1. Repository

* Là vùng chứa dữ liệu
* Gồm 2 loại : local repository và remote repository

1. Lệnh căn bản của git

* Git init: khởi tạo 1 repository
* Git add thêm file vào local
* Git commit lưu file vào local và lưu lại lịch sử snapshort
* Git push đẩy file từ local lên remote

1. Giải thuật

* Là bao gồm các chỉ thị để giải quyết 1 vấn đề
* Các cách biểu diễn thuật toán:

+ mã giả: nn tự nhiên (tiếng anh) để diễn đạt chương trình

+ flow chart: dùng các hình khối để diễn đạt chương trình

+ ngôn ngữ lập trình

1. Mô tả thuật toán tìm kiếm phần tử trong mảng
2. Trình bày ý tưởng của 1 thuật toán:
3. Mô tả thuật toán sắp xếp bảng số nguyên bằng mã giả
4. Các thẻ tạo danh sách

* <ul> : có dấu chấm tròn ở đầu dòng (k được sắp xếp theo thứ tự, đánh dánh bằng ký tự đặc trưng)
* <ol> : có stt ( sắp xếp theo thứ tự, đánh dấu bằng số hoặc chữ cái)

1. Các thẻ tạo nên table

* Table
* <tr> : tạo 1 dòng (table row)\_
* <th> : in đậm căn giữa (tablehead)
* <td>: khai báo 1 ô dữ liệu (table data)

1. Phân biệt innerHTML và innerText

* Html: định dạng theo html và xử lý 1 thẻ html nếu đc tìm thấy trong 1 chuỗi
* innerText: định dạng theo Text và không xử lý theo html nếu được tìm thấy trong chuỗi

1. 1 số thẻ html cơ bản

* <h1> đến <h6>
* Tạo văn bản:

+ <p>: rớt dòng

+<span>: dòng liền

+ <hr> : tạo đường ngang

+ <br>: xuống dòng

* <a href=””></a>: tạo liên kết
* Tạo danh sách: <ul> và <ol>
* Chèn hình ảnh: img

1. Các thuộc tính cơ bản thẻ form

* Thuộc tính action: có giá trị là 1 URL, quy định nơi dữ liệu gửi đến
* Thuộc tính name: quy định tên của form, sau đó có thể
* Thuộc tính methood: (get/post) quy định cách thức dữ liệu được gửi đến sever

1. Phân biệt get/post

* Get:

+ thông tin được hiển thị ở URL

+ không bảo mật

+ dữ liệu được gửi đi theo dạng Text

* Post:

+ k hiển thị thông tin ở URL

+ bảo mật

+ dữ liệu gửi đi đa dạng, text, img, video vì dữ liệu gửi đi thông qua HTTP request

1. Var và let, phạm vi khai báo biến

* Let: + k có cơ chế hoistring

+ k khai báo lại được

+ block scope( sử dụng trong 1 khối lệnh)

* Var: + có cơ chế hoistring ( sử dụng trước, khai báo sau)

+ khai báo lại được

+ function: phạm vi khai báo ở trong function

* Có 2 loại phạm vi biến
  + Biến cục bộ (local scope): lưu khai báo trên 1 hàm ( cùng 1 tên biến có thể khai báo ở nhiều hàm khác nhau)
  + Biến toàn cục (global scope): được khai báo ngoài hàm

1. Các kiểu dữ liệu trong js, làm sao biết kiểu dữ liệu
   1. Các kiểu dữ liệu trong js

* Kiểu nguyên thủy:
  + String: khai báo chuỗi các kí tự
  + Number : khai báo kiểu số nguyên, số thập phân, số - +
  + Boolean : true hoặc false
  + Undefiled : 1 biến chưa gán giá trị
  + Null : không chứa giá trị nào
  + Symbol
* Kiểu đối tượng
  + Array (mảng)
  + Object : do ctrinh cung cấp hoặc do lập trình viên tự định nghĩa
  1. Xác định kiểu dữ liệu
* Typeof x và typeof (x) dùng để kiểm tra kiểu dữ liệu của biến
* Null là kiểu dữ liệu được khai báo cho biến
* Undefiled: biến chưa gán giá trị

1. Tạo chuỗi chứa dấu nháy

* Đơn (‘’)
* Đôi (“”)
* Huyền (``)

1. NaN là gì, NaN === NaN không, why?

* NaN là not a number: không phải là số
* NaN === NaN không đúng tại vì NaN có thể được hiểu là nhiều kiểu dữ liệu

1. Tiền tố ++a, hậu tố b++
2. Các loại toán tử trong js, độ ưu tiên
   1. Các loại toán tử

* Số học ( +, -, \*, %, /0
* So sánh ( <, >, =, <=, >=, ==, ===, !=, !== )
* Logic (&&, ||, !)
* Gán ( +=, -=, \*=, /=, %=, = )
* Toán tử cộng chuỗi
* Toán tử typeof
* Toán tử bit
  1. Độ ưu tiên
* Số học ( cộng chuooix0 – so sánh – logic – gán

1. Toán tử 3 ngôi , cú pháp

* Toán tử 3 ngôi là biểu thức điểu kiện
* Cú pháp: condition ? true\_case: false\_case;

1. So sánh null và rỗng

* Null: chỉ có giá trị stack chưa có giá trị rõ ràng bên heap
* Undefiled: giá trị có vùng nhớ nằm bên trong heap và stack

1. Toán tử == và === giống và khác

* == chỉ kiểm tra giá trị và tự chuyển đổi giá trị nếu kiểu dữ liệu khác nhau rồi mới so sánh
* === kiểm tra giá trị và kiểu dữ liệu của giá trị. K chuyển đổi về cùng kiểu dữ liệu nên 2 giá trị k cùng kiểu dữ liệu thì trả về false

1. && và ||

* Toán tử && ( and ) sẽ trả về giá trị
  + True nếu 2 giá trị là true
  + False nếu 1 trong 2 giá trị là false
* Toán tử || ( or ) sẽ trả về giá trị:
  + True nếu 1 trong 2 là false thì vẫn trả về true
  + False cả 2 giá trị false

1. Các hàm tạo thông báo trong js

* Alert()
* Confirm(): thông báo gồm cancel và ok
* Prompt()

1. Các câu lệnh điều kiện

* Câu lệnh điều kiện: câu lệnh ra quyết định
  + Lựa chọn 1 khối lệnh dựa trên 1 điều kiện cho trước
  + Gồm if và switch case
* Câu lệnh lặp: for, while, do...while
* Câu lệnh nhảy: break, continue, return

1. So sánh if và switch case

|  |  |
| --- | --- |
| **If** | **Switch \_case** |
| Có thể so sánh >,<,=… | Chỉ so sánh = = =, hoặc # |
| Biểu thức yêu cầu trả về dạng Boolean | Có thể trả về number, string, Boolean |
| Chỉ có 1 khối lệnh được thực thi | Nếu điều kiện đúng mà không có câu lệnh break thì tất cả các khối lệnh phía sau cũng được thực thi |

1. Switch case so sánh bằng === sử dụng khi điều kiện đầu vào là đơn điều kiện
2. Các biểu thức và luồng thực thi của for

* For(khởi tạo, điều kiện, tăng giảm){

Body}

1. In 5 lần 1 dòng chữ
2. Ss for, while và do while
3. Ss break và continue

* Break: kết thúc vòng lặp
* Continue: bỏ câu lệnh phía dưới continue và tiếp tục vòng lặp tiếp theo

1. Đặc điểm mảng 1 chiều

* Là 1 loại biến đặc biệt, có thể lưu nhiều giá trị đồng thời

1. Khởi tạo mảng string

* [ ] hoặc new array() có thể bỏ new

1. 1 số hàm thao tác vs mảng

* push() – thêm 1 phần tử vào cuối mảng
* unshift() – thêm vào 1 phần từ vào đầu mảng
* pop() – xóa 1 phần tử khỏi cuối mảng
* shift() – xóa 1 phần tử khỏi đầu mảng
* slice() – Sử dụng copy mảng
* Array.isArray() – Kiểm tra có phải mảng hay không?
* length – trả về kích cỡ của mảng
* sort() – sắp xếp mảng theo alphabet
* reverse() – đảo ngược mảng và trả về 1 mảng
* **- Phân biệt push() và unshift()?**
* push(): thêm vào phần tử ở cuối mảng
* unshift(): thêm vào phần tử ở đầu mảng
* **- Phân biệt push() và pop()?**
* push(): thêm vào phần tử ở cuối mảng
* pop(): lấy ra phần tử ở cuối mảng
* **- Phân biệt shift() và unshift()?**
* shift(): lấy ra phần tử ở đầu mảng
* unshift(): thêm vào phần tử ở đầu mảng

1. Tham trị và tham chiếu

* tham trị là copy giá trị của đối số, mọi sự thay đổi của tham số k ảnh hưởng đến đối số ( xảy ra với kiểu nguyên thủy)
* tham chiếu là tham số và đối số cùng chiếu đến 1 địa chỉ của vùng nhớ heap, mọi sự thay đổi của tham số làm ảnh hưởng đến đối số ( xảy ra với kiểu tham chiếu array va fobject)

1. Hàm có return và hàm k return

* K có return thì gọi là hàm void
* Có return thì trả về giá trị

1. OOP

* Object oriented progamming: là ánh xạ các sự vật hiện tượng ngoài thực tế vào trong lập trình.
* Mỗi đối tượng bao gồm:

+ thuộc tính: biến

+ hành vi: hàm

1. Phân biệt class và Object

* Class là khuôn mẫu ( bản thiết kế ) bao gồm các thuộc tính và hành vi của đối tượng
* Object là các thể hiện được tạo ra dựa trên class