1. Thuật toán: là tập hợp hữu hạn các bước để giải quyết bài toán/ 1 vấn đề nào đó

- Mã giả (Pseudo-code):

+ Ưu: đơn giản, gần với ngôn ngữ tự nhiên

+ Nhược: không có quy định rõ ràng, nhất quán

- Lưu đồ (Flowchart):

+ Ưu: có quy tắc rõ ràng, chuẩn chỉnh, tránh nhầm lẫn khi truyền đạt

+ Nhược: đôi khi quá chặt chẽ, gây khó nhớ cho người dùng

- Elipse: bắt đầu/kết thúc

- Bình hành: nhập/xuất dữ liệu

- Chữ nhật: công thức tính toán

- Thoi: rẽ nhánh (biểu thức điều kiện)

- Mũi tên: biểu thi luồng chương trình

1. Git: hệ thống điểu khiển các phiên bản theo hình thức phân tán, sử dụng để quản lý mã nguồn, lưu trữ thay đổi

- VCS: version control systems

- git init: tạo file quản lý repo (thư mục .git)

- git remote add origin <url>: kết nối local-repo với remote-repo

- git add **.** : thêm toàn bộ file vào thư mục quản lý

-git add <tên file>: thêm file có tên ở bên vào thư mục quản lý

- git commit -m ”<message>”: ghi nhận các thay đổi, tạo snapshot (các phiên bản)

- git push origin master: đẩy lên remote

- git clone: sao chép 1 repo

- git pull: cập nhật mã nguồn từ remote về local

1. HTML - HyperText Markup Language: là ngôn ngữ đanh dấu siêu văn bản, xây dựng cơ bản giao diện trang web

- h1 => h6: thẻ tiêu đề, làm to tiêu đề, tô đậm

- p: đánh dấu văn bản; id, class => innerText, innerHTML

- a: đánh dấu siêu liên kết; src, target

- img: hiển thị hình ảnh; src, alt

- input: nhập liệu; value, type, id, class, name, placeholder => value

- ol: orderlist: có thứ tự; type = 1,A,a,I

- ul: underlist: không thứ tự; style=””

- li: listitem: thành phần của danh sách

- hr: kẻ ngang: width

- br: xuống dòng

- span

- label: for=”id”

- div: bao khối

- form: tạo biểu mẫu: method, action

- pre

- audio: phát nhạc: src, controls (autoplay)

- video: phát video: src

- b (bold): in đậm

- i (italic): in nghiêng

- button: tạo nút bấm, đi kèm các sự kiện: onclick

- select-option: tạo lựa chọn dạng dropdown

- textarea: văn bản lớn: row, col

\* Table: tạo bảng

- tr: tạo các hàng, dòng

- td: tạo các cell, ô

- th: tiêu đề, in đậm và căn giữa

- colspan, rowspan: gom hàng, gom cột

CSS: Cascading Style Sheets: chỉnh sửa gian diện, tương tác vs các thẻ HTML

1. JavaScript: là ngôn ngữ lập trình được sử dụng trên nhiều website; thay đổi nội dung và giao diện của trang web

- 3 cách:

+ inline: viết trực tiếp vào thẻ html

+ internal: dùng cặp thẻ <script>

+ external: tách thành file .js và dẫn link

- onclick: click vào 1 thành phần HTML

- onmouseout: di chuột ra khỏi thành phần HTML

- onmouover: di chuột qua thành phần HTML

- onchange: thay đổi trong thẻ input => click sang thành phần khác

- oninput: khi có sự thay đổi trong thẻ input => tác động ngay khi gõ phím

- window.onload: khi mn bật của sổ window

- window.addEventListener(các sự kiện, hàm cần gọi): keydown, keyup, keypress

- document: chính là màn hình

**\* Chú ý Coding Consvention, CamelCase cách Debug**

\*) Biến: là tên gọi gắn vs 1 vùng nhớ chưa dữ liệu; vd: numberOne, hinhChuNhat = rectangle, numberB = numB, sideA = canhA

\*) 7 kiểu dữ liệu:

- Number: số (0-9)

- String: chuỗi (a-z), 1 chuỗi dài

- Boolean: true/false

- Undefined: chưa định dạng, không xác định

- Null

- Symbol

*- Object: đối tượng*

- const: hằng, giá trị không đổi

- let: cục bộ, trong 1 khối nó được khai báo

- var: toàn cục, toàn bộ ct

\*) Toán tử:

- Toán tử số học: + - \* / % \*\*

- Toán tử tăng giảm: ++ --

- Toán tử logic: || && !

- Toán tử so sánh: > < >= <= == === != !==

- Toán tử gán: =

- Toán tử kết hợp: += \*= /= -=

1. Câu lệnh điều kiện: là câu lệnh rẽ nhánh, nhằm thay đổi luồng thực thi của chương trình

- Cú pháp:

+ If: đúng thì làm

+ If… else: dúng thì làm if, sai thì làm trong else

+ If bậc thang (if…else if): đúng thì là if, sai thì sang else xét tiếp đk if trong else

+ If lồng: đúng thì xét tiếp điều kiện if tiếp theo

+ Switch…case: switch đúng case nào thì làm case đó

- break: dừng sau khi thực hiện xong case

- default: nếu k case nào đúng thì vào cái này

+ Toán tử 3 ngôi: điều kiện ? true : false

1. Câu lệnh lặp: là thực thi lặp đi lặp lại 1 khối lệnh khi điều kiện còn đúng (dựa theo 1 điều kiện cho trc)

- Cú pháp:

- for

- while

- do…while

- for…in: trả về index phần tử của mảng

- for…of: trả về giá trị phần tử của mảng

+ break: dừng vòng lặp

+ continue: bỏ vòng lặp hiện tại

1. Mảng: là biến đặc biệt, lưu trữ được nhiều giá trị; là các phần tử của mảng, được sắp xếp liền kề nhau trong bộ nhớ

- Cú pháp:

- Mảng 1 chiều:

+ let array = []

+ let array = new Array(số lượng các phần tử của mảng)

- Mảng 2 chiều:

+ let array = [[],[]]

+ let array = new Array(số lượng các phần tử của mảng)

for (let i = 0; i < array.length; i++) {

array[i] = new Array(số lượng các phần tử của mảng)

}

- length: độ dài của mảng, số lượng các phần tử

- index: 0 => length-1: chỉ số, vị trí các phần tử trong mảng

\*) Các hàm có sẵn:

- pop(): xóa cuối

- push(): thêm cuối

- shift(): xóa đầu

- unshift(): thêm đầu

- concat(): nối mảng

- join(): nối các phần tử => chuỗi

- toString(): chuyển thành chuỗi: phần tử 1, phần tử 2, phần tử 4, phần tử 4,…

- map()

- splice(): xóa phần tử từ index, số lượng phần tử cần xóa

- slice(): xóa phần tử từ index thứ 1 => index thứ 2, nhưng k xóa thằng cuối

- indexOf(): trả về chỉ số phần tử trong mảng

- reverse(): đảo mảng

- sort(funtion(a,b) {return a-b}): sắp xếp

1. Hàm: tập hợp các câu lệnh để thực hiện 1 nhiệm vụ nhất định

- Tên hàm: tuân thủ Coding Consvention: camelCase, tên có ý nghĩa

- Cú pháp:

function <tên hàm> (các tham số nếu có) {

statement…

}

- Hàm có tham số, có giá trị trả về

- Hàm có tham số, không gtri trả về

- Hàm không tham số, có gtri trả về

- Hàm không có gì cả

1. OOP:

- khái niệm, so sánh, ưu nhược điểm

- các thành phần: class, object, property, method

- cú pháp khởi tạo lớp; khởi tạo đối tượng (cú pháp JS ES6)

- sử dụng đối tượng gọi đến thuộc tính, phương thức

- 4 tính chất