coding conventions

|  |  |
| --- | --- |
| **Tên dự án** | Phần mềm quản lý nhà sách |
| **Thực hiện dự án** | * Phan Lê Quang * Trương Diệu Linh * Nguyễn Đức Hưng * Hoàng Phan Lê |
| **Phiên bản tài liệu** | 1.0 |

Mục lục

1. Các nguyên tắc chung 1
2. NodeJS 1
   1. Khoảng trắng 1
   2. Toán tử ba ngôi 2
   3. Đặt tên 2
   4. Dòng và số ký tự trên một dòng 3
   5. Từ khoá và các giá trị True/False/Null 3
   6. Class, hàm và phương thức 3
   7. Comment 4
3. HTML 4
4. CSS 4
   1. Đặt ID và class 5
   2. Giá trị thuộc tính name 5
   3. CSS selector 5
   4. Sắp đặt các rule trong cùng một file 6
   5. Rule 6
   6. Các thuộc tính dạng shorthand 6
   7. Giá trị và đơn vị 7
   8. Thứ tự các thuộc tính trong rule 7
   9. Ngăn cách selector và các khai báo thuộc tính 7
   10. Ngăn cách các rule 7
   11. Các dấu trích dẫn 8
5. Javascript 8

# Các nguyên tắc chung

* Chỉ sử dụng UTF-8, không được chứa ký tự đặc biệt BOM.
* Sử dụng 4 khoảng trắng cho 1 ký tự tab (1 tab = 4 spaces), không sử dụng ký tự tab.
* Một hàm hoặc phương thức chỉ được xử lý một công việc duy nhất mà công việc này không thể tách nhỏ hơn được. Nếu công việc còn có thể tách nhỏ (về mặt ý nghĩa) thì phải tách nhỏ thành các hàm nhỏ hơn.
* Một hàm hoặc phương thức chỉ được phép truyền tối đa 5 tham số.
* Một file chỉ chứa một class.
* Mỗi một đoạn chức năng phải có comment rõ ràng.
* Trước khi định nghĩa một điều gì mới (class, hàm, phương thức, hằng,…) cần tìm kiếm trong project xem đã được viết chưa. Nếu không tìm thấy thì cần liên lạc với các dev khác để hỏi thông tin. Sau khi được xác nhận là chưa có thì mới tiến hành định nghĩa mới. Tránh việc trùng lắp code.
* Tên file, biến, hàm, class,… đều phải sử dụng tiếng Anh.
* Comment sử dụng tiếng Việt.
* Không hardcode các giá trị có nghĩa hoặc các giá trị được sử dụng nhiều lần. Cần phải khai báo hằng cho các giá trị này.

# NodeJS

## Khoảng trắng

* Luôn sử dụng khoảng trắng trước và sau các toán tử hai ngôi và các toán tử logic. Ví dụ:

// Đúng:

a = b + c;

// Sai:

a=b+c;

* Cần có 1 khoảng trắng nằm phía sau từ khoá của **các cấu trúc điều khiển** như if, while, for, foreach, switch, và catch. Ví dụ:

// Đúng:

if (isFoo()) {

a = 'foo';

}

// Sai:

if(isFoo()) {

a = 'foo';

}

* Ép kiểu:

// Đúng:

(int)foo;

// Sai:

(int) bar;

( int )bar;

( int ) bar;

## Toán tử ba ngôi

Chỉ sử dụng toán tử ba ngôi khi biểu thức rất ngắn và rõ ràng. Ví dụ:

a = b > 0 ? 0 : 1;

## Đặt tên

Tên bằng tiếng Anh phải có nghĩa, không quá dài, phải bắt đầu bằng chữ cái bảng chữ cái tiếng Anh hoặc chỉ 1 dấu gạch dưới \_.

* **Namespace và class** phải được đặt tên theo kiểu Pascal (PascalCase), là danh từ, chữ cái đầu mỗi từ phải viết hoa. Ví dụ:

// Đúng:

namespace App\Service;

class UserService

// Sai:

class userservice

class Userservice

class userService

* **Hằng** phải được đặt tên bằng các chữ cái in hoa, là danh từ, mỗi từ ngăn cách nhau bởi dấu gạch dưới \_. Ví dụ:

// Đúng:

const CENTRE\_NAME = ‘Nhập môn Công nghệ Phần mềm’;

// Sai:

const centre\_NAME = ‘Nhập môn Công nghệ Phần mềm’;

const centre\_name = ‘Nhập môn Công nghệ Phần mềm’;

const CENTRENAME = ‘Nhập môn Công nghệ Phần mềm’;

* **Biến** phải được đặt tên theo kiểu camel thường (lowerCamelCase), là danh từ, chữ cái đầu tiên trong tên viết thường, các chữ cái đầu mỗi từ còn lại phải viết hoa. Ví dụ:

// Đúng:

userTable

// Sai:

usertable

Usertable

UserTable

getUserTable

* **Hàm và phương thức** phải được đặt tên theo kiểu camel thường (lowerCamelCase), là động từ, chữ cái đầu tiên trong tên viết thường, các chữ cái đầu mỗi từ còn lại phải viết hoa. Ví dụ:

// Đúng:

getTitle()

// Sai:

GetTitle()

Gettitle()

gettitle()

## Dòng và số ký tự trên một dòng

* Không được để dòng trống (empty line) ở cuối mỗi file hay class.
* Không được để khoảng trắng ở cuối mỗi dòng.
* Mỗi dòng chỉ được phép chứa tối đa 120 ký tự không kể các ký tự khoảng trắng dùng để canh lề.
* Khuyến khích số ký tự trên mỗi dòng chỉ ít hơn hoặc bằng 80 ký tự.
* Mỗi dòng chỉ được chứa một statement.

## Từ khoá và các giá trị True/False/Null

* Các hằng true, false, null phải được viết thường.

## Class, hàm và phương thức

* Các dấu mở ngoặc nhọn { phải nằm trên dòng mới khi định nghĩa class, hàm và phương thức.
* Class, hàm và phương thức không được rỗng.

## Comment

* Chỉ sử dụng // cho comment đơn. Theo sau // phải có duy nhất một khoảng trắng. Phải sử dụng dấu câu để kết thúc comment đơn.
* Chỉ sử dụng /\* \*/ cho inline block-comment. Phải có dòng trống trước và sau inline block-comment.
* Mở đầu comment phải là chữ cái viết hoa.

# HTML

* Hạn chế tối đa việc viết Javascript vào file HTML. Trừ trường hợp xử lý nhỏ.
* Cần tách riêng các phần dùng chung để có thể tái sử dụng và trong trường hợp file HTML đã quá dài.
* Các thẻ HTML phải được căn chỉnh lề chính xác.
* Các thẻ HTML không phải dạng khối chỉ được chứa text hoặc các thẻ HTML inline. Ví dụ cặp thẻ <p></p> không được chứa thẻ <div>.
* Tên các thuộc tính của thẻ HTML phải viết toàn bộ bằng chữ thường. Giá trị của các thuộc tính được đóng khung bởi dấu nháy đôi “ ”.
* Khi có thẻ mở thì buộc phải có thẻ đóng. Các thẻ rỗng thì buộc phải sử dụng dạng tự đóng /> (trước hai ký hiệu này phải có duy nhất 1 khoảng trắng). Ví dụ: <hr></hr> 🡪 <hr />.
* Hạn chế tối đa việc sử dụng CSS dạng Internal và Inline.
* Comment có dạng:

<!-- Comment ngắn trên một dòng -->

<!--

Comment dài. Comment nhiều dòng.

Comment dài. Comment nhiều dòng.  
-->

* Phải sử dụng Bootstrap để bảo đảm tính responsive.
* Đặt ID và class cho các HTML element: xin tham khảo mục [5.1](#Ref511404130).

# CSS

Dựa theo Google HTML/CSS Style Guide.

(<https://google.github.io/styleguide/htmlcssguide.html>)

* Tên file viết thường, ngăn cách các từ bởi dấu gạch giữa -.

## Đặt ID và class

* Sử dụng tên ID và class có nghĩa.
* Hạn chế đặt quá nhiều ID và class, nếu có thể nên sử dụng CSS selector.
* Tên ID và class cần ngắn gọn nhưng phải thể hiện được ý nghĩa của nó. Ví dụ:

/\* Nên dùng \*/

#nav {}

.author {}

/\* Không nên dùng \*/

#navigation {}

.atr {}

* Các chữ cái có trong tên ID và class phải viết thường.
* Các từ trong tên ID và class cần ngăn cách bởi dấu gạch giữa -.

## Giá trị thuộc tính name

* Các chữ cái có trong giá trị của thuộc tính name phải viết thường.
* Các từ trong giá trị của thuộc tính năm cần ngăn cách bởi dấu gạch dưới -.

## CSS selector

* Khi định nghĩa rule sử dụng ID, không sử dụng các ancestor selector. Ví dụ:

/\* Đúng \*/

#child {}

/\* Sai \*/

#parent #child {}

.button #child {}

body #child {}

* Không sử dụng tên element cho ID. Hạn chế sử dụng tên element cho class trừ phi có các rule khác nhau cho từng element có cùng class. Ví dụ:

/\* Đúng \*/

#element {}

/\* Nên dùng nếu chỉ có một loại element \*/

.button {}

/\* Sai \*/

div#element {}

/\* Không nên dùng nếu chỉ có một loại element \*/

a.button {}

## Sắp đặt các rule trong cùng một file

* Các rule có sự liên quan như cùng ID, cùng class, cùng chỉ đến một element thì cần nằm liền kề nhau.
* Nên có các dòng comment ngăn cách các rule cho các phần khác nhau. Ví dụ:

/\*-------- Toolbar ---------\*/

rule1 {}

rule2 {}

/\*-------- News --------\*/

rule3 {}

rule4 {}

## Rule

* Không viết rule trên một dòng.
* Sau selector là một khoảng trắng rồi đến dấu mở ngoặc nhọn {.
* Mỗi khai báo thuộc tính kết thúc bởi dấu chấm phẩy ;.
* Trong mỗi dòng khai báo thuộc tính, trước giá trị của thuộc tính phải có khoảng trắng.

## Các thuộc tính dạng shorthand

Sử dụng các thuộc tính dạng shorthand khi có thể.

/\* Nên dùng \*/

border-top: 0;

font: 100%/1.6 palatino, georgia, serif;

padding: 0 1em 2em;

/\* Không nên dùng \*/

border-top-style: none;

font-family: palatino, georgia, serif;

font-size: 100%;

line-height: 1.6;

padding-bottom: 2em;

padding-left: 1em;

padding-right: 1em;

padding-top: 0;

## Giá trị và đơn vị

* Không sử dụng đơn vị cho giá trị 0 nếu không cần thiết. Một số thuộc tính luôn yêu cầu đơn vị hoặc trình duyệt IE sẽ lỗi nếu không có đơn vị.
* Lược bỏ số 0 đối với các giá trị trong khoảng từ -1 đến 1. Ví dụ:

font-size: .8em;

## Thứ tự các thuộc tính trong rule

* Các thuộc tính trong một rule phải được sắp xếp theo thứ tự alphabet, bỏ qua các tiền tố của phần vendor-specific (-moz, -webkit, ...).
* Đối với một thuộc tính sử dụng các vendor-specific thì cần sắp xếp các vendor-specific đó theo thứ tự alphabet.
* Ví dụ:

background: fuchsia;

border: 1px solid;

-moz-border-radius: 4px;

-webkit-border-radius: 4px;

border-radius: 4px;

color: black;

text-align: center;

text-indent: 2em;

## Ngăn cách selector và các khai báo thuộc tính

Mỗi selector hoặc khai báo thuộc tính phải được đặt trên một dòng riêng biệt. Ví dụ:

/\* Đúng \*/

h1,

h2,

h3 {

font-weight: normal;

line-height: 1.2;

}

/\* Sai \*/

a:focus, a:active {

position: relative; top: 1px;

}

## Ngăn cách các rule

Giữa mỗi rule phải có một dòng trống. Ví dụ:

html {

background: #fff;

}

body {

margin: auto;

width: 50%;

}

## Các dấu trích dẫn

* Nên sử dụng dấu nháy đơn ‘ ’ thay vì dấu nháy kép “ ”.
* Không sử dụng dấu trích dẫn đối với các URI (url()).
* Trường hợp ngoại lệ là @charset, nếu cần thiết vẫn có thể dùng dấu nháy kép.
* Ví dụ:

/\* Đúng \*/

@import url(https://www.google.com/css/maia.css);

html {

font-family: 'open sans', arial, sans-serif;

}

/\* Sai \*/

@import url("https://www.google.com/css/maia.css");

html {

font-family: "open sans", arial, sans-serif;

}

# Javascript

6.1 Quy tắc chung dành cho Jquery

* Tên file viết thường, ngăn cách các từ bởi dấu gạch giữa -.
* Tránh việc định nghĩa lại các hàm có sẵn.
* Luôn sử dụng jQuery để bảo đảm khả năng chạy trên nhiều trình duyệt khác nhau.
* Khi viết các event handler cho các HTML element, cần lưu ý:
  + Element đó luôn luôn tồn tại: selector đến đúng element đó để viết event handler.
  + Element đó có thể biến mất trong quá trình xử lý: selector đến element nào chắc chắn luôn tồn tại. Sử dụng phương thức on của jQuery.

Ví dụ:

// Element a luôn tồn tại

$(a).click(...);

// Element a có thể không tồn tại

$(document).on(‘click’, ‘a’,...);

* Trong một hàm, nếu một selector được sử dụng nhiều lần thì nên khai báo biến lưu trữ selector đó. Ví dụ:

// Nên dùng

var thisSel = $(this);

thisSel.text();

thisSel.html();

// Không nên

$(this).text();

$(this).html();

* Nên sử dụng dấu nháy đơn ‘ ’ thay vì dấu nháy kép “ ”.

6.2 . Quy tắc chung dành cho Javascript

* Luôn kết thúc câu lệnh bằng dấu “;”

Nên dùng

var values = ["Volvo", "Saab", "Fiat"];

Không nên

var values = ["Volvo", "Saab", "Fiat"];

* Khi làm việc với Đối tượng    
  + Luôn đặt dấu ngoặc nhọn “{“ ở cùng một dòng với tên đối tượng
  + Luôn có dẫu phẩy và một khoảng trắng sau đó giữa các thuộc tính va giá trị của thuộc tính tương ứng
  + Không nên thêm dấu phẩy “,” ở các thuộc tính cuối cùng của đối tượng.
  + Kết thúc các đối tượng bằng dấu ngoạc nhọn “}” ở dòng mới và không có khoảng trắng nào .
  + Luôn kết thúc đối tượng bằng dấu chẩm phẩy “;”
* Độ dài 1 dòng không quá 80 kí tự
* Sử dụng cú pháp đơn giản để nhúng một tập tin javascript

Nên dùng

<script src="myscript.js"></script>

Không nên

<script src="myscript.js" type=”text/javascript”></script>

* Sử dụng chữ cái in thường cho tên tập tin javascript