    width: 644px;

  }

}

/\* > PC low resolution \*/

@media (min-width: 1024px) and (max-width: 1239px) {

  .wide {

    width: 984px;

  }

}

2) Row – Dòng (Dòng – chiều ngang, chứa Column)

- Chứa các columns, giúp các columns nằm theo chiều ngang

- Khi tổng chiều ngang columns vượt quá kích thước Row, cho columns xuống hàng

- Loại bỏ khoảng thừa do gutters tạo ra ở 2 phía

/\* >= Tablet \*/

@media (min-width: 740px) {

    .row {

        margin-left: -8px;

        margin-right: -8px;

    }

}

/\* PC medium resolution > \*/

@media (min-width: 1113px) {

    .row {

        margin-left: -12px;

        margin-right: -12px;

    }

}

/\* > PC low resolution \*/

@media (min-width: 1024px) and (max-width: 1239px) {

    .wide .row {

        margin-left: -12px;

        margin-right: -12px;

    }

}

3) Column – Cột (Chứa nội dung / thành phần trên website)

* Có mấy cách khai báo CSS?
* + internal
* // viết css trong html luôn
* <style>
* </style>
* + external
* // tạo file index.css
* // trong html tạo đường dẫn đễn file index.css
* <link rel="stylesheet" href="./index.css">
* + inline
* // viết css trong thẻ html luôn
* <div style=""></div>
* Hãy nêu độ ưu tiên trong CSS?

1. Internal, External? ông nào gọi sau thì ưu tiên hơn

2. Inline - 1000 điểm

3. #id - 100 điểm

4. .class - 10 điểm

5. tag - 1 điểm // tag là tên thẻ ví dụ div, h1,..

* Sass với Less giống và khác nhau ở điểm gì?

Về cơ bản thì LESS và Sass đều giống nhau chỉ khác ở một số cú pháp. Biến trong LESS sử dụng ký tự '@' viết đầu, Sass sử dụng ký tự '$'.

- Phân biệt giữa server-side và client-side

- Promise và Async-Await

- Cấu trúc HTML gồm có:

Gồm 3 phần:

+ <!DOCTYPE> dùng để khai báo chuẩn hóa HTML or XHTML

+ <head></head> dùng để khai báo thẻ meta title css javascript

+ <body></body> dùng chứa nội dung trang wed

- So sánh thuộc tính class và id: dùng để đặt tên và phân lội các phân tử mục đích tiện quản lý và định dạng. mục đích của đặt ID là duy nhất, mục đích của dùng class là dùng lại

- Trong css có 1 định nghĩ là selector gống như là 1 con trỏ trong css để chỉ định ra những thẻ mà mình muốn định dạng or tạo kiểu trong html nó có 6 loại elements, sudo elements, class, sudo class, attribute, id style

- Boxmodel là khoảng ko gian phần tử chiếm trong trang web, gồm 4 thành phần chính contex, padding, border, margin

- display block(1 item nằm trên 1 dòng chiếm toàn bộ phần tử), inline (item nằm trên 1 dòng)

- kiểu dữ liệu trong javascript: null, string, number, Boolean, object, undefined

- Phương thức xử lý mảng trong javacript

1. concat() 6. indexOf()

2. filter() 7. Join()

3. find() 8. Map()

4.forEach() 9. Pop()

5. includes() 10. Shift() 11. Push()

- So sách map() và forEach()

+ forEach() sẽ ko trả về giá trị nào

+ map() có trả về giá trị

+ forEach() dùng hàm callback và thay đổi giá trị mảng ban đầu

+ map() không thay đổi mảng ban đầu và trả về một mảng mới

- Những tính năng mới ES6

+ let, const

+ template string

+ destructuring

+ spread

+ rest

+ class

+ promises

|  |  |  |  |
| --- | --- | --- | --- |
|  | const | let | var |
| Global scope | no | no |  |
| Function scope |  |  |  |
| Block scope |  |  | no |
| Can be reassigned | no |  |  |

Reactjs

Props : chỉ có thể đọc, ko thể thay đổi giá trị props

State : giá trị riêng của component, có thể thay đổi giá trị setState khi setState thay đổi thì component render lại