1. Các câu lệnh để submit 1 file README.MD lên github là gì? Giải thích tác dụng của từng câu lệnh

Trả lời :

* git add . đưa các tập tin thêm hoặc thay đổi vào khung
* git commit tạo ra các snap slot so sánh ở local và respository, thay đổi những chổ nào cần thay đổi chứ không thay đổi hết
* git push đẩy các tập tin lên repository

1. Nêu các thẻ để tạo danh sách ? Sự khác nhau giữa <ul> và <ol> ?

Trả lời :

* Thẻ tạo danh sách <ul> <ol><li>
* ul là không theo thứ tự, ol theo thứ tự

1. Các thẻ tạo nên 1 table? Phân biệt th và td?

Trả lời :

* <table>, <tr>, <th>, <td>
* <th> in giữa, in đậm (tiêu đề) <td> thì in thường và căn trái

1. Phân biệt innerHTML và innerText?

Trả lời :

* innerHTML là cho phép in ra ở định dạng HTML
* innerText là in ra ở dạng văn bản thuần túy

1. Phân biệt get và post.

Trả lời :

* get : chỉ gửi data kích thước nhỏ, hiển thị ở bookmark và history , truyền nhanh, data gửi đi thông qua thanh URL
* post : data gửi đi được đóng gói (video, mp3, ảnh,…) không bị giới hạn và bảo mật.

|  |  |
| --- | --- |
| **Get** | **Post** |
| * Dự liệu được gửi thông qua thanh URL * Gửi dữ liệu nhanh hơn * Dữ liệu bị giơi hạn số lượng ký tự theo quy định của trình duyệt * Có thể xem lại history/ bookmark | * Dữ liệu được đóng gói và gửi đi theo kiểu gửi tin * Có thể gửi nhiều loại dữ liệu hư text, file, video… * Được sử dụng để gửi những dữ liệu nhạy cảm * Không giới hạn dung lượng |

1. Mô tả thuật toán tìm kiếm 1 phần tử trong mảng bằng mã giả?

Trả lời :

1. Mô tả thuật toán sắp xếp mảng số nguyên bằng mã giả?

Trả lời :

1. Cách khai báo biến bằng từ khóa let và var khác nhau như thế nào ?. Phạm vi của biến?

Trả lời :

* var có thể khai báo bất kỳ ở đâu và khai báo lại nhiều lần, kết quả là giá trị khai báo, gán cuối cùng. Khi khai báo var sẽ hoisted (dòng khai báo biến sẽ được trình duyệt chuyển lên đầu)
* let khai báo trước khi sử dụng , chỉ sử dụng bên trong dấu ( ) gần nhất
* Phạm vi biến là xác định phạm vi mà một biến có thể được sử dụng (truy cập)
  + Biến cục bộ (local variable) là biến được khai báo ở vị trí bên trong một hàm và chỉ truy cập bên trong hàm gọi nó
  + Biến toàn cục (glocal variable) là biến khai báo bên ngoài hàm, có thể truy cập bất cứ nơi đâu trong Javascript

1. Trong Javascript có bao nhiêu loại kiểu dữ liệu ?. Làm thế nào để xác định được biến có kiểu dữ liệu gì?

Trả lời :

* 2 kiểu dữ liệu:
  + Nguyên thủy : number, string, symbol, boolean, underfined
  + Object : null, object
* Xác định kiểu dữ liệu typeof tên(xác định kiểu)

1. Các cách ép sang kiểu String trong Javascript?

Trả lời :

* tên.toString
* String(số/mảng/object) riêng kiểu object các phần tử hiển thị là object
* array.join(‘-’) : gộp các phần tử trong mảng thành chuỗi( dùng cho mảng 1 chiều
* thực hiện cộng 1 số với 1 chuỗi

1. Các cách tạo chuỗi chứa dấu nháy ?

Trả lời :

* ‘ …. “…” ….’ Nháy kép trong nháy đơn hay ngược lại (Ví dụ: “ tên ‘thảo’ ”)
* \’ hay \” (vd: tên \’ thảo \’)

1. NaN là gì? NaN === NaN có đúng không?Vì sao?

Trả lời :

* NaN : Not a Number, kiểu dữ liệu số, biểu diễn giá trị không xác định, thường xuất hiện là số chia chuỗi, chuỗi chia số, chuỗi chia chuỗi,..
* NaN === NaN trả về false. NaN là một số không xác định nên các NaN là khác nhau

1. Toán tử ba ngôi là gì? Cú pháp?

Trả lời :

* Là câu lệnh rút gọn của câu lệnh If else
* Cú pháp : condition ? expression (condition true) : expression (condition false)

1. Toán tử == và === khác và giống nhau như thế nào?

Trả lời :

* + Đều sử dụng trong phép so sanh bằng
  + === so sánh kiểu dữ liệu và giá trị/ == chỉ so sánh giá trị

1. Phân biệt giữa 2 toán tử && và || trong JavaScript?

Trả lời :

* if(condition1&&condition2) thì để điều điều kiện if thực thi thì cần 2 điều kiện đều đúng
* if(condition1||condition2) để điều kiện if thực hiện thì cần 1 trong 2 điều kiện là đúng

1. Các hàm dùng để tạo thông báo trong Javascript ?

Trả lời :

* alert( ) hiển thị thông báo 1 lựa chọn Ok
* confirm( ) hiển thị thông báo 2 lựa chọn Ok -- Hủy
* console.log( ) hiển thị thông tin trên console
* prompt( ) nhập thông tin bên ngoài vào
* document.write ( ) hiển thị lên màn hình trình duyệt
* document.wrieln ( ) hiển thị lên màn hình trình duyệt (thêm 1 dấu cách cuối)

1. Casting data type trong JS?

Trả lời :

* Chuyển kiểu String
  + String(biến)
  + Biến.toString( ) // biến có thể là số / boolean / array
* Chuyển kiểu Boolean
  + Boolean(…) – 0, false, ‘’ hoặc “”, undefined, NaN, null trả về false / giá trị khác trả về true
* Kiểu Number
  + Number(…) –Chuổi/undefined = NaN-- số = số-- true = 1-- false/null/ ‘’/ ‘ ’ (dấu cách or ko) = 0
  + farseInt( )
  + farseFloat( )
* Chuyển đổi tự động
* String + Number/boolean… = String
* String(số) \* ( chia, trừ ) String(số) = Number

1. Có bao nhiêu Statement control trong Java Script ?

Trả lời :

* Có 3 loại : lặp, điều kiện, nhảy
  + Điều kiện :
    - if/else
    - switch/case/default + break
  + Lặp :
    - for : for( đk bắt đầu ; giới hạn ; đk tăng) {…}
    - for in : for (let x in Array) {…}
    - for of : for (let x of Array) {…}
    - for each :

Array.forEach(myFunction);

function myFunction(item, index) {  
  document.getElementById("demo").innerHTML += index + ":" + item + "<br>";  
}

* + - while : while(condition) {… }
    - do while : do {… } while (condition)
* Nhảy : continue, break, return

1. So sánh sự khác nhau giữa if và switch case?

Trả lời :

|  |  |  |
| --- | --- | --- |
|  | **Lệnh if** | **Lệnh switch/case** |
| Điều kiện | > < = khác | = hoặc khác |
| Trả về | True/false | Char,int,string,boolean |
| 1 đk kiểm tra | 1 lần | Nhiều lần |
| Điều kiện | Biểu thức | 1 giá trị duy nhât |
| 1 đk thực hiện | 1 công việc | Nhiều công việc nếu ko có break |

1. Switch case so sánh == hay ===. Đặt ra trường hợp là so sánh bằng thì khi nào sử dụng if bậc thang? Khi nào sử dụng switch case?

Trả lời :

* So sánh ===
* If bậc thang: tùy vào thang điểm để xếp loại học viên
* Switch/case: xét tháng có bao nhiêu ngày

1. Các biểu thức và luồng thực thi của for. Nếu thiếu 1 hoặc tất cả các biểu thức thì vòng for sẽ chạy như thế nào?

Trả lời :

* For ( biến khởi tạo ; đk lặp ; câu lệnh thực thi sau mỗi vòng lặp) {

Khối thực thi

}

* Lặp vô tận : thiếu 2 , thiếu cả 3
* Thiếu 1,3 : không biết số lần lặp

1. Đặt ra 1 bài toán. Xác định bài toán cần sử dụng vòng lặp nào?

Trả lời :

* If/else : người vào công ty, nếu người lạ không cho vào, nếu nhân viên cho vào
* Switch/case : xét ngày theo tháng
* For : hiển thị số học sinh trong lớp
* While : liệt kê 100 số chia hết cho 3
* Do/while : thiết kế bộ câu hỏi phỏng vấn cho máy tính, chia các câu hỏi thành các phần tử mảng. Khi chạy chương trình do/while luôn hiện ra câu hỏi đầu tiên là ‘Xin chào bạn’

1. So sánh giống và khác nhau giữa for, while và do..while?

Trả lời :

Giống nhau : Đều là vòng lặp cho phép tự động thực hiện các khối lệnh lặp đi lặp lại nhiều lần dựa trên một điều kiện cho trước. có thể thay thế cho nhau trong chương trình

|  |  |  |  |
| --- | --- | --- | --- |
|  | **For** | **While** | **Do/while** |
| Số lân lặp | Biết trước | Không biết | Không biết |
| Điều kiện | Đúng >> thực hiện | Đúng >> thực hiện | Sai >> thực hiện 1 lần |

1. So sánh null và rỗng

Trả lời :

|  |  |  |
| --- | --- | --- |
|  | **null** | **undefined** |
| Kiểu data | Object | undefined |
| Biến | Được gán null (không gì cả) | Không gán |
| null == undefined |  | |

|  |  |
| --- | --- |
| **null** | **Rỗng là undefined** |
| Là không gì cả, là kiểu object | Tồn tại nhưng không có giá trị |

1. Đặc điểm mảng một chiều trong Javascript?

Trả lời :

* Là một biến chứa nhiều giá trị
* Mỗi giá trị gọi là phần tử
* Mỗi phần tử có 1 vị trí (từ 0, 1, 2,…)
* Số phần tử gọi là độ dài mảng
* Người dùng có thể lấy phần tử thông qua vị trí mảng

1. Các cách khởi tạo một mảng kiểu String trong JavaScript?

Trả lời :

* let tên\_biến = [string1, string2, …];
* let tên\_biến = new Array (string1, string2, …);
* let tên\_biến = Array (string1, string2, …);
* let tên\_biến = ‘string’; // mỗi ký tự là một phần tử

1. Câu hỏi :

- 1 số hàm thao tác với mảng?

Trả lời : toString – join( ) – push( ) – pop( ) – shift( ) – unshift( ) – concat( ) – splice( ) – slice ( ) – split ( ) – indexOf( )

- Phân biệt push() và unshift()?

Trả lời : push() thêm phần tử vào cuối mảng / unshift() thêm phần tử vào đầu mảng

- Phân biệt push() và pop()?

Trả lời : push () thêm phần tử vào cuối mảng / pop () xóa phần tử cuối mảng

- Phân biệt shift() và unshift()?

Trả lời : unshift()?thêm phần tử vào đầu mảng / shift() xóa phần tử đầu mảng

1. Phân biệt tham trị và tham chiếu trong Javascript?

Trả lời :

|  |  |  |
| --- | --- | --- |
|  | **Tham trị** | **Tham chiếu** |
| Kiểu dữ liệu truyền | Nguyên thủy | Object |
| Khi truyền giá trị vào hàm,  biến bên ngoài | Không thay đổi | Thay đổi |

1. Phân biệt giữa Hàm có return và hàm không có return?

Trả lời :

|  |  |
| --- | --- |
| **Return** | **Ko return** |
| Hàm có giá trị trả về, phù hợp khi tái sử dụng | Hàm không có giá trị trả về dùng để lặp đi lặp lại 1 khối lệnh  Không thể thay đổi nội dung thông báo khi tái sử dụng |
| Vd: truyền vào a,b trả về tổng a,b | Vd: sử dụng onclick để gọi hàm |

1. So sánh break, continue và return

Trả lời : Đều là câu lệnh nhảy

|  |  |  |
| --- | --- | --- |
| **break** | **continue** | **return** |
| -Dừng vòng lặp khi gặp lệnh break  -Không trả về giá trị khi kết thúc vòng lặp | -Nhảy tới vòng lặp tiếp theo khi gặp lệnh continue  -Không trả về giá trị khi kết thúc vòng lặp | -Thoát khỏi hàm khi gặp câu lệnh return  -Trả về giá trị cho hàm đó |

1. Lập trình hướng đối tượng là gì?

Trả lời :

Object Oriented Programming(OOP) là kỹ thuật mà LTV tạo ra đối tượng từ bên ngoài vào trong ngôn ngữ lập trình

Đối tượng : thuộc tính (dữ liệu, tính chât của đối tượng), hành vi (khả năng, hành động của đối tượng thể hiện)

1. Các đặc điểm trong Lập trình hướng đối tượng. Đưa ra 1 ví dụ 1 trong 4 tính chất?

Trả lời :

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **Định nghĩa** | **Ví dụ** |
| Tính đóng gói | Truy, xuất vẫn đảm bảo riêng tư của đối tượng | Class nhân viên ( tên, tuổi, lương) chỉ chia sẽ tên tuổi nhưng không chia sẽ tiền lương |
| Tính kế thừa | Có thể chia sẻ, mở rộng mà không cần định nghĩa lại | Kế thừ giữa cha con: con giống cha về tóc, tay chân, chơi thể thao giỏi |
| Tính trừ tượng | Loại bỏ những thuộc tính, hành vi không cần thiết chỉ giữ lại những thuộc tính, hành vi cần dùng | 1>Một cái điện thoại mua về chỉ sử dụng chức năng nghe gọi, những chức năng khác không dùng tới  2>Trong lớp, chỉ cần biết thông tin tên, tuổi của học viên. Ko cần lấy thông tin gia đình |
| Tính đa hình | Cùng một hành vi nhưng mỗi đối tượng thể hiện khác nhau | 1>>1 nhóm vận động viên, có người chạy nhanh có người chạy chậm  2>>Trong lớp là học viên, nhưng về nhà là con cái, hoặc mua hang là khách hang |

1. Constructor là gì? Trong 1 class có nhiều hơn 1 contrustor được hay không?

Trả lời :

* Khởi tạo thuộc tính cho class
* Có nhiều constructor sẽ gây lỗi SyntaxError
* Mỗi constructor có thể sử dụng từ khóa supper để gọi đến constructor của class cha
* Nếu bạn không định nghĩa constructor thì constructor mặc định sẽ được sử dụng

1. Những phương thức nào cho phép tương tác với chuỗi

Trả lời :

* charAt(): Trả về ký tự tại một vị trí được chỉ định
* concat(): Nối hai hoặc nhiều chuỗi để tạo thành một chuỗi mới
* endsWith(): Kiểm tra xem liệu một chuỗi có kết thúc giống với một chuỗi khác hay không
* startsWith(): Kiểm tra xem liệu một chuỗi có kết thúc giống với một chuỗi khác hay không
* indexOf(): Trả về vị trí của một chuỗi ở trong một chuỗi khác. Nếu chuỗi này xuất hiện nhiều lần thì lấy vị trí của lần xuất hiện đầu tiên.
* lastIndexOf(): Trả về vị trí của lần xuất hiện cuối cùng một chuỗi ở trong một chuỗi khác.
* replace(): Thay thế một chuỗi bằng một chuỗi khác.
* split(): Tách một chuỗi thành một mảng các chuỗi
* substr(): Lấy một phần nhỏ của chuỗi
* toLowerCase(): Chuyển một chuỗi thành viết thường
* toUpperCase(): Chuyển một chuỗi thành viết hoa
* trim(): Loại bỏ các dấu khoảng trắng ở hai đầu của chuỗi

1. Các câu lệnh để vẽ 1 hình tròn. Giải thích từng câu lệnh?

Trả lời :

* Trong HTML
* <canvas id="myCanvas" width="1368px" height="768px" style="border:1px solid #d3d3d3;"></canvas>

// tạp thể <canvas> là khung hiển thị canvas

* Trong thẻ <script>
* var ctx = ***document***.getElementById("myCanvas").getContext("2d");

// tạo đường truyền ctx trỏ lới id ‘myCanvas’ lấy nội dung 2d

* ctx.beginPath();

// bắt đầu

* ctx.arc(***x***, y, ***radius***, 0, 2 \* ***Math***.PI);

// xác định vị trí trục x,y , bán kính radius, bắt đầu vẽ từ vị trí 0 đến 2PI (1 vòng tròn)

* ctx.fill();

// vẽ hình

1. Trình bày ý tưởng của 1 thuật toánbất kì?

Trả lời :

Chuyển đổi tiền tệ Dola > VND

|  |  |
| --- | --- |
| **Pseudo-Code** | **Flow-Chart** |
| Begin  Input Dola  VND = Dola \* 23000  Display “VND”  End | Display “VND”  VND = Dola \* 23000  Input Dola |

1. Cho 1 bài toán liên quan đến ++ trước và ++ sau. Yêu cầu dự đoán kết quả, và tại sao lại có kết quả như vậy?

Trả lời :

Let x = 5;

Let y = x++; // y = 5 , x = 6

Let z = ++x; // z = 7 , x = 7

Let sum = x + y + z; // 19;

1. Các loại toán tử trong js? Cho biết độ ưu tiên của các toán tử trong một biểu thức?

Trả lời :

* Toán học : + - \* / = % , num++ và ++ num
* So sánh : && , || , == , === , != , !== , > , < , >= , <=
* Logic : += , -= , /= , \*= , ++ , --
* typeof : typeof phanTuXacDinhKieu

