Biến là gì?

Biến là một không gian chứa một giá trị dữ liệu, và có thể được gán đi gán lại các giá trị khác nhau trên cùng 1 biến.

Cấu trúc điều kiện là gì?

Cấu trúc điều kiện là kiểm tra một điều kiện và thực hiện một hành động đúng với điều kiện đúng còn nếu sai thì sẽ thực hiện hành động theo điều kiện sai mà ta đã đặt ra. Vd if, if else.

Cấu trúc lặp là gì?

Cấu trúc lặp là một điều khiển thực hiện công việc lặp đi lặp lại khi chưa đủ số lần lặp hoặc khi một điều kiện nào đó còn đúng. Vd for, do while, while,

Hàm là gì?

Hàm là là một khối câu lệnh có thể tái sử dụng để thực hiện một tác vụ cụ thể.

Lập trình là gì?

Lập trình là quá trình tạo ra tập các chỉ dẫn để ra lệnh cho máy tính hoàn thành một công việc nào đó.

Pseudo-code là gì?

Mã giả là cách mô tả thuật toán bằng ngôn ngữ tự nhiên

Ưu điểm: đơn giản, không cần kiến thức về cách biểu diễn(lưu đồ, ngôn ngữ lập trình)

Flowchart là gì?

Lưu đồ là mô tả thuật toán bằng các sơ đồ hình khối, mỗi khối quy định một hành động.

1. SCMs là gì?  
2. VCS là gì?  
3. Git là gì?  
4. Các định nghĩa cơ bản về git và thứ tự để commit 1 file README.md lên git repositiry

1. Source code management system – Hệ thống quản lý mã nguồn là một phần mềm hỗ trợ:

Phối hợp giữa các thành viên trong một nhóm phát triển phần mềm

Quản lý tập tin và kiểm soát phiên bản

Các nhà phát triển khả năng làm việc đồng thời trên các tập tin, hợp nhất với các thay đổi khác của nhà phát triển khác

Theo dõi và kiểm tra các thay đổi được yêu cầu và thực thi

Theo dõi tình trạng sửa lỗi và thực thi

1. Version control system (VCS) – hệ thống quản lý phiên bản mã nguồn là một phần mềm hỗ trợ:

Khôi phục lại phiên bản cũ của các file

Khôi phục lại phiên bản cũ của toàn bộ dự án

Xem lại các thay đổi đã được thực hiện theo thời gian

Xem ai là người thực hiện thay đổi cuối cùng có thể gây ra sự cố

Khôi phục lại các file vô tình xoá mất

1. Git là một hệ thống điều khiển phiên bản (version control system) theo hình thức phân tán

Git được sử dụng để quản lý mã nguồn (source code) và ghi nhận các thay đổi

Github là một dịch vụ Git được cung cấp miễn phí

Github có phiên bản trả phí dành cho các doanh nghiệp

**Lệnh truy cập terminal**

git add . (đưa tất cả các file vào vùng)

git commit -m”nội dung tin nhắn (ghi nhận các thay đổi so vs phiên bản trc đó và tạo snapshot)

git push origin main: đẩy mã nguồn từ local đến remote

1. **Snapshot, commit, resposiroty, pull, push**

**Snapshot**

Là toàn bộ mã nguồn tại một thời điểm

Các snapshot như là các lát cắt được tạo ra trong quá trình phát triển

Lập trình viên quyết định lúc nào thì tạo một snapshot

Có thể quay lại một snapshot bất kỳ

**Commit**

Là cách để tạo các snapshot

Các commit thường được tạo ra khi có một thay đổi đáng kể đối với mã nguồn:

tạo một tính năng mới

sửa được một lỗi

cải tiến mã nguồn

….

Một commit bao gồm các thông tin:

Thay đổi các file so với trước

Một tham chiếu đến commit trước nó (gọi là commit cha)

Một mã băm đại diên, thường có dạng như ….(một dãy mã có tính duy nhất)

**Resposity**

Thường được gọi ngắn gọn là repo

Respository là *nơi chứa toàn bộ mã nguồn*

Respository bao gồm toàn bộ các file và lịch sử của các file đó

Respository chứa tất cả các commit

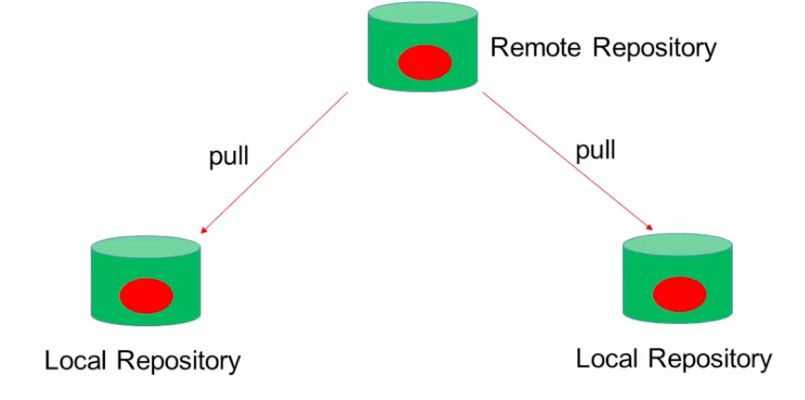
Có 2 loại respository:

*Local respository*: ở trên máy của lập trình viên

*Remote respository*: ở trên một máy chủ chia sẻ (chẳng hạng như GITHUB)

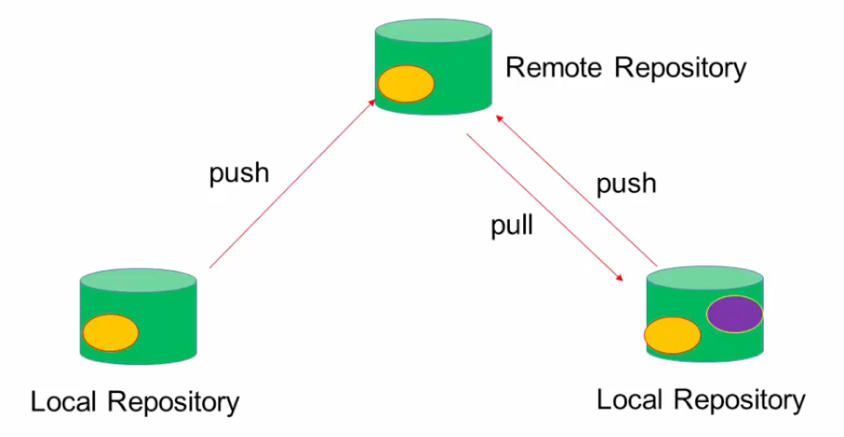
**Pull**

cập nhật mã nguồn từ một Remote respository về local respository



**Push**

Đẩy mã nguồn từ Local respository lên remote respository



1. Mục đích ý nghĩa của form ?  
2. Phân biệt sự khác nhau giữa POST và GET ?  
3. Mục đích sử dụng bảng ? Các thẻ tạo bảng?

1. . Form là cơ chế cho phép người dùng nhập dữ liệu và gửi về sever

. Chẳng hạn, chúng ta có form đăng nhập, form đăng ký, form tìm kiếm…

. Một form có thể bao gồm nhiều trường nhập dữ liệu khác nhau

1. . Post: Dữ liệu được gửi đi KHÔNG hiển thị thanh địa chỉ của trình duyệt.

Được sử dụng để gửi các dữ liệu bảo mật

Không hạn chế dung lượng dữ liệu gửi đi. Thường được sử dụng để gửi dữ liệu lớn lên sever.

. Get: Dữ liệu được gửi đi sẽ hiển thị trên thanh địa chỉ của trình duyệt.

Không nên sử dụng để gửi các dữ liệu nhạy cảm (chẳng hạn như bảo mật).

Thường được sử dụng để gửi những dữ liệu nhỏ.

1. Các thẻ tạo bảng:

<table>: khai báo một bảng

<tr>: khai báo một dòng

<th>: khai báo một ô tiêu đề

<td>: khai báo một ô dữ liệu, căn lề sang phải <td align=”right”>

Thuộc tính trong <td>: colspan mở rộng cột

Rowspan mở rộng hàng

Mục đích: muốn hiển thị dữ liệu dưới dạng các cột và các dòng

1.JS là gì?  
2. Các cách nhúng JS?  
3. Quy ước đặt tên  
4. Các cách để viết chú thích trong JS?  
5. 1 số phương pháp debug?

1. JS là một ngôn ngữ script được dùng để tạo các script ở máy client và máy server. Các script ở máy client có thể thực thi được trong trình duyệt trong khi các script ở máy server lại thực hiện trên server.

Ứng dụng:

-thay đổi giao diện

-thay đổi nội dung

-tăng tính tương tác(bởi vì nó xử lý sự kiện)

1. Inline: viết trực tiếp trong thẻ HTML

* Internal: dùng cặp thẻ JS và viết đoạn code bên trong
* External: viết 1 file JS

5- 1 số phương pháp debug:

Đọc mã nguồn( hand-trace)

Chèn các lệnh in ra các giá trị trong từng đoạn của chương trình để kiểm tra các giá trị và việc thực thi các câu lệnh

Sử dụng debugger- một chương trình cho phép quan sát quá trình thực thi của một ứng dụng

Các IDE thông thường cũng tích hợp sẵn debugger

Các công cụ debug của trình duyệt

Hàm nhập xuất trong JS và Demo

-prompt: nhập thông tin

-confirm: xác nhận thông tin

-Alert: hiện thị thông báo

-console.log: debug

-

//lắng nghe sự kiện Click: addEventListener

//khi sự kiện click xảy ra thì nó sẽ thực hiện funcion và Alert

Document.getElementById(“btn”).addEventListener(“click”, function () {

Alert(“Hello”)

})

1.Biến là gì?  
2. Cú pháp khai báo biến?  
3. Các cách khai báo biến nào? Phân biệt sự khác nhau giữa các cách?  
4. Nêu 1 số quy tắc đặt tên biến?  
5. Kiểu dữ liệu là gì?  
6. Các kiểu dữ liệu trong javascript?  
7. Các toán tử trong javascript? Độ ưu tiên các toán tử?  
8. Phân biệt == (bằng) và === (bằng giá trị đồng thời cùng kiểu dữ liệu)

1. **Biến** là một tên gọi được gắn cho một vùng nhớ chứa dữ liệu

* Dữ liệu được lưu trữ trong vùng nhớ của biến được gọi là **giá trị** (value)
* Có thể truy nhập, gán hay thay đổi giá trị của biến
* Khi gán một giá trị mới thì giá trị cũ sẽ bị ghi đè lên
* Cần phải khai báo biến trước khi sử dụng

2. Cú pháp khai báo biến

let diemToan = 9;

Const PI = 3.14;

var

3.

1. điểm khác nhau:

khai báo lại

hoising

scope

***Luôn luôn sử dụng let hoặc const để khai báo biến (ES6)***

4.

Tên biến phải bắt đầu bằng một ký tự alphabet (a-zA-z\_) • Theosaukýtựđầucóthểlàcáckýtựchữ,số...  
• Nên tránh đặt tên biến trùng tên các từ khoá  
• Tên biến nên mô tả được ý nghĩa của nó

• Tránh dùng các ký tự gây lầm lẫn  
• Tên biến có phân biệt chữ hoa và chữ thường  
• Nên áp dụng các quy ước đặt tên biến chuẩn khi lập trình

5.Kiểu dữ liệu

String: a = “Kien”

Number: b=5

Boolean a=true hoặc a=false

Underfined: let a;

Null: let a = null;

Object

Symbol: ES6

6. Các loại toán tử

Số học: +, -,\*,/,%

So sánh: ==, ===, >=, <=, >, <, !=, !==

Gán: =, +=, -=, /=, \*=, %=

cộng chuỗi: +

logic: &&, ||, !

typeof: typeof a

MẢNG

**Thêm phần tử vào mảng:**

Thêm vào CUỐI:

cars.**push**(“Kia”)

console.log(cars[cars.**length -1]).**

Thêm vào ĐẦU

Cars.unshift(“Ferari”)

console.log(cars[0])

**Xoá phần tử của mảng:**

Xoá phần tử CUỐI:

Cars.length

**Let last= cars.pop()**

**Cars.length**

Xoá phần tử ĐẦU:

Cars.length

**Let last = cars.shift()**

**Cars.length**

**Sắp xếp vị trí:**

ĐẢO NGƯỢC:

Let arr= [“Toyota” , “Subaru”, “BMW”]

**Document.write(arr.sort() + “<br>”)**

**Document.write(arr.reverse())**

**Theo thứ tự**

let points = [40, 100, 1, 5, 25, 10]

**points.sort(function(a, b) {**

**return a - b**

**})**

document.write(points + "<br>");

**points.sort(function(a, b) {**

**return b - a**

**})**

document.write(points);