1. **Repository là gì? có bao nhiêu loại repository?**

• Thường được gọi ngắn gọn là repo

• Repository là nơi chứa toàn bộ mã nguồn

• Repository bao gồm toàn bộ các file và lịch sử của các file đó

• Repository chứa tất cả các commit

• Có 2 loại repository:

• Local Repository: Ở trên máy của lập trình viên

• Remote Repository: Ở trên một máy chủ chia sẻ (chẳng hạn như GitHub)

1. **Các câu lệnh căn bản để làm việc với git?**

• git clone: Sao chép một repository

• git init: Khởi tạo một repository

• git add: Thêm file vào local repository

• git commit: Thực hiên lưu file và ghi lại lịch sử ở snapshort

• git push: Đẩy file từ local repository (folder ở máy tính) lên remote repository (github)

1. **Giải thuật là gì? các cách để biểu diễn thuật toán?**

Thuật toán/giải thuật bao gồm các chỉ thị để giải quyết một vấn

đề

**• Có thể sử dụng các cách khác nhau để mô tả thuật toán, các**

**cách thông dụng là:**

- ***Mã giả (pseudo-code):*** Ngôn ngữ tự nhiên được viết bằng tiếng anh để diễn đạt chương trình.

*Ưu điểm:*

• Đơn giản, không cần kiến thức về cách biểu diễn (lưu đồ, ngôn ngữ lập

trình)

• *Nhược điểm*

• Dài dòng, không cấu trúc

• Đôi lúc khó hiểu, không diễn đạt được thuật toán

*Mã giả: Ví dụ*

*• Giải thuật giải phương trình bậc nhất dạng ax + b = 0 như sau:*

*BEGIN*

*INPUT a, b*

*IF a = 0 THEN*

*IF b = 0 THEN*

*PRINT “Phương trình vô số nghiệm”*

*ELSE*

*PRINT “Phương trình vô nghiệm”*

*END IF*

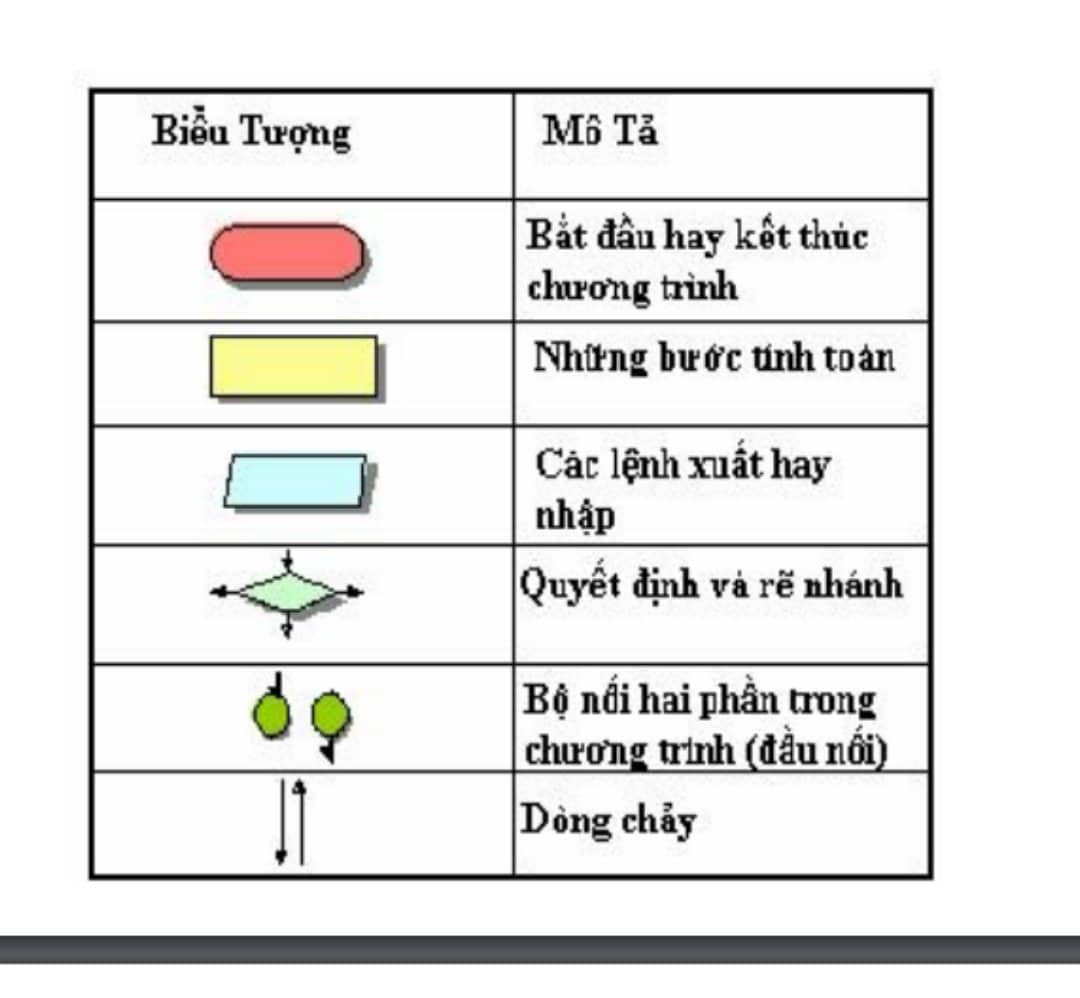
*ELSE*

*PRINT “Phương trình có nghiệm x = -b/a”*

*END IF*

*END*

***- Lưu đồ (flowchart)*** sử dụng hình vẽ để diễn đạt chương trình.



*- Ngôn ngữ lập trình*

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

Begin-> Input N dãy số a1,a2,….aN. i=0 number=?

While(i<=N)

Do

If a[i]=number then number có trong mảng

End if

Else number không có trong mảng

i++

End Do

Display ->End

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

Begin

Input N dãy a1,a2….an i=0

For ( i = 0; i < num.length; i++)

{ for (let j = 0; j < num.length - i; j++)

{ if (num[j] <= num[j + 1])

{ let b = num[j];

num[j] = num[j + 1];

num[j + 1] = b;

} } }

Display dãy

End

1. **Trình bày ý tưởng của 1 thuật toán bất kì?**

Begin -> Input (a1,a2,….aN),N,max=a1,i=2

While (i<=n) D0

If (a1>max) then max=ai

End if

I=i+1

End Do -> Output

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

Các thẻ để tạo danh sách:

* Có thứ tự

VD:

<ol>

<li>Coffee</li> 1.Coffee

<li>Tea</li> 2. Tea

<li>Milk</li> 3. Milk

</ol>

* Không có thứ tự

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

*Kết quả:*

* Coffee
* Tea
* Milk

Khác nhau:

|  |  |
| --- | --- |
| **<ul>** | **<ol>** |
| unordered list | ordered list |
| các mục con của nó sẽ không được sắp xếp theo thứ tự | các mục con của nó được sắp xếp theo thứ tự |
| Được đánh dấu bằng một ký tự đặc trưng | Hiển thi bằng số hoặc chữ cái |

1. **Các thẻ tạo nên 1 table?**

• <table>: Khai báo một bảng

• <tr>: Khai báo một dòng (table row)

• <th>: Khai báo một ô tiêu đề (table head)

• <td>: Khai báo một ô dữ liệu (table data)

**Phân biệt th và td?**

<th> in đậm căn giữa

<td> viết thường, căn trái

1. **Phân biệt innerHTML và innerText?**

|  |  |
| --- | --- |
| **innerHTML** | **innerText** |
| Định dạng theo HTML | Định dạng theo văn bản |
| Sẽ xử lý một thẻ HTML nếu được tìm thấy trong một chuỗi | Không xử lý thẻ HTML nếu được tìm thấy trong một chuỗi |

1. **Nêu một số thẻ HTML cơ bản mà bạn biết?**

* Tiêu đề: <h1> đến <h6>
* Tạo văn bản:

+ <p>: rớt dòng

+ <span>: nhóm các phần tử nội tuyến lại với nhau, tiện cho việc định dạng CSS

+ <br>: Ngắt dòng

+ <hr>: Tạo đường ngang

* Tạo liên kết

+<a>

* Tạo danh sách:

<ul>

<ol>

1. **Các thuộc tính cơ bản của thẻ form?**

* Thuộc tính action: có giá trị là một URL, quy định nơi dữ liệu được

gửi đến

* Thuộc tính name: quy định tên của form, sau đó có thể được dung để

truy xuất đến các giá trị trong form.

* Thuộc tính method: (get/post) được sử dụng để quy định cách thức dữ liệu được gửi lên server.

1. **Phân biệt phương thức get và post trong thẻ form**

|  |  |
| --- | --- |
| **Get** | **Post** |
| Hiển thị dữ liệu ở URL | Dữ liệu được gửi đi KHÔNG hiển thị dữ liệu ở URL |
| Kém bảo mật | Bảo mật hơn |
| Dữ liệu gửi đi theo dạng text(văn bản) | Đa dạng kiểu dữ liệu gửi đi (văn bản, âm thanh,hình ảnh) vì post gửi thông tin thông qua HTTP request |

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

|  |  |
| --- | --- |
| **let** | **var** |
| Không khai báo lại được | Có thể khai báo lại |
| Không có cơ chế hoisting | Có cơ chế hoisting(sử dụng trước, khai báo sau) |
| Block scope (phạm vi tác dụng trong 1 khối lệnh) | Function scope (phạm vi tác dụng trong 1 function) |

Có hai loại phạm vi biến:

* Biến cục bộ (local scope)
* Biến toàn cục (global scope).

Biến toàn cục được khai báo ngoài hàm , trong khi đó biến cục bộ được lưu khai báo bên trong một hàm.(Nên với cùng một tên biến, ta có thể khai báo ở những hàm khác nhau).

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

**Các kiểu dữ liệu:**

+Kiểu nguyên thủy

* String: Khai báo chuỗi các kí tự (vd: “a,b,c”, ‘a’)
* Number: khai báo số nguyên, số thập phân, số âm, số dương
* Boolean: true & false
* Underfiled: 1 biến chưa gán giá trị.
* Null: không chứa giá trị nào (object)
* Dữ liệu tham chiếu (object)
* Symbol

+Kiểu đối tượng:

* Array (mảng)
* Object: Do chương trình cung cấp hoặc lập trình viên tự định nghĩa

Để kiểm tra kiểu dữ liệu của một biến, bạn có thể sử dụng typeof.

* Có hai dạng: dạng toán tử typeof x hoặc ở dạng hàm typeof(x).
* Kết quả trả về là một string ứng với tên của kiểu dữ liệu.
* typeof null trả về object. Đây là một lỗi sơ khai của JavaScript. Thực chất, null không phải là object.

*Có 3 cách để xác định một biến trong JavaScript:*

* ***Var****– Câu lệnh các biến JavaScript được sử dụng để khai báo một biến và ta có thể khởi tạo giá trị của biến đó. Ví dụ: var a =10; Khai báo biến được xử lý trước khi code được thực thi.*
* ***Const****– Hàm const không được cho phép sửa đổi đối tượng (object) mà chúng được gọi. Khi một hàm được khai báo là const, nó có thể được gọi trên bất kỳ kiểu đối tượng nào.*
* ***Let****– Đây là tín hiệu cho thấy biến có thể được gán lại, chẳng hạn như bộ đếm trong vòng lặp hoặc hoán đổi giá trị trong thuật toán. Nó cũng báo hiệu rằng biến sẽ chỉ được sử dụng trong block mà nó được xác định.*

1. **Các cách tạo chuỗi chứa dấu nháy ?**

ba cách để viết một chuỗi

- Dấu nháy đơn (''),

- Dấu nháy kép ("")

- Dấu nháy (``)

Vd:

`This string uses backticks.`;

Output

This is a string in the console.

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

NaN là một dạng viết tắt của “Not a Number”.

Vì NaN luôn so sánh không bằng với bất kỳ số nào, kể cả NaN, nó thường được sử dụng để chỉ ra điều kiện lỗi cho một hàm mà đáng ra phải trả về một số hợp lệ. Khi một chuỗi hoặc một cái gì đó khác đang được chuyển đổi thành một số và điều đó không thể được thực hiện, thì ta sẽ thấy NaN.

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?**

Tiền tố ++a và hậu tố a++

let a=100 ; let b=200; let c=++a + b++

console.log(c)=301

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

*Các loại toán tử:*

* Số học (+, -, \* , /, %)
* So sánh (< , > , >=, <= , = = , = = = , !=, != =)
* Logic (&&, ||, !)
* Gán ( +=, -=, \*=, /=, %=, =)
* Toán tử cộng chuỗi
* TT type of
* TT bit

*Độ ưu tiên của các toán tử:*

Số học (cộng chuỗi) – so sánh – logic – gán.

1. **Toán tử ba ngôi là gì? Cú pháp?**

Toán tử 3 ngôi đánh giá một biểu thức dựa vào 1 điều kiện cho trước.

Toán tử 3 ngôi còn được gọi là biểu thức điều kiện.

*Cú pháp:*

**condition ? true\_case: false\_case**

condition: biểu thức điều kiện dùng để đánh giá

true\_case: biểu thức sẽ được sử dụng trong điều kiện true

false\_case: biểu thức sẽ được sử dụng trong điều kiện false

1. **So sánh null và rỗng**

*Undefined là một biến đã được khai báo nhưng chưa được gán giá trị.*

*Null là một giá trị gán. Nó có thể được gán cho một biến dưới dạng đại diện không có giá trị. Ngoài ra, undefined và null là hai kiểu riêng biệt: undefined là một kiểu riêng biệt (kiểu dữ liệu không xác định) trong khi null là một đối tượng (object).*

Cách giải thích của anh Tiến:

* Rỗng: giá trị có vùng nhớ nằm bên vùng nhớ stack.
* Null: chỉ có giá trị stack, chưa có giá trị rõ ràng bên heap (chưa xác định rõ ràng)

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

Toán tử “==” chỉ kiểm tra giá trị và tự chuyển đổi nếu các giá trị có kiểu dữ liệu khác nhau trước khi so sánh.

Mặt khác, toán tử “===” kiểm tra cả giá trị và kiểu dữ liệu của giá trị. Nó không chuyển đổi các giá trị khi so sánh, nên nếu hai giá trị không cùng kiểu dữ liệu thì kết quả sẽ trả về là false.

Ví dụ: nếu bạn so sánh một số với một chuỗi có ký tự số, “==” cho phép điều đó, nhưng “===” không cho phép, bởi nó không chỉ kiểm tra giá trị mà còn kiểm tra kiểu của hai biến, nếu hai biến không cùng loại, “===” trả về false, trong khi “==” trả về true.

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

*Toán tử && (and) sẽ trả về giá trị:*

true: nếu hai giá trị là true

false: nếu hai giá trị là false, hoặc một giá trị là true và một giá trị là false

*Toán tử || (or) sẽ trả về giá trị:*

true: nếu hai giá trị là true, hoặc một giá trị là true và một giá trị là false

false: nếu hai giá trị là false

1. **Các hàm dùng để tạo thông báo trong JavaScript ?**

Hàm alert(), confirm() và prompt() đây là ba hàm thường được sử dụng tạo hộp thoại thông báo và lấy thông tin từ người dùng.

1. **Có bao nhiêu Statement control(câu lệnh điều khiển) trong Java Script ?**

Có 3 loại:

* **Câu lệnh điều kiện**= câu lệnh ra quyết định.

+Lựa chọn 1 khối lệnh dựa trên việc đánh giá 1 điều kiện cho trước

+ Bao gồm: if và switch case.

* **Câu lệnh lặp:** for, while, do- while
* **Câu lệnh nhảy**: break, continue, return

1. **So sánh sự khác nhau giữa 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 == hay ===. Đặt ra trường hợp là so sánh bằng thì khi nào sử dụng if bậc thang? Khi nào sử dụng switch case**

Switch case so sánh = = =. Sử dụng SC khi điều kiện đầu vào đơn điều kiện(1 biến so sánh)

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?**

**Cấu trúc:**

for (khởi tạo; điều kiện;tăng giảm){

//body

}

* 1. Khởi tạo: các câu lệnh khởi tạo được thực thi
  2. Điều kiện: Biểu thức điều kiện được đánh giá. Nếu điều kiện đúng thì khối lệnh bên trong thân vòng lặp được thực thi. Nếu điều kiện sai thì vòng lặp kết thúc
  3. Body: Khối lệnh bên trong thân vòng lặp được thực thi
  4. Tăng giảm: Các câu lệnh tang giảm được thực thi
  5. Bắt đầu vòng lặp mới từ bước 2 – 3 - 4

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

Viết chương trình hiển thị chuỗi “The number is N” 5 lần. Với N sẽ hiển thị từ 0 đến 5 **-> lần sử dụng vòng lặp for**

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

|  |  |  |
| --- | --- | --- |
| **for** | **while** | **do-while** |
| Biết trước số lần lặp. | Không biết trước số lần lặp | Không biết trước số lần lặp |
| Xét điều kiện trước khi lặp | Xét điều kiện trước khi lặp | Chạy trước ít nhất 1 lần dù điều kiến sai. |
| **Cấu trúc:**  *for (khởi tạo; điều kiện;tăng giảm){*  *//body}* | **Cấu trúc:**  *while (điều kiện){*  *//body*  *}* | **Cấu trúc:**  *do{*  *//body*  *}while()* |

1. **So sánh break và continue**

Break: kết thúc vòng lặp

Continue: bỏ hết câu lệnh phía dưới, tiếp tục chu kì tiếp theo (chuyển đến vòng lặp tiếp theo)

1. **Đặc điểm mảng một chiều trong JavaScript**

Một loại biến đặc biệt, nó có thể lưu trữ nhiều giá trị đồng thời.

Vd: câu lệnh bên dưới dùng để tạo một mảng có tên là mobile và mảng đó chứa năm giá trị:

let mobile = ["HTC", "Nokia", "SamSung", "LG", "Apple"];

HTC

Nokia

SamSung

LG

Apple

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

Có 2 cách:

* Sử dụng []

Vd: let sinhVien=[‘Nguyen Van A’, ‘Nguyen Van B’];

* Sd new Array() có thể lược bỏ new

1. **- 1 số hàm thao tác với 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. **Phân biệt tham trị và tham chiếu trong JavaScript**

|  |  |
| --- | --- |
| **Tham trị** | **Tham chiếu** |
| Tham số coppy giá trị của đối số | Tham số và đối số cùng chiếu đến 1 địa chỉ ở vùng nhớ Heap |
| Mọi sự thay đổi của tham số không làm ảnh hưởng đến đối số (value type) | Mọi sự thay đổi của tham số làm ảnh hưởng đến đối số  (reference type) |
| Xảy ra với kiểu nguyên thủy(number, Boolean,symbol, str..) | Xảy ra với kiểu tham chiếu: arr, object |

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

- Hàm không trả về giá trị (void)

- Hàm có giá trị trả về: return

Return: trả về giá trị ngay lời gọi hàm & kết thúc hàm.

1. **Lập trình hướng đối tượng là gì**

**Lập trình hướng đối tượng = OOP(object oriented programming)**

-Là kỹ thuật đem các sự vật ngoài thực tế vào trong lập trình..

-Mỗi đối tượng bao gồm:

+ Thuộc tính (property):mô tả thuộc tính của đối tượng.

+ Hành vi (Behavior): mô tả hành vi,hành động của đối tượng.

**Vd:** Trong thực tế: học viên

* Thuộc tính: tên, ngày sinh, giới tính
* Hành vi: ăn, uống, chạy…

Trong lập trình:

- Thuộc tính: biến

- Hành vi: Hàm

1. **Phân biệt class và object**

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

1. **Các đặc điểm trong Lập trình hướng đối tượng. Đưa ra 1 ví dụ 1 trong 4 tính chất. (lưu ý cần phải trình bày được 4 tính chất bằng tiếng anh)**

* Kế thừa (inheritance): lớp con thừa hưởng lớp toàn bộ các thuộc tính và hành vi của lớp cha.

VD:

* Đóng gói (encapsulation): che giấu dữ liệu ở Object. Muốn truy xuất hay sửa đổi phải thông qua getter/setter.

VD: những thông tin cân bảo mật.

* Tính trừu tượng (Abstraction): ẩn đi các chi tiết triển khai của hành động, chỉ thể hiện chức năng đến với người dung.

VD: hành vi đi

* Đa hình (Polymorphism): 1 hành vi thể hiện theo nhiều cách khác nhau.

*(triến khai ra chi tiết trong tính trừu tượng)*

Vd: đi bằng bao nhiêu chân

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

- Constructor là một hàm đặc biệt dùng để khởi tạo đối tượng.

- Trong JS mỗi class có 1 constructor duy nhất.

- Cú pháp: constructor() {.....}

- Được gọi thông qua từ khóa new.

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

-string.length //KIỂM TRA ĐỘ DÀI chuỗi

-string.indexOf('tp') //TÌM CHỮ ở vị trí

-string.slice(3) //CẮT CHUỖI từ vị trí thứ 3 đến hết chuỗi

-string.replace('tp','city') // THAY CHỮ 'tp’ xuất hiện lần đầu tiên ='city'

-string.replace(/tp/g,'city')+'<br>') // thay tất cả chữ 'tp' ='city'

-string.toLowerCase() // thay chữ hoa thành chữ thường

-string.toUpperCase() // thay chữ thường thành chữ hoa

-string.trim() //loại bỏ ký tự khoảng trắng ở 2 đầu chuỗi

-string.split(' ') //chuyển chuỗi thành mảng,chia thành từng cụm

-string.split('') //chuyển chuỗi thành mảng, chia thành từng ký tự

-string.charAt('5') //lấy ra ký tự từ vị trí index

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

let canvasObj = document.getElementById("canvasObjectHTML");  
let ctx = canvasObj.getContext("2d");  
  
// 1. Suy nghi y tuong  
ctx.arc(200, 200, 100, 0, 2 \* Math.PI);  
  
  
// 2. Ve vien  
ctx.strokeStyle = "red";  
ctx.stroke();  
  
// 3. Chon mau  
ctx.fillStyle = "yellow";  
  
// 4. To mau  
ctx.fill();