FSL-Question Bank Solution (PT-2)

Chapter 4: COOKIES AND BROWSER DATA

1.Define cookies/importance of cookies.

Ans:

- A cookie is an amount of information that persists between a server-side and a client-side. A web browser stores this information at the time of browsing
- A cookie contains the information as a string generally in the form of a name-value pair separated by semi-colons. It maintains the state of a user and remembers the user's information among all the web pages.
- Cookies are data, stored in small text files, on your computer.
- Cookies were invented to solve the problem "how to remember information about the user".
- When a user visits a web page, his/her name can be stored in a cookie.
- Next time the user visits the page, the cookie "remembers" his/her name.
  Cookies are saved in name-value pairs like:
  username = John Doe

2. Writing/reading/deletion of cookies/expires date of cookies-syntax and example.

Ans:

- Writing of cookie.
  document.cookie = “name=value”;
- Reading of cookie.
  document.cookie = “username=james”;
- Deletion of cookie.
  document.cookie = “username=king; expires=Mon, 3-Sep-2025 09:20:00 UTC”;
- Expires date of cookie.
  document.cookie = “username=bond; expires=Tue, 4-Jan-2025 08:40:00 GMT”;

Example:

<html>

<body>

3. Write a program to open window.

Ans:

```
<html>
<head>
<script>
function openWin(){
window.open("https://www.google.com");
}
</script>
</head>
<body>
<form>
<input type="button" value="Open Window" onclick="openWin()"/>
</form>
</body>
</html>
```

5. Enlist and explain any 3/4 specs attributes of open method of window

Ans:

Spec attributes of open method of window are –

A comma-separated list of items, no whitespaces.

- Fullscreen = yes|no|1|0
  Whether or not to display the browser in full-screen mode. Default is no. A window in full-screen mode must also be in theater mode.
- height = pixels
  The height of window. Min value is 100px.
- scrollbar = yes|no|1|0
  Weather or not to display scrollbars.
- status = yes|no|1|0
  Weather or not to display status bar.
  Example:

```
<html>
<head>
<script>
function openWin() {
 window.open("https://www.w3schools.com","status=no, menubar=yes, scrollbars=yes, height=400");
}
</script>
</head>
<body>
<form>
 <input type="button" value="Open Window" onclick="openWin()">
</form>
</body>
</html>
```

6. Write a program for Blur and focus a new window.

Ans:

```
<html>
<head>
<script>
var myWindow;
function openWin() {
 myWindow = window.open("https://www.google.com", "_blank", "width=400, height=200");
 }
function blurWin() {
 myWindow.blur();
}
function focusWin() {
 myWindow.focus();
}
</script>
</head>
<body>
<input type="button" value="Open new window" onclick="openWin()">
<input type="button" value="Blur new window" onclick="blurWin()">
<input type="button" value="Focus new window" onclick="focusWin()">
</body>
</html>
```

7. Explain screen object with its 3/4 properties.

Ans:

The screen object contains information about the visitor's screen. Properties of the screen objects are like availHeight, availWidth, colorDepth, pixelDepth, height, width, etc.

- availHeigth
  i. The availHeight property returns the height of the user's screen.
  ii. The availHeight property returns the height in pixels.
  iii. The availHeight property returns the height minus interface features like the Windows Taskbar.
  Syntax: screen.availHeight
  Example:

```
<html>
<body>
<p id="demo"></p>
<script>
height = screen.availHeight;
document.getElementById("demo").innerHTML = height + "px";
</script>
</body>
</html>
```

- availWidth
  i. The availWidth property returns the width of the user's screen.
  ii. The availWidth property returns the width in pixels.

iii. The availWidth property returns the width minus interface features like the Windows Taskbar.
Syntax: screen.availWidth
Example:

```
<html>
<body>
<p id="demo"></p>
<script>
let width = screen.availWidth;
document.getElementById("demo").innerHTML = width + "px";
</script>
</body>
</html>
```

- colorDepth
  i. The colorDepth property returns the screen's color depth.
  ii. The colorDepth property returns the depth in bits per pixel.
  iii. The colorDepth property is read-only.
  Syntax: screen.colorDepth
  Example:

```
<html>
<body>
<p id="demo"></p>
<script>
let depth = screen.colorDepth;
document.getElementById("demo").innerHTML = depth + " bits per pixel";
</script>
</body>
</html>
```

- height
  i. The height property returns the total height of the user's screen.
  ii. The height property returns the height in pixels.
  iii. The height property is read only.
  Syntax: screen.height
  Example:

```
<html>
<body>
<p id="demo"></p>
<script>
let height = screen.height;
document.getElementById("demo").innerHTML = height + "px";
</script>
</body>
</html>
```

8. Explain location object with its properties.

Ans:

- The location object contains information about the current URL.
- The location object is a property of the window object.
- The location object is accessed with:
  window.location or just location

Property	Description
href	Sets or returns the entire URL
origin	Returns the protocol, hostname and port number of a URL
pathname	Sets or returns the path name of a URL
protocol	Sets or returns the protocol of a URL

9. Write a script to demonstrate setTimeout(), setinterval() and clearTimeout().

Ans:

- setTimeout()

```
<html>
<body>
<h2>JavaScript Timing</h2>
<button onclick="setTimeout(myFunction, 3000);">Try it</button>
<script>
function myFunction() {
    alert('Hello');
}
</script>
</body>
</html>
```

- setInterval()

```
<html>
<body>
<h2>The setInterval() Method</h2>
<p id="demo"></p>
<script>
setInterval(displayHello, 1000);
function displayHello() {
    document.getElementById("demo").innerHTML += "Hello";
}
</script>
</body>
</html>
```

- clearTimeout()

```
<html>
<body>
<h2>setTimeout() Function.</h2>
<button onclick="myVar = setTimeout(myFunction, 3000)">Try it</button>
<button onclick="clearTimeout(myVar)">Stop it</button>
<script>
function myFunction() {
    alert("Hello");
}
</script>
</body>
</html>
```

Chapter 5:

1. Write a script to call a child window using javascript function.

Ans:

frame.html

```
<html>
<head>
<frameset rows="50%,50%">
<frame src="webpage1.html" name="topPage"/>
<frame src="webpage2.html" name="bottomPage"/>
</frameset>
</head>
</html>
```

webpage1.html

```
<html>
<head>
<script>
function ChangeContent(){
    alert("Function Called!");
}
</script>
</head>
<body>
<form action="http://www.google.com" method="post">
<p>
<input name="WebPage1" value="webpage1" type="button" />
</p>
</form>
</body>
</html>
```

webpage2.html

```
<html>
<body>
<form action="http://www.google.com" method="post">
<p>
<input name="WebPage2" value="webpage2" type="button"
onclick="parent.topPage.ChangeContent()"/>
</p>
</form>
</body>
</html>
```

2. Write a script to change the content of a child window.

Ans:

webpage1.html

```
<html>
<head>
<script language="Javascript" type="text/javascript">
function changeContent(){
    parent.topPage.location.href='WebPage3.html';
}
</script>
</head>
<body>
<form action="http://www.google.com" method="post">
<p>
<input name="WebPage1" value="webpage1" type="button" onclick="changeContent()"/>
</p>
</form>
</body>
</html>
```

webpage2.html

```
<html>
<body>

<form action="http://www.google.com" method="post">
<p>
<input name="WebPage2" value="webpage2" type="button"/>
</p>
</form>
</body>
</html>
```

webpage3.html

```
<html>
<body>
<form action="http://www.google.com" method="post">
<p>
<input name="WebPage3" value="webpage3" type="button"/>
</p>
</form>
</body>
</html>
```

3. Writing to a Child Window from a JavaScript

Ans:

4. Explain types of modifiers of regular expression with the help of examples

Ans:

1. The “g” modifier specifies a global match.
   A global match finds all matches.

```
<html>
<body>
<h2>JavaScript Regular Expressions</h2>
<p id="demo"></p>
<script>
    let text = "Is this all there is?";
    let pattern = /is/g;
    let result = text.match(pattern);
    document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

2. The “I” modifier specifies a case-insensitive match.

```
<html>
<body>
<h2>JavaScript Regular Expressions</h2>
<p id="demo"></p>
<script>
    let text="Visit W3Schools";
    let pattern=/w3schools/i;
    let result=text.match(pattern);
    document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

3. The “m” modifier specifies a multiline match.
   - It only affects start ^ and end $
   - ^ specifies a match at the start of the string.
   - $ specifies a match at the end of the string.
   - With the “m” set, ^ and $ also match at the beginning and end of each line.

```
<html>
<body>
<h2>JavaScript Regular Expressions</h2>
<p id="demo"></p>
<script>
let text = `Is this
all there
is`
let pattern = /^is/m;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

5. Examples of regular expression.

Ans:

Example	Description
text.match(pattern)	The string method match()
text.search(pattern)	The string method match()
pattern.exec(text)	The RexExp method exec()
pattern.test(text)	The RegExp method test()

Example:

```
<html>
<body>
<h1>JavaScript Strings</h1>
<p id="demo"></p>
<script>
let text = "The rain in SPAIN stays mainly in the plain";
let result = text.match("ain");
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

6. Create rollover effects which changes colour of its text

Ans:

```
<html>
<body>
<p onmouseover="this.style.color='red'" onmouseout="this.style.color='blue'">
Test Example Text.
</p>
</body>
</html>
```

7. To create rollover effect that involves text and images. When the user places his or her mouse pointer over a book title, the corresponding book image appears.

Ans:

```
<html>
<head>
<title>Rollover Effect</title>
</head>
<body>
<table>
<tbody>
<tr valign="top">
<td width="50">
<a><img src="abc.jpg" name="book" height="100" width="100"></a>
</td><td></td>
<td><a onmouseover="document.book.src='a.png'"><b>Visual Basic 2010 Made Easy</b></a>
<br>
```

```
<a onmouseover="document.book.src='b.png'"><b>Visual Basic 2008 Made Easy</b></a>
<br>
<a onmouseover="document.book.src='c.png'"><b>Visual Basic 6 Made Easy</b></a>
<br>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

Chapter-3:

1. Write a script to create rotating banner ads

Ans:

```
<html>
<head>
<script language="javascript" type="text/javascript">
MyBanners=new Array('banner1.jpg','banner2.jpg','banner3.jpg','banner4.jpg');
banner=0;
function ShowBanners(){
    if (document.images){
        banner++;
if (banner==MyBanners.length) {
    banner=0;
    }
document.ChangeBanner.src=MyBanners[banner];
setTimeout("ShowBanners()",5000);
    }
}
</script>
<body onload="ShowBanners()">
<center>
<img src="banner1.jpg" width="900" height="120" name="ChangeBanner"/>
</center>
</body>
</html>
```

2. Creating Rotating Banner Ads with URL Links.

Ans:

```
<html>
<head>
<script language="Javascript">
MyBanners=new Array('1.jpg','2.jpg','3.jpg');
MyBannerLinks=new Array('http://www.a.net/','http://www.b.com/','http://c.com/');
banner=0;
function ShowLinks(){
    document.location.href="http://www."+ MyBannerLinks[banner];
}
function ShowBanners(){
    if (document.images){
        banner++;
```

```
        if(banner==MyBanners.length){
            banner=0;
        }
document.ChangeBanner.src=MyBanners[banner];
setTimeout("ShowBanners()",5000);
    }
}
</script>
<body onload="ShowBanners()">
<center>
<a href="javascript: ShowLinks()">
<img src="1.jpg" width="900" height="120" name="ChangeBanner"/></a>
</center>
</body>
</html>
```

3. Write a script to creating slide show

Ans:

```
<html>
    <head>
    <script language="Javascript">
    MySlides=new Array('1.jpg','2.jpg','3.jpg');
    Slide=0;
    function ShowSlides(SlideNumber){
        Slide=Slide+SlideNumber;
        if(Slide>MySlides.length-1){
            Slide=0;
        }
        if(Slide<0){
            Slide=MySlides.length-1;
        }
    document.DisplaySlide.src=MySlides[Slide];
    }
    </script>
    </head>
    <body>
    <p align="center"><img src="1.jpg" name="DisplaySlide" width="900" height="120" /><p>
    <center>
    <table border=0>
    <tr>
    <td align=center>
    <input type="button" value="Back" onclick="ShowSlides(-1)">
    <input type="button" value="Forward" onclick="ShowSlides(1)">
    </td>
    </tr>
    </table>
    </center>
    </body>
    </html>
```

4. Enlist and define types of framework (backend and front end)

Ans:

FRONT-END FRAMEWORKS

REACT

React.js is an efficient and flexible JavaScript library for building user interfaces created by Facebook. Technically, React is a JS library, but it is often discussed as a web framework and is compared to any other open source JavaScript framework. React makes it easy to create interactive user interfaces because it has predictable JavaScript code that is easy to debug. Furthermore, it provides a REACT component system where blocks of JavaScript code can be written once and reused repeatedly in different parts of the application or even other applications.

ANGULAR

AngularJS is a popular enterprise-level JavaScript framework used for developing large and complex business applications. It is an open-source web framework created by Google and supported by both Google and Microsoft.

VUE

Vue.js is a progressive framework for building user interfaces. It is an up-and-coming framework that helps developers in integrating with other libraries and existing projects. It has an ecosystem of libraries that allow developers to create complex and solid single-page applications.

BACK-END FRAMEWORKS

EXPRESS

Express.js is a flexible, minimalistic, lightweight, and well-supported framework for Node.js applications. It is likely the most popular framework for server-side Node.js applications. Express provides a wide range of HTTP utilities, as well as high-performance speed. It is great for developing a simple, single-page application that can handle multiple requests at the same time.

NEXT.JS

Next.js is a minimalistic framework that allows a a JavaScript developer to create a server-side rendering and static web applications using React.js. It is one of the newest and hottest frameworks that takes pride in its ease of use. Many of the problems developers experience while building applications using React.js are solved using Next.js. It has many important features included “out of the box,” and makes development a JavaScript breeze.