**QUESTION NO 1:**

**What are URI and URL?**

**ANSWER:**

|  |  |
| --- | --- |
| **URI** | **URL** |
| **1**: URI is acronym for uniform Resource  Identifier. | **1**: URL is an acronym for Uniform Resource Locator. |
| **2**: URI contains two subsets, URN which tell the name and URL, which tells the location. | **2**: URL is the subset of URI, which tells the only location of the resource. |
| **3**: All URIs cannot be URLs, as they can tell either name or location. | **3**: URL is the subset of URI, which tells the only location of the resource. |
| **4: An example** of a URI can be ISBN 0-486-35557-4. | **4: An example** of an URL is https://www.javatpoint.com. |
| **5**: The URI scheme can be protocol, designation, specification, or anything. | **5**: The scheme of URL is usually a protocol such as HTTP, HTTPS, FTP, etc. |

**QUESTION NO 2:**

\

**What are Web Clients and Web Server?**

**Answer:**

|  |  |
| --- | --- |
| **Web Client** | **Web Server** |
| **1**: A web client is an application that communicates with a web server, using Hypertext Transfer Protocol (HTTP). Hypertext Transfer Protocol is the protocol behind the World Wide Web. | **1**: A web server is a piece of software designed to serve web pages/web sites/web services. |
| **2**:  HTTP is invoked. HTTP is behind every request for a web document or graphic, every click of a hypertext link, and every submission of a form. The Web is about distributing information over the Internet, and HTTP is the protocol used to do so. | **2**: **Examples** are IIS, Apache and many more. The name can also refer to the hardware running this software. |

**QUESTION NO #3:**

What is the difference between HTTP Protocol and Messages?

ANSWER:

|  |  |
| --- | --- |
| HTTP PROTOCOL | HTTPS |
| 1: **HTTP stands for Hypertext Transfer Protocol, and it is a protocol—or a prescribed order and syntax for presenting information—used for transferring data over a network.** | **1: HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses.** |
| **2: Most information that is sent over the Internet, including website content and API calls, uses the HTTP protocol.** | **2: HTTPS is far more secure than HTTP. A website that uses HTTP has HTTP:// in its URL, while a website that uses HTTPS has HTTPS://.** |

**QUESTION NO #4:**

What is HTTP Method?

**ANSWER:**

There are the common HTTP Methods.

**1**: GET **4**: PATCH

**2**: POST **5**: DELETE

**3**: PUT **6**: HEAD

**7**: OPTION

**These methods allow for a much richer communication between the client and the server.**

**1: GET METHOD#**

1: The GET method requests data from a server using URL parameters.

2: It is by far the most commonly used HTTP method on the web.

3: The GET request parameters are formatted as name-value pairs.

This example instructs Google to GET the results for recipes using the query parameter q.  
In response, Google will return a page with search results.

**FOR EXAMPLE: www.google.com/search?q=recipes**

## 2: POST Method #

The POST method sends data to a server to change a resource (often a database record).

POST data is sent in the request body which cannot be seen by the user.

**FOR EXAMPLE:**

**POST /customer/edits HTTP/1.1**

**Host: company.com**

**Id=1&firstname=Steve&lastname=Allison**

## 3: PUT Method:[#](https://www.dofactory.com/html/http-methods#put)

1: Just like POST, the PUT method also sends data to a server for modification.

2: However, POST is used to *add* new data to a database, whereas PUT is used to *update* existing data.

3: Unlike POST, PUT always produces the same results, irrespective how often is being called.

## PATCH Method:

## [#](https://www.dofactory.com/html/http-methods#patch)1: PATCH is a more recently added HTTP method -- in 2010.

2: The difference between PATCH and PUT is subtle and can be confusing.

3: Generally, PUT is used to update entire records and all relevant data is included in the request body.

4: PATCH also updates data, but it only sends the data that need change -- much like a 'delta'.

## DELETE Method:[#](https://www.dofactory.com/html/http-methods#delete)

* The DELETE method is used to delete a specified resource - usually a database record.

## HEAD Method:[#](https://www.dofactory.com/html/http-methods#head)

The HEAD sends a request for data, but it does not receive any data from the source.

This method is used to validate a resource before calling it with a GET request.

## OPTIONS Method:[#](https://www.dofactory.com/html/http-methods#options)

The OPTIONS method defines the communication options for the target resource.

**Question No #5:**

**What is an HTML web form?**

**Answer:**

A Web Forms is a part of an HTML document containing HTML form elements such as **input**, **select**, **checkboxes**, **radio**, **buttons**, etc. Users typically complete a form by entering texts, selecting options, and modifying form elements before submitting the form for processing.

**What is the Need for web Form On any Website?**

Some web forms are commonly used, such as contact forms, user subscription forms, registration forms, etc. However, web forms may vary for every business, as their needs are different.

**Common types of web forms are as follows:**

* Contact form
* Registration form
* Sign-in Form
* Lead generation form
* Order form
* Survey form
* Search form
* Email form

**QUESTION NO#5:**

**What are Query strings?**

**ANSWER:**

A query string is a set of characters input to a computer or Web browser and sent to a query program to recover specific information from a database On the Internet, a query string (also called an HTTP query string) is part of the set of characters automatically input in the address bar of a dynamic Web site when a user makes a request for information according to certain criteria.

In a URL (Uniform Resource Locator), the query string follows a separating character, usually a question mark (?). Identifying data appears after this separating symbol. For example, consider the following URL:

http://www.bookfinder4u.com/search\_author/Ernest\_Hemingway.html?sort=date

**QUESTION NO #6:**

**WHAT is HTML Web Storage API / Cookies?**

**ANSWER:**

With web storage, web applications can store data locally within the user's browser.

Before HTML5, application data had to be stored in cookies, included in every server request. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

HTML web storage provides two objects for storing data on the client:

* window.localStorage - stores data with no expiration date
* window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)

**QUESTION NO# 7:**

**What is Session?**

**Answer:**

* Whenever a document is loaded in a particular tab in the browser, a unique page session gets created and assigned to that particular tab. That page session is valid only for that particular tab.
* A page session lasts as long as the tab or the browser is open, and survives over page reloads and restores.
* Opening a page in a new tab or window creates a new session with the value of the top-level browsing context, which differs from how session cookies work.
* Opening multiple tabs/windows with the same URL creates session Storage for each tab/window.
* Duplicating a tab copies the tab's session Storage into the new tab.
* Closing a tab/window ends the session and clears objects in session Storage.

Data stored in session Storage is specific to the protocol of the page. In particular, data stored by a script on a site accessed with HTTP is put in a different session Storage object from the same site accessed with HTTPS

**QUESTION NO#8:**

**JavaScript – events, synchronous/asynchronous requests, form validation?**

**ANSWER:**

**EVENTS:**

JavaScript has events that provide a dynamic interface to a webpage. These events are connected to elements in the (DOM).

Also, these events by default use the bubbling propagation i.e., upwards in the DOM from children to parent. We can bind events either as inline or in an external script. With the help of JavaScript, you can detect when certain events happen, and cause things to occur in response to those events.

## ****Types of Events in JavaScript****

There are different types of events in JavaScript that are used to react to events. Here, we will discuss some of the famous or most commonly used events such as:

* **On click**
* **One-up**
* **Onmouseover**
* **On load**
* **On focus**

**ASYNCHRONOUS:**

Asynchronous is a non-blocking architecture, so the execution of one task isn't dependent on another. Tasks can run simultaneously.

**SYNCHRONOUS:**

Synchronous is a blocking architecture, so the execution of each operation is dependent on the completion of the one before it.

**QUESTION NO#9:**

**WHAT IS OBJECT-RELATIONAL-MAPPING (ORM)?**

**ANSWER:**

An object-relational mapped provides an object-oriented layer between relational databases and object-oriented programming languages without having to write SQL queries. It standardizes interfaces reducing boilerplate and speeding development time.

Object-oriented programming includes many states and codes in a format that is complex to understand and interpret. ORMs translate this data and create a structured map  to help developers understand the underlying database structure. The mapping explains how objects are related to different tables. ORMs use this information to convert data between tables and generate the SQL code for a relational database to insert, update, create and delete data in response to changes the application makes to the data object. Once written, the ORM mapping will manage the application’s data needs and you will not need to write any more low-level code.

## Types of ORMs

ORMs employ two different strategies: active record pattern and data-mapped pattern.

### 1-Active record pattern

### 2-Data-mapped patterns

**QUESTION NO#10:**

**What is REST Services?**

**ANSWER:**

REST stands for **Representational State Transfer**. REST is web standards based architecture and uses HTTP Protocol. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods.

## [#](https://www.dofactory.com/html/http-methods#post)