React Day-1

Q) **What is CORS (Cross origin resource sharing)**.

```
Cross-Origin Resource Sharing (CORS) is a security feature implemented by
web browsers that restricts web pages from making requests to a different
domain than the one that served the web page. In some cases, a web page may
need to make a "preflight" request to a server before making the actual
request for the resource. This is known as a "preflight" or "OPTIONS" call.

A preflight request is an HTTP request that is made by the browser to the
server to check whether the actual request is allowed to be made.
It is typically an HTTP OPTIONS request, and it includes a number of HTTP
headers that the server can use to determine whether the actual request
should be allowed. The preflight request is made automatically by the
browser, and it is typically made before any other HTTP request is made to
the server.

The purpose of a preflight request is to allow the browser to check whether
the server is willing to accept the actual request before sending the
request and potentially wasting resources. It is a way for the browser to
check whether the server is willing to accept the request before making the
actual request, and it helps to prevent unnecessary network traffic and
improve performance.
```

```
The "Access-Control-Allow-Origin" (ACAO) HTTP header is a response header
that is used to indicate which domains are allowed
to make requests to a server. When a web page makes a request to a server,
the browser sends an HTTP header called the "Origin" header,
 specifies the domain of the web page making the request. The server can
then check the value of the Origin header against the value of the ACAO
header.
If the Origin header matches the value of the ACAO header, the server will
allow the request to be made. If the Origin header does not match the value
of the
ACAO header, the server will block the request.
```

```
The "Access-Control-Allow-Methods" (ACAM) HTTP header is a response header
that is used in conjunction with the Cross-Origin
Resource Sharing (CORS) protocol to specify which HTTP methods are allowed
to be used when making a request to a server.
The ACAM header is used by servers to indicate which HTTP methods are
```

```
allowed when making cross-origin requests to the server.

When a web page makes a request to a server, the browser sends an HTTP
header called the "Origin" header, which specifies
the domain of the web page making the request. The server can then check
the value of the Origin header against the values of the
 Access-Control-Allow-Origin (ACAO) and Access-Control-Allow-Methods (ACAM)
headers to determine whether the request should be allowed.
 If the Origin header matches the value of the ACAO header and the request
method matches one of the methods listed in the ACAM header,
 the server will allow the request to be made. If the Origin header or the
request method does not match the values specified in the ACAO and ACAM
headers,
 the server will block the request.
```

```
The ACAM header can have a number of different values, including:

"*": This value indicates that any HTTP method is allowed to be used when
making a request to the server.
A list of specific HTTP methods: This value indicates that only the
specified HTTP methods are allowed to be used when making a request to the
server.

CORS and the ACAO header work together to provide a secure way for web
pages to make requests to servers on different domains.
The ACAO header is an important part of the CORS protocol, and it is used
by servers to help protect against cross-site scripting (XSS) attacks and
other types of web-based attacks. It is a widely-used standard, and it is
supported by all modern web browsers.
```

Q) What is cross-origin in script tag

```
- The "crossorigin" attribute is an HTML attribute that can be used on the
"script" tag to indicate how a script file should be treated when it is
loaded from a different origin (i.e., a different domain) than the one that
served the web page. The attribute can have two possible values:

"anonymous": This value indicates that the script file should be loaded
from the other origin, but the browser should not send any credentials
(such as cookies or HTTP authentication headers) when making the request
```

```
for the script. This is the default value if the "crossorigin" attribute is
not specified.
"use-credentials": This value indicates that the script file should be
loaded from the other origin, and the browser should send credentials (such
as cookies or HTTP authentication headers) when making the request for the
script.
The "crossorigin" attribute is used in conjunction with the Cross-Origin
Resource Sharing (CORS) protocol, which is a security feature implemented
by web browsers that restricts web pages from making requests to a
different domain than the one that served the web page. CORS works by
adding HTTP headers to server responses that specify which domains are
allowed to make requests to the server, and the "crossorigin" attribute is
used to specify how the browser should handle credentials when making
requests to the server.

The "crossorigin" attribute is commonly used when loading third-party
scripts, such as those from a content delivery network (CDN), to ensure
that the script is loaded correctly and that the correct credentials are
sent when making the request for the script. It is a widely-used standard,
and it is supported by all modern web browsers.
```

Q) **What is difference between React and React DOM**

```
React and ReactDOM are two separate libraries that are both developed and
maintained by Facebook. They are commonly used together to build user
interfaces for web applications, but they have different purposes and
functions.

React is a JavaScript library for building user interfaces. It provides a
declarative, efficient, and flexible way to build complex and scalable
applications by allowing developers to define UI components as reusable,
self-contained pieces of code. React uses a virtual DOM (a lightweight
in-memory representation of the actual DOM) to optimize updates to the user
interface, which helps to improve performance and reduce the amount of work
that the browser has to do.

ReactDOM is a separate library that provides functions for rendering React
components to the DOM (the Document Object Model, which is the structure of
an HTML or XML document as represented in a web browser). It provides a
bridge between React and the DOM, and it is responsible for updating the
DOM to reflect the changes made to a React component.
```

In general, React is used to define and manage UI components, while ReactDOM is used to render those components to the DOM. Together, these libraries provide a powerful and flexible way to build user interfaces for web applications.

Q) Async vs defer

The "async" and "defer" attributes are HTML attributes that can be used on "script" tags to specify how a script file should be loaded and executed by the browser. These attributes are used to help optimize the loading and execution of script files, and they can be used to improve the performance of a web page.

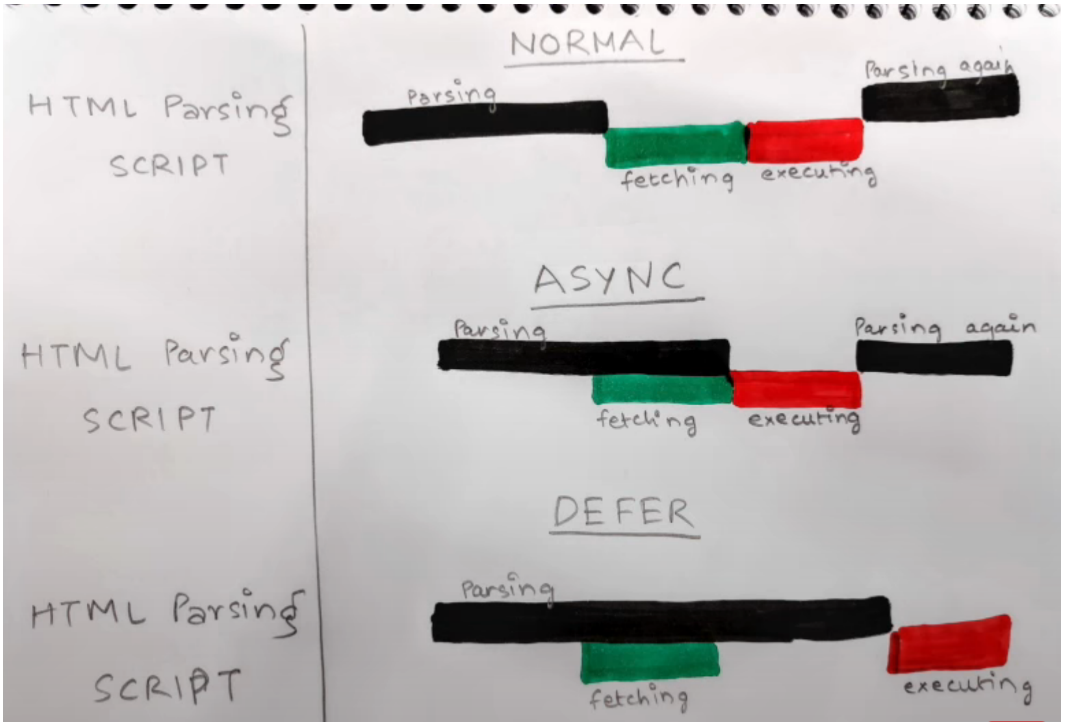
The "async" attribute indicates that the script file should be loaded and executed asynchronously, which means that the browser should not block the rendering of the page while the script is being loaded and executed. This can improve the performance of the page, as it allows the browser to continue rendering the page while the script is being loaded in the background. However, it also means that the script may not be executed in the order in which it appears in the HTML, and it may not be executed until after the page has finished loading.

The "defer" attribute indicates that the script file should be loaded and executed after the page has finished loading. This can also improve the performance of the page, as it allows the browser to render the page and load other resources before the script is loaded and executed. The "defer" attribute also ensures that the script is executed in the order in which it appears in the HTML.

In general, the "async" attribute is best used for scripts that are independent of the page and do not need to be executed in a specific order, while the "defer" attribute is best used for scripts that depend on the page or need to be executed in a specific order. Both attributes can help to improve the performance of a web page by allowing the browser to load and execute scripts in an optimal way.

Note: if there are multiple script tags which are dependent on each other than the async tag does not guarantees the order of execution of the script

Defer attribute maintains the order of execution of the script.
It is best to use defer tag.



Q) What is difference between react.development.js and react.production.js files via CDN?

```
The "react.development.js" and "react.production.js" files are versions of
the React JavaScript library that are intended for use in different
environments. These files are available via a content delivery network
(CDN), which is a network of servers that are used to deliver content (such
as JavaScript libraries) to web browsers.

The "react.development.js" file is the development version of the React
library, and it is intended for use during the development of a web
application. It includes features such as error checking and warnings that
are designed to help developers identify and fix issues in their code.
However, these features can also add additional overhead and make the
library larger in size, which can impact the performance of the
application.

The "react.production.js" file is the production version of the React
library, and it is intended for use in a live, production environment. It
has been optimized for performance, and it includes fewer error checks and
warnings than the development version. As a result, it is typically smaller
in size and may run faster than the development version.
```

```
In general, it is recommended to use the "react.development.js" file during
the development of a web application, and to switch to the
"react.production.js" file when deploying the application to a live,
production environment. This allows developers to take advantage of the
additional error checking and debugging features during development, while
still getting the performance benefits of the production version in a live
environment.
```