**QUY TẮC**

QUY TẮC CHUNG ĐỂ VIẾT CODE "SẠCH":

- Suy nghĩ đơn giản và viết code đơn giản nhất có thể

- Luôn dọn code sạch hơn lúc ta nhận nó lần đầu, dù là của mình hay của người khác

- Luôn tìm ra nguyên nhân gốc rễ của vấn đề

QUY TẮC ĐẶT TÊN:

- Đặt tên không mơ hồ, mô tả đúng chức năng của định danh

- Không dùng từ viết tắt

- Tên class và tên biến thường dùng danh từ

- Tên hàm thường dùng động từ

- Không nên sử dụng magic number (dùng số trực tiếp trong code) mà nên đặt tên riêng cho hằng số

QUY TẮC VỀ SỐ LƯỢNG:

- Một dòng code không nên dài quá 80 ký tự

- Một câu lệnh nên lồng tối đa 4 cấp

- Một hàm không nên chứa quá 5 tham số

- Một hàm không nên quá 30 dòng

- Một class không nên quá 500 dòng

- Mỗi hàm chỉ nên làm duy nhất 1 việc, nếu thực hiện hai mục đích khác nhau thì tên hàm cần thể hiện rõ

QUY TẮC XUỐNG HÀNG:

- Nếu hàm có nhiều cấp lồng nhau, mỗi cấp nên xuống dòng

- Các đoạn code bằng cấp nên ở cùng 1 cột với nhau, dòng xuống hàng nên bắt đầu cùng cấp với dòng phía trên

- Xuống hàng trước các toán tử (VD: +, -, \*, ?,...)

QUY TẮC COMMENT:

- Hạn chế dufmg comment để giải thích code hoặc chú thích những sự kiện những sự thật hiển nhiên

- Không comment khi đóng thẻ/ngoặc

- Xóa code không dùng, đừng comment out đoạn code đó

- Comment cảnh báo hậu quả

- Comment làm rõ ý nghĩa cỉa code (nếu logic phức tạp)

Install NGROK => tạo địa chỉ https để các thiết bị khác truy cập vào web

1. Download **ngrok** => giản nén file

2. Chọn file đã được giải nén => chuột phải => chọn **Git Bash Here**

3. Làm theo các bước trên trang chủ và lấy mã giác thực (khi login thì trang chủ sẽ hiển thị mã xác thực)

*./ngrok authtoken* *2QVflTEeIngiMJG8NMKhDp7OT0n\_7u3AREmdtTKQCcHpQfEE8*

4. **Git Bash Here** => nhập *./ngrok authtoken [mã xác thực]*

=> Git sẽ hiển thị thông tin truy cập của web:

*web interface https://127.0.0.1:4040*

=> Trong đó: **[4040]** là cổng của server web mặc định của ngrok

**Git Bash Here** => nhập *./ngrok http 5500* => Trong đó: **[5500]** là cổng của server web => xem cổng theo live server trên VScode

Git sẽ hiển thị:

*Forwarding* [*https://651b-27-3-138-5.ngrok-free.app*](https://651b-27-3-138-5.ngrok-free.app) *https://localhost:5500*

Trong đó:

[*https://651b-27-3-138-5.ngrok-free.app*](https://651b-27-3-138-5.ngrok-free.app)

* Đây là địa chỉ truy cập trên các thiết bị khác

**CSS Flexbox**

Các thuộc tính CSS:

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Giá trị | Định nghĩa |
| display-flex | flex | inline-flex | thuộc tính |
| flex-direction | row | colunm | thay đổi phương hướng của axis (phần cha - flex-container) |
| flex-wrap | nowrap | wrap | wrap-reverse | xuống dòng |
| flex-basic | <length> | xét kích thước main size cho flex-item |
| justify-content | flex-start | flex-end | center | space-between | space-around | space-evenly | căn chỉnh flex-item (css flex-container áp dụng cho flex-item) |
| align-content | flex-start | flex-end | center | space-between | space-around | space-evenly | => căn chỉnh flex-item (css flex-container áp dụng cho flex-item)  Trong đó:   * flex-start: Các dòng được dồn về phía trên cùng của hộp chứa. * flex-end: Các dòng được dồn về phía dưới cùng của hộp chứa. * center: Các dòng được dồn về trung tâm của hộp chứa. * space-between: Các dòng được trình bày với khoảng cách bằng nhau giữa chúng. * space-around: Các dòng sẽ được trình bày với khoảng cách bằng nhau xung quanh chúng. * stretch: Các dòng sẽ được kéo dài để phù hợp với hộp chứa. |
| align-items | flex-start | flex-end | center |stratch | baseline | - flex-start: Các phần tử con được căn chỉnh lên trên cùng của container (theo trục chính).  - flex-end: Các phần tử con được căn chỉnh xuống dưới cùng của container (theo trục chính).  - center: Các phần tử con được căn chỉnh vào giữa của container (theo trục chính).  - stretch: Các phần tử con được căn chỉnh để đạt độ cao của container (theo trục chính).  - baseline: Các phần tử con được căn chỉnh theo đường baseline của chúng. |
| align-self | flex-start | flex-end | center | thuộc tính bổ trợ cho flex-item |
| flex-grow | <number> | tỉ lệ phóng to |
| flex-shrink | <number> | tỉ lệ thu nhỏ |
| flex | <number> | viết gộp nhiều thuộc tính  E.x: flex: 1;  => flex-grow: 1;  flex-shrink: 1;  flex-basic: 1; |
| flex-flow | <number> | viết gộp nhiều thuộc tính  Trong đó: flex-direction & flex-wrap  E.x: flex-flow: row wrap;  => flex-direction:row; flex-wrap:wrap; |
| order | <number> | xác định thứ tự của các flex-item |

Trang web tham khảo: [**https://codepen.io/enxaneta/full/adLPwv/**](https://codepen.io/enxaneta/full/adLPwv/)

Nếu muốn rèn luyện thêm và nắm chắc hơn về flexbox thì các bạn có thể tham khảo 3 games sau:

1. <https://codingfantasy.com/games/flexboxadventure/play>

2. <https://flexboxfroggy.com/>

3. <http://www.flexboxdefense.com/>

Use the justify-content property on the tower group container to move your towers into effective positions. justify-content positions a container's items along the main axis and accepts the following values:

* flex-start: group items at the start of a container's main axis
* flex-end: group items at the end of the main axis
* center: group items in the center of the main axis
* space-between: evenly distribute items along the main axis such that the first item aligns at the start and the final item aligns at the end
* space-around: evenly distribute items along the main axis such that all items have equal space around them

Try justify-content: center; for example, to move the container's towers to the center of the main axis.

**BEM**

1. BEM là gì?

=> Block Element Modifier

Trong đó:

Block : khối

Element : thành phần trong khối

Modifier : bổ sung ý nghĩa cho 'Block' hoặc 'Element'

- Là tiêu chuẩn đặt tên class khi viết CSS

2. Tại sao phải dùng BEM:

- Mỗi người một kiểu

- Members dặt class trùng nhau, CSS đè lên nhau

3. Cú pháp

* .block
* .block\_element
* .block--modifier
* .block\_element--modifier

4. Tính ứng dụng

- Xây dựng layout website

- Xây dựng thành phần trên website

|  |  |
| --- | --- |
| 5. Ưu điểm | 6. Nhược điểm |
| Tính rõ ràng | Tên class dài |
| Tái sử dụng dễ dàng | Một số người cho là xấu |
| Giúp cả team làm việc với nhau dễ dàng |  |
| Tính module, không lo CSS của class này ảnh hưởng đến CSS của class khác |  |

7. Khi nào dùng BEM thì phù hợp

- Dự án nhiều members

- Dự án lớn, số lượng pages nhiều hoặc số lượng thành phần trên giao diện nhiều

8. Thực hành

- Làm button

Enabled: Pointer, hover, effect

Disnable: arrow, no effect

- Làm message

- Làm 1 thành phần trên website

9. Trường hợp Block lồng Block

- Block con là thành phần dùng chung

- Block con chứa nhiều element

Khóa học cần hoàn thành trên F8:

1. HTML, CSS từ Zero đến Hero
2. Responsive Với Grid System
3. Lập Trình JavaScript Cơ Bản
4. Lập Trình JavaScript Nâng Cao
5. Xây Dựng Website với ReactJS

**Những việc phải làm tiếp theo là gì?**

1. Viết CV xin việc tại: https://cv.fullstack.edu.vn/, CV được điền sẵn nội dung theo chuẩn xin việc IT, bạn chỉ cần điền thêm thông tin cá nhân và cập nhật lại nội dung cho phù hợp, nên lựa mẫu CV bố cục 1 cột, ít màu mè để NTD dễ đọc hơn. Tham khảo thêm: <https://fb.com/sondnf8/posts/2195606467243484>
2. Ôn tập lại các kiến thức quan trọng, tìm kiếm các bài viết chia sẻ về câu hỏi PV thường gặp, kỹ năng mềm khi phỏng vấn, ...
3. Tìm các tin tuyển dụng, đọc mô tả công việc (JD), tìm hiểu về công ty.
4. Ứng tuyển vào công ty (gửi mail cho công ty hoặc inbox với HR trên MXH)
5. Khi có lịch PV, lên nhắc nhở - hẹn giờ và có mặt đúng giờ.
6. Khi PV, hãy thoải mái, tự tin, thể hiện tinh thần cầu tiến và cố gắng hoàn thành tốt phần trả lời câu hỏi chuyên môn.

**Responsive Với Grid System**

Responsive là gì?

=> là kỹ thuật giúp website hiển thị tương thích với nhiều kích cỡ màn hình khác nhau (mobile, tablet, PC,...)

- Tối ưu trãi nghiệm người dùng:

1. hiển thị rõ ràng các thành phần (hình ảnh, cỡ chữ, nút bấm,..)

2. Ẩn / hiện các thành phần phù hợp theo kích thước màn hình

Chúng ta sẽ làm gì?

1. Dùng CSS thay đổi kích thước phù hợp cho các thành phần hiển thị trên website (hình ảnh, cỡ chữ, nút bấm,...)

2. Dùng CS để ẩn / hiện các thành phần phù hợp theo kích thước màn hình

*E.x: ẩn 1 đối tượng => display: none;*

Sử dụng - cài đặt công cụ:

*F12* và tùy chọn trên dev tool kích thước tỉ lệ màn hình

*- Extension trên Chrome: Viewport Resizer – Responsive Testing Tool*

Khái niệm Viewport:

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

màn hình hiển thị trên thiết bị của User

**content**="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

"*width*" --- định nghĩa chiều ngang của viewport

"*initial-scale*" --- định nghĩa độ zoom khi truy cập website

"*minimum-scale*" --- độ thu nhỏ tối thiểu của website

"*maximuun-scale*" --- độ phóng to tối đa của website

"*height*" --- chiều cao của khung hình

"*user-scalable*" --- chặn việc người dùng zoom to / nhỏ

***\*\*\*Nguyên tắc bất ngờ nhỏ nhất trong lập trình phần mềm\*\*\****

Media query

@media "not|only" mediatype "and" (mediafeature "and|or|not" mediafeature)

{ CSS code }

1. Keywords:

* not
* only
* and
* or

2. Mediatype:

* print
* screen
* speech
* all
* default

3. Mediafeatre:

* min-width
* max-width
* …

4. Polyfill – *respond.js polyfill cdn*

Cách 1:

* <https://cdnjs.com/libraries/respond.js/1.4.2>
* Copy <https://cdnjs.cloudflare.com/ajax/libs/respond.js/1.4.2/respond.js> dán vào phần head
* <script src=<https://cdnjs.cloudflare.com/ajax/libs/respond.js/1.4.2/respond.js>></ script>

Cách 2: <https://en.wikipedia.org/wiki/Conditional_comment>

Copy:

## Examples[[edit](https://en.wikipedia.org/w/index.php?title=Conditional_comment&action=edit&section=1)]

Here is a simple example that demonstrates how conditional comments work.

*<!--[if lt IE 9]>*

*<script src="https://cdnjs.cloudflare.com/ajax/libs/respond.js/1.4.2/respond.js" integrity="sha512-BWbLJlfp8hzXlxT6K5KLdxPVAj+4Zn2e4FVq5P7NSFH/mkAJ18UiZRQUD4anR3jyp0/WYkeZ0Zmq5EWWrDxneQ==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>*

*<![endif]-->*

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

* Thay thế vào script trong hình

Tra cứu những thuộc tính khác nhau trên các trình duyệt: <https://caniuse.com/>

Breakpoints

Breakpoints là những *điểm/vị trí* mà bố cục website *sẽ thay đổi* – thích ứng để tạo nên giao diện responsive.

<https://ui.dev/rwd/develop/browser-feature-support/media-queries-for-common-device-breakpoints>

|  |  |
| --- | --- |
| Mobile | width < 740px    @media only screen and (max-width: 739px) {} |
| Tablet | width >= 740px and width < 1024px  @media only screen and (min-width: 740px) and (max-width: 1023px) {} |
| PC | width >= 1024px  @media only screen and (min-width: 1024px) {} |

E.x:

E.x:



Media queries: Px, rem hay em?

* @media sử dụng em

Px to em converison: <https://www.w3schools.com/tags/ref_pxtoemconversion.asp>

Grid system

1. Tạo nên các khung nền, hỗ trợ việc sắp xếp bố cục theo trật tự/thống nhất cân bằng.

* Lưới nhiều cột (Mutilcolumn grid)
* Lưới 1 cột (Single column grid)
* Lưới modun (Modular grid)
* Lưới đường cơ sở (Baseline grid)

2. Vai trò:

* Tổ chức: có các đường căn gióng tiện lợi, dễ sắp xếp các thành phần ngăn nắp
* Cân bằng: Dù là đối xứng hay 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

3.Ứng dụng:

* Lưới trong thiết kế UI/UX: Vai trò đặt biệt quan trọng trong Responsive web design
* Lưới trong in ấn: Google “Grid system”

4. Responsive web design

* Grid: Thành phần cha
* Row: Dòng
* Column: Cột ( Column Parent – Column Child )
* Gutter: Khoảng cách 2 phía của cột 9.1.2 Thành phần chính (lý thuyết)

Trong đó:

* Column - Cột 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, tablet 8 cột, mobile 4 cột)
* Gutter - Đường ngăn cách (rãnh ngăn) Là khoảng cách 2 phía của 1 cột, tạo nên rănh ngăn giữa các cột. Độ rộng rãnh ngăn có thể thay đổi cho phù hợp với thiết kế hoặc độ rộng màn hình (VD: PC/Tablet 24px, mobile 16p)  
  Margin - Phần lề Là khoảng cách 2 bên trái/phải của bố cục chính của website. Độ rộng phần lề thay đối để phù hợp với các kích thước màn hình. VD: Phần lề lớn thích hợp cho màn hình lớn như PC, phần lề nhỏ thích hợp cho màn hình nhỏ như Tablet, mobil 9.1.3 Thành phần chính (làm việc với CSS)
* Grid - Lưới (Thường là phần cha, chứa Row và Column)
* Row - Dòng (Dòng - chiều ngang, chứa Column)
* Column - Cột (chứa nội dung / thành phần trên website) 9.2 Tự build Thư Viện

*\*\*\* Grid Tạo thư viện CSS ứng dụng Grid system.*

*Kết quả:*

*Tự tay xây dựng được thư viện CSS đầu tiên*

*Biết cách ứng dụng Grid system vào xây dựng layout*

*Hiểu về Grid layout của bootstrap 9.3 Tạo đối tượng Grid 9.3.1.*

*Tạo class*

*Grid : Full-width, chiếm hết chiều ngang đối tượng chứa (cha)*

*Wide : Chiều ngang đối đa 1200px*

*Hiện tại bây giờ màn hình đa số là 1280px => Người ta chọn chiều ngang là 1200px làm chuẩn (*[*https://1200px.com/*](https://1200px.com/)*)*

*Hồi xưa 1024px => người ta lấy 960px làm chuẩn (*[*https://960.gs/*](https://960.gs/)*)*