A-LEADE IT CAMBODIA

JS question

Answer

1. What is the difference between Block element and Inline element?
   Answer: - A block-level element is an HTML element that begins a new line on a Web page and extends the full width of the available horizontal space of its parent element.
   - Inline element is an element that cannot start on a new line.

2. CSS Sprites are a means of combining multiple images into a single image file for use on a website, to help with performance. We need to use CSS sprites because we want to have the performance of our website.

3. Responsive CSS is the CSS style that we create on the website that look good when to display all device (Computer, tablets and mobile devices).

4. GET and POST:
   - GET: It requests the data from a specified resource

     Some other features of GET requests are:

     - It remains in the browser history
     - It can be bookmarked
     - It can be cached
     - It have length restrictions
     - It should never be used when dealing with sensitive data
     - It should only be used for retrieving the data

   - POST: It submits the processed data to a specified resource.

     Some other features of POST requests are:

     - This requests cannot be bookmarked
     - This requests have no restrictions on length of data
     - This requests are never cached
     - This requests do not remains in the browser history

5. difference between asynchronous HTTP request and synchronous HTTP request:
   - **Synchronous request**: involves a client that waits for the server to response to a request. Request are able to flow in both directions, to and from. Essentially it means that synchronous request is a two way communication. i.e. Sender sends a message to receiver and receiver receives this message and gives reply to the sender. Sender will not send another message until it gets a reply from receiver.

   - **Asynchronous request**:
     Asynchronous request involves a client that does not wait for a response from the server. An event is used to trigger a request from a server. So even if the client is down, the request will complete successfully. Asynchronous request means that, it is a one way communication and the flow of communication is one way only.

6. Explain these terminologies:
   - **Brute force** (also known as brute force cracking) is a trial and error method used by application programs to decode encrypted data such as passwords or Data Encryption Standard (DES) keys.
   - **Cross-Site Scripting (XSS):** is one of the most common application-layer web attacks. XSS vulnerabilities target scripts embedded in a page that are executed on the client-side (in the user's web browser) rather than on the server-side.
   - **SQL Injection:** is an application security weakness that allows attackers to control an application's database – letting them access or delete data, change an application's data-driven behavior, and do other undesirable things – by tricking the application into sending unexpected SQL commands.
   - **Denial of Service (DoS):** is an attack targeting the availability of web applications. Unlike other kinds of attacks, DoS attacks' primary goal is not to steal information but to slow or take down a web site.
   - **Man-in-the-Middle Attacks:** is one in which the attacker secretly intercepts and relays messages between two parties who believe they are communicating directly with each other.

7. meanings of these server response codes:
   - **404**: Not found: This error message is shown when a site or folder on a server are requested but cannot be found at the given URL.
   - **500**: Internal Server Error: The server encountered an unexpected condition which prevented it from fulfilling the request.
   - **200:** OK: The request was fulfilled.
   - **304**: Not Modified: If the client has done a conditional GET and access is allowed, but the document has not been modified since the date and time specified in If-Modified-Since field, the server responds with a 304 status code and does not send the document body to the client.

8. **Security Token**: A security token is a portable device that authenticates a person's identity electronically by storing some sort of personal information. The owner plugs the security token into a system to grant access to a network service. Security tokens are issued by Security Token Services (STS), which authenticate the person's identity.
9. **MVC explanation: MVC** is the name of a methodology or design pattern for successfully and efficiently relating the user interface to underlying data models. The MVC pattern has been heralded by many developers as a useful pattern for the reuse of object code and a pattern that allows them to significantly reduce the time it takes to develop applications with user interfaces.

A **Model**, which represents the underlying, logical structure of data in a software application and the high-level class associated with it. This object model does not contain any information about the user interface.

View, which is a collection of classes representing the elements in the user interface (all of the things the user can see and respond to on the screen, such as buttons, display boxes, and so forth)

Controller, which represents the classes connecting the model and the view, and is used to communicate between classes in the model and view.

Exercise

1. We have this JSON list

```
var superHeroList = [
        {id:1, name:"Oliver Queen", title:"Arrow"},
        {id:2, name:"Clark Kent", title:"Superman"},
        {id:3, name:"Steve Rogers", title:"Captain America"},
        {id:4, name:"Peter Parker", title:"Spiderman"},
];
```

1.1 What is the result of the following statements

1.1.1 superHeroList[2].title

1.1.2 superHeroList[3].name

Answer: 1.1 the result of the following statements:

1.1.1 superHeroList[2].title is: Captain America

1.1.2 superHeroList[3].name is : Peter Parker

1.2 Create function body for searchByFieldName(list, fieldname, keyword)

that this function can be used as the following

```
var spiderMan = searchByFieldName(superHeroList, "name","Peter Parker");
var superMan = searchByFieldName(superHeroList, "title", "Superman");
console.log(JSON.stringify(spiderMan));
// output result: {id:4, name:Peter Parker, title:Spiderman}
console.log(JSON.stringify(superMan))
// output result: {id:2, name:Clark Kent, title:Superman}
```

Answer:

```
function searchByFieldName( superHeroListObj, fieldName, keyword ){
        for( objs in superHeroListObj ){
            if(superHeroListObj[objs]!=null && superHeroListObj[objs][fieldName] === keyword ){
                return superHeroListObj[objs];
            }
        }}
```