**QUY CHUẨN ĐẶT TÊN:**

1. PascalCase: Tên lớp. Vd: UserClass, CategoryClass…
2. camelCase: Tên hàm, tên phương thức, tên biến. Vd: getUser, …
3. UPPER\_CASE: Tên hằng số. Vd: DISCOUNT\_PERCENT, …
4. snake\_case: Tên bảng, tên cột CSDL SQL. Vd: company\_name, …
5. kabab-case: Tên file, tên UsRL . Vd: style-mobile.css, …

\* Chú ý:

- Không đặt tên biến bắt đầu bằng số.

- Không chứa ký tự đặc biệt.

- Không dùng trùng function mặc định của ngôn ngữ.

**CÁC PHÍM TẮT QUAN TRỌNG TRONG VISUAL CODE:**

1. F12: Đi đến hàm đang dùng.
2. Shift + F12: Đi đến những nơi hàm này được dùng.
3. Ctrl + `: Mở Terminal.
4. Ctrl + /: Comment.
5. Alt + <-: Trở về
6. Ctrl + p: Tìm mở file.
7. Ctrl + p + >: Tìm command của Visual Code hoặc của Extension.
8. Ctrl + p + #: Tìm hàm
9. Ctrl + p + :: Tìm hàng
10. Ctrl + Shift + k: Xóa 1 dòng.
11. Alt + Click: Chọn nhiều con trỏ sửa cùng lúc.
12. Alt + Up/Down: Di chuyển lên trên hoặc xuống dưới.
13. Alt + Shift + Up/Down: Copy lên trên hoặc xuống dưới.
14. Alt + Shift + i: Chọn nhiều con trỏ ở cuối hàng.
15. Ctrl + d: Tìm từng đoạn bôi đen giống nhau.
16. Ctrl + Shift + l: Tìm tất cả đoạn bôi đen giống nhau.
17. F2: Đồng bộ đổi tên hàm, biến.
18. Window + .: Thêm emoji.
19. Alt + z: Tự động xuống dòng (Wrap text).

**CÁC LỆNH CƠ BẢN CỦA TERMINAL, POWER SHELL, CMD, …**

**GIT CƠ BẢN**

1. git status: Kiểm tra tình trạng code thay đổi so với repository.
2. git log:
3. git add: Thêm code vào trong repository.
4. git commit: Thêm code vào trong repository.
5. git clone: Lấy remote repository từ trên server về
6. git pull: Đồng bộ từ trên server về máy.

**Cách up code lên git:**

1. Lên trang gitHub/gitLab tạo 1 Repository.
2. git init: Tạo Local repository trong máy tính (file .git).
3. git status: Kiểm tra tình trạng code thay đổi so với repository.

**HOW TO CODE**

1. Have a clear GOAL.
2. DON’T copy code.
3. DO tasks/ project/ practices.
4. Create your own style.
5. Calmdown.

**GLOSSARY**

1. Attributes: Thuộc tính
2. Properties: Đặc tính
3. Methods: Phương thức
4. Elements: Phần tử
5. Parameter: Tham số
6. Variable scope: Phạm vi biến – Global, Public, Private,…. Toàn cục, nội bộ…
7. API: Application Programming Interface.

**HTML**

1. ID > Class > Tag

**CSS**

1. Selector: Cú pháp thứ tự để chon
2. Inline > Internal > External

**JAVASCRIPT**

1. Closure (js): Hàm có thể truy cập biến của hàm khác.
2. DOM (js): API liên kết giữa Js và browser html

**DOM**

1. *property:* document.querySelector(‘*Tên .class/#id/selector*’).*element*
2. *property:* document.querySelectorAll(‘*Tên .class/#id/selector*’).*element*
3. *property*: document.createElement(‘Tên *tag*’) – thêm một thẻ tag mới
4. *element*: textContent – nội dung text trong thẻ tag html
5. *element*: innerHTML – nội dung code html trong thẻ tag html­­­­­
6. *element*: insertAdjacentHTML – thêm nối tiếp nội dung code html
7. *element*: classlist – thêm/xóa class.
8. *element*: prepend/append – di chuyển vào đầu/cuối trong thẻ tag code html
9. Dom tree: nodes
10. NodeList: No update. Vd: document.querySelectorAll
11. HTMLCollection: Update. Vd: document.getElementsByTagName

**ARRAY**

1. Methods: slice, splice, reverse, concat, join.
2. Methods: foreach, map, filter, reduce.

**FUNCTION**

1. Regular/Declaration: functionName(parameter1, parameter2, parameter3){ *code* }
2. Expression: const fuctionName = function(parameter1, parameter2, parameter3) { *code* }
3. Arrow: functionName = (parameter1, parameter2, parameter3) => { code }
4. Callback function: Hàm được gán làm tham số của hàm khác