**1/ Thuật toán/ giải thuật**: là các chỉ thị đưa ra để giải quyết vấn đề

**2/ Mã giả và lưu đồ**   
**Mã giả**: Ngôn ngữ tự nhiên bằng tiếng anh, không thực thi được, ko có quy định chặt chẽ

**Lưu đồ**: ký hiệu:

Elip: bắt đầu/ kết thúc

Chữ nhật: Tính toán

Bình Hành: Xuát nhập

Thoi: Qđịnh, rẽ nhánh

Mũi Tên: Dòng chảy Bộ nối

**3/ Git**

Git là 1 VCS (Version Control System) - Hệ thống quản lý phiên kiểu Phân tán

* **Câu Lệnh** : 1 Remote chỉ nên kết nối vs 1 local

+ Git clone : Sao chép 1 repository

+ Git init : khởi tạo repo (file .git)

+ Git remote add origin <url>: kết nối giữa local vs remote

+ Git add < tên file> : Thêm các file có tên ở bên vào git để quản lý

+ Git add . : thêm tất cả các file vào git

+ Git commit –m “message”: Ghi nhận thay đổi (tạo phiên bản/snapshot)

+ Git push origin master/main: Đưa các thay đổi từ Local lên Remote

+ Git reset: Hủy bỏ các thay đổi trước đó

+ Git status: Xem trạng thái hiện tại của các thay đổi

**HTML: Hypertext Markup Language - ngôn ngữ đánh dấu siêu văn bản**

<html></html>: đánh dấu trang HTMl

Thẻ <p></p> : tạo đoạn văn bản

Thẻ <hi> </hi>: Các thẻ tiêu đề (1 => 6)

Thẻ <a> </a> : Thẻ tạo liên kết, đi cùng thuộc tính href=”url” target=”\_blank, \_top, \_self…”

Thẻ <img> : Chèn hình ảnh, có thuộc tính: src=”url”, alt

Thẻ <ol></ol>: danh sách có thứ tự đi với thẻ <li></li>; thuộc tính type=”A”

Thẻ <ul></ul>: danh sách không có thứ tự đi với thẻ <li></li>; style=”list-type-style: disc”

Thẻ <hr>: Thẻ kẻ 1 đường nằm ngang tài liệu HTML

<br>: Xuống dòng

<pre>: Định nghĩa đoạn vb được định dạng trước như thơ

<button></button>: gắn với các sự kiện của Java script để xử lý thao tác ng dùng, là 1 cái nút

Thuộc tính:

id=”bookmark”: tạo index cho thành phần (element) của html

class=”bookmark”: tạo nhóm chung cho các thành phần (element) của html

href=”…” : định nghĩa địa chỉ link liên kết

alt: thông báo thay thế ảnh ko hiển thị được

src: định nghĩa url hình ảnh

float : đặt ảnh bên trái hoặc phải văn bản

type: định dạng kiểu cho biểu tượng các phần tử

Target:

\_blank: mở liên kết trong của sổ/tab mới

\_self: tab/cửa sổ hiện tại

\_top: của sổ chính

\_parent: parent frame

\_framename

**FORM**:

<form></form>: đánh dấu thành phần form

Thuộc tính :

Action: quy định nơi dữ liệu được gửi đến

Name: tên của trường dữ liệu khi được gửi đi

Method

- get: hiển thị trên thanh địa chỉ

- post: Không hiển thị

Thẻ Input:

<text area></text area>: nhập dữ liệu dạng text lớn

<input type = “text”>: nhập dữ liệu chuỗi ngắn

<input type =” password” > : nhập pass

<input type =”reset” value = “reset”>: tạo nút nhấn, khi nhấn sẽ trả lại dữ liệu ban đầu trc khi nhập

<input type =”submit” value = “submit”>: khi nhấn sẽ gửi dữ liệu lên server, thông qua action

<input type =”button” value = “clickMe”>: khi nhấn sẽ thao tác như 1 thẻ button (có value)

Kiểu lựa chọn:

<input type =”radio”>: cho phép lựa chọn 1 giá trị cho trước

<input type =”checkbox”>: cho phép lựa chọn nhiều hơn 1 giá trị cho trước

<select> <option></option> </select> : hiển thị danh sách lựa chọn ở dạng dropdowm

**Table:**

<table></table>: đánh dấu bảng

<tr> </tr> : Khai báo dòng

<th></th>: khai báo ô tiêu đề

<td> </td>: Khai báo 1 ô dữ liệu

Thuộc tính:

colspans: gộp cột (thực chất là gộp các cell ở các cột khác nhau)

rowspans: gộp hàng (thực chất là gộp các cell ở các hàng khác nhau)

**4/ Java script** là ngôn ngữ lập trình web và các ứng dụng back-end chạy trên trình duyệt. Đơn giản, dễ sử dụng, thường dùng để thay đổi nội dung và giao diện của trang web

**Cách nhúng**: đi kèm với thuộc tính src= ‘’ ‘’

Inline : Viết thẳng vào thẻ html

Internal: Viết trong cặp thẻ <script></script>

External: Viết trong file .js, nhúng thông qua thẻ <script> đi kèm với thuộc tính src= ‘’ ‘’ (kéo vào html)

**Trong javascript có**:

* Giá trị
* Toán tử
* Biểu thức
* Từ khóa
* Chú thích

**Nhập dữ liệu**:

prompt(”..”) hiện thị dialog box có 1 ô input để nhập dữ liệu, mặc định kiểu dữ liệu trả về là string

Dùng input: document.getElementById(“id”).value

**Cách để hiện kết quả trong js**: các thuộc tính

* document.write: hiện thị ra trang HTML; xóa các thành phần sẵn có
* alert(“thông báo”): hiển thị thông báo trong 1 dialog box
* console.log(): hiển thị trong thẻ console của f12/inspect/kiểm tra
* document.getElementById(“id”)
  + innerText: hiện thị tất cả dạng text
  + innerHTML: nhận biết và hiển thị các thẻ HTML dưới dạng thẻ

**Biến, kiểu dữ liệu, toán tử**

**Cách khai báo biến:**

Biến là tên gọi của 1 bộ nhớ chứa dữ liệu

Var = biến toàn cục, chạy trên cả chương trình

Let = biến cục bộ, chỉ chạy trong khối được khai báo

Const: khai báo hằng

**Các kiểu dữ liệu:**

* number: 0-9
* string: a-z, dạng chuỗi
* null (# rỗng)
* undefined : không định kiểu
* symbol: ký tự đặc biệt
* object: đối tượng
* boolean : true - false, nếu let = NaN, rỗng, null, undefined, 0 sẽ tự động bị ghép kiểu về boolean là FALSE

**Các loại toán tử**

* Toán học : + - \* / % ++ -- \*\*
* Gán : =; +=; -=; \*=; /=; %=
* Cộng chuỗi
* So sánh: ==; ===; !=; !==; >; < ; >=; <=
* Logic: &&; ||; !
* typeOf: lấy về kiểu dữ liệu của biến hoặc giá trị

**Note: a++ là cộng sau khi gán; ++a là cộng trước khi gán; riêng toán tử cộng số học, phải đảm bảo 2 số đều là dạng Number**

**5/ Câu lệnh điều kiện**

Câu lệnh điều kiện là câu lệnh ra quyết định, thay đổi luồng chạy của chương trình dựa trên các điều kiện cho trước.

Cú pháp: if(condition) { các câu lệnh;}

Các câu lệnh sẽ được thực thi nếu trả về true

Exp: if ( i=0 ) { i<10; i=i++;

Console.log (“i”) }

**Câu lệnh if else:**

If (condition) { câu lệnh;}

Else { câu lệnh;}

**If else lồng nhau:**

IF(condition1) {

If (condition2) { true statement;

Else false statement ; }}

Else { false statement;}

**Câu Lệnh Switch case**: Sử dụng trong so sánh bằng, và chỉ có 1 điều kiện duy nhất để so sánh, nếu điều kiện đúng mà không có thẻ <break> thì tất cả các câu lệnh sau đó đều được thực thi, cho đến hết hoặc khi gặp câu lệnh có break.

Cú pháp

Switch (biểu thức) {

Case 1:

Câu lệnh 1

Break;

Case 2:

Câu lệnh 2

Break;

Default:

**}**

**Toán tử ba ngôi: condition ? True : False**

Exp:

let x = true;

let a;

x === false ? a = 10 : a = 5

**Vòng lặp (loop**) : cho phép thực hiện 1 khối lệnh lặp đi lặp lại dựa vào 1 điều kiện cho trước bao gồm:

Một điều kiện để đánh giá ( lúc nào thì lặp)

Một khối lệnh để thực thi ( lặp cái gì?)

Các vòng lặp hỗ trợ bởi java script: for; while và do-while

For ( initial action; loop continuation –condition; action after each iterration) { statement;}

1. (2) (4) (3)

**Vòng lặp while**: while ( loop continuation-condition) {statement;}

* While thường sử dụng khi không biết trước số lần lặp
* For thường sử dụng khi biết trước số lần lặp

**Do while:**

Do { statement;}

While (loop condition);

* Nếu điều kiện trả về true thì tiếp tục vòng lặp, nếu false thì vòng lặp kết thúc. Thân của vòng lặp do while sẽ thực hiện ít nhất 1 lần kể cả khi false.