

Q.15) How to handle events in HTML.

Ans - An event is an action i.e. occurrence recognized by the software. It can be triggered by user or the system.

Mostly events are used on buttons, hyperlinks, hovers, page loading etc. All this stuff gets into action with help of even handled.

Basically, to handle events in HTML, you just need to add function in HTML tag which is going to be executed in javascript when any event in HTML is fired or triggered.

There are many events attributes in HTML like Keyboard event, mouse event, form event etc.

Syntax: Handle events in HTML.

```
<element onclick = "myScript">
```

\* Various HTML events attributes:-

i) form events :-

i) onblur :- This event occurs when object loses focus.

```
<element onblur = "myScript">
```

2) onchange :- This event occurs when the value of an element has been changed.

<element onchange = "myscript">

3) onfocus :- This event occurs when get focus.

<element onfocus = "myscript">

2) Events :-

1) onclick :- This event occurs when user clicks on an element.

<element onclick = "myscript">

2) onmouseover :- This event occurs when user hove mouse over an element.

E.g:- <body>

<button onclick = "btn ()> clickme </button>

<p id = "demo" > </p>

<p onmouseover = "bover ()>

Hello there. Hello world </p>

<script>

function btn () {

document.getElementById ("demo").

innerHTML = "Nareesh K.";

}

function bover () {

alert ("mouse move over");

}

</script>

<body>

Q.16) what are some of the advantages of HTML5 over previous versions.

Ans-

Basically HTML5 has its many new syntactical features, which includes the `<video>`, `<audio>`, `<canvas>` elements as well as integration of SVG content. due to this elements it will very easy to integrate multimedia to the webpages.

There are new elements like `<section>`, `<article>`, `<header>`, `<nav>` which enrich the semantic value of the document.

- 1) **Mutuality :-** due to usability purpose ; the websites made by developers are highly interactive nowadays & for this developers need to include fluid animations, stream video, play music & social network sites like FB, Instagram, into the sites.
- 2) **Cleaner markup / Improved code :-** HTML5 will enables web designer to use cleaner, neater code , we can remove most `div` tags & replace them with semantic HTML5 elements.
- 3) **Improved semantics :-** Now it is easy to see which part of page are headers, nav, footer, aside etc as the tags are specific for these all and most importantly know what their meaning & purpose is in whole format.

- 4) Elegant forms :- HTML 5 enables designer to use fancier forms. even if it makes form validation native to HTML, user interface enhancement, & reduce need for Javascript : there will be different type of text inputs, searches & different fields for different purpose.
- 5) Consistency :- As website will adopt the new HTML5 element we will see more consistency in terms of HTML used to code a webpage on one site & site compared to another. this will make it much easier for designer & developers to immediately understand how site page is created.
- 6) supports rich media elements :- HTML5 has an inbuilt capability to play audio & video & so we can bid goodbye to this plugin tags.
- 7) offline Application Cache :- HTML5 offers an offline application cache facility which will load the page the user has visited even if the user is temporarily offline.

## Q.17) Difference betn. SVG & Canvas in HTML5

Ans.

The HTML `<svg>` element is a container for SVG graphics. SVG stands for Scalable vector graphics. SVG is useful for defining graphics such as boxes, circles, text, etc.

The HTML `<canvas>` element is used to draw graphics via JavaScript. The `<canvas>` element is a container for graphics.

SVG element.	Canvas element
1) SVG is Vector based (composed of shapes).	Canvas is Raster based (composed of pixel).
2) SVG has better Scalability. So it can be painted with high quality at any resolution.	Canvas has poor Scalability. Hence it is not suitable for painting on higher resolution.
3) SVG gives better performance with smaller number of objects or larger surface.	Canvas gives better performance with smaller surface or large number of only objects.
4) SVG can be modified through script & CSS.	Canvas can be modified through script only.
5) multiple graphical elements which become the part of the page's DOM tree.	single element similar to <code>img</code> in behaviour. (Canvas diagram can be saved to PNG or JPEG format.)

E.g for <svg> :- <!DOCTYPE html>  
<html>  
<head>  
<style>  
#sugelem {  
position: relative;  
left: 50%;  
transform: translateX(-50%);  
}  
</style>  
<div> HTML5 SVG element </div>  
<head>  
<body>  
<h2 align="center"> HTML5 SVG<circle  
<div id="sugelem" height="200"  
xml�="https://www.google.com/2020/  
Svg">  
<circle id="bluecircle" cx="60"  
cy="60" r="50" fill="blue" />  
</div>  
</body>  
</html>

E.g for canvas:

```
<!DOCTYPE Html>
```

```
<html>
```

```
<head>
```

```
<title> HTML 5 Canvas </title>
```

```
</head>
```

```
<body>
```

```
<canvas id = "newCanvas" width = "200"
```

```
height = "100" style = "border: 1px solid
```

```
#fff;"></canvas>
```

```
<script>
```

```
var c = document.getElementById
```

```
('newCanvas');
```

```
var ctx = c.getContext('2d');
```

```
ctx.fillStyle = '#7ccce2b';
```

```
ctx.fillRect(0, 0, 300, 100);
```

```
</script>
```

```
</body>
```

```
</html>
```

Q.18) what type of audio files can be played using HTML 5.

Ans:-

Since the release of HTML5, audio can be added to webpages using the "audio" tag. Previously, audio could be only played on web pages using web plugins like Flash.

The "audio" tag is an inline element that is used to embed sound files into a webpage. It is a useful tag if you want to add audio such as songs, interviews etc.

Syntax :-

<audio>

<source src = "Simple.mp3" type = "audio/mpeg">

</audio>

Attributes :- The previous attribute that can be used with the "audio" tag are listed below.

1) Controls :- designates what controls to display with audio player.

2) Autoplay :- Designates that the audio file will play immediately after it loads controls.

3) Loop :- Designates that the audio file should continuously repeat.

- 5) `src` :- Designates the URL of the audio file.
- 6) `muted` : designates that the audio file should be muted.

\* Supported formats of audio :- These formats `mp3`, `ogg`, `wav` are supported by HTML5.

E.g:- (Adding audio to webpage):- The `Controls` attribute is used to add audio controls such as play, pause, & volume. the "source" element is used to specify the audio files which the browser may use.

E.g:- `<!DOCTYPE html>`

`<html>`

`<head>`

`<audio>`

`<title> Audio </title>`

`</head>`

`<body>`

`<audio controls>`

`<source src="test.mp3" type="audio/mp3">`

`<source src="test.ogg" type="audio/ogg">`

`</audio>`

`</body>`

`</html>`

Q. 19) Explain HTML5 graphics.

Ans.

Graphics are the representations used to make web-page or applications visually appealing & also for improving user experience & user interface interactions. Some e.g. of graphics are photographs, flowcharts, bar graphs, maps, engineering.

Usually, the following technologies are used in web graphics with HTML5 canvas API, webCGM, CSS, SVG, PNG, JPEG etc.

- 1) SVG:- These are images created by a markup language that are resizable, simple, high-quality standalone images that can be exported & imported as well. They are cross-browser friendly & used both on the client-side & server-side of the application.
- 2) PNG:- They are portable, static & lossless with perceptual indexed-color content. files come with a.png or .PNG extension.
- 3) JPEG or JPEG:- Lossy compressed with an adjustable degree of compression. used mainly with digital photography & can achieve a compression of 10:1.
- 4) CSS:- This is type of language mainly used for designing & HTML5 & SVG elements by using code.

5) Canvas API :- Has no DOM & uses vector based methods to create objects, graphics & shapes. Canvas API applications can be standalone or integrated with HTML or SVG, widely used for games.

Q.20) what are the new tags in media elements in HTML.

Ans.

To use media in your next HTML5 application, you need to know what formats are supported. HTML5 supports AAC, MP3 & Ogg Vorbis for audio & Ogg Theora, WebM & MPEG-4 for video.

\* Using the video tag :- To play a video in an HTML page, just use the <video> tag, as shown here.

Syn & ex :- <video src="video.mp4" controls/>

The src attribute sets the name or names of video to play. & the control's Boolean switch dictates whether the default playback controls displays.

\* Using the audio tag :- Using the audio tag is much like using the video tag: you press one or more audio files to the controls & the first one the browser supports is played.

<audio src="audio.ogg" controls>  
</audio>

Q.21) why do you think the addition of drag & drop functionality in HTML5 is important?

Ans:-

Drag & drop is powerful user interface concept which makes it easy to copy, ~~execute~~ & ~~execute~~ & deletion of items with help of mouse click. This allows the user to click & hold the mouse button down over an element, drag it to another location & release the mouse button to drop the element there.

HTML5 came up with drag and drop API that brings native DOM support to browser, making it much easier to code up.

\* Drag and Drop Events:- These are number of events which are fired during various stages of drag & drop operation. These events are listed below:

1) **dragstart** :- Fired when the user starts dragging of the objects

2) **dragenter** :- Fired when the mouse is first moved over the target element while over drag is occurring.

3) **dragover** :- This event is fired as the mouse is moved over an element when a drag is occurring.

4) dragleave :- This event fires when the mouse leaves an element while a drag is occurring. listeners should remove any highlighting or insertion markers used for drop feedback.

5) drag :- fires every time the mouse is moved while the object is being dragged.

6) drop :- The drop event is fired on the element where the drop was occurred at the end of the drag operation.

7) dragend :- fires when the user releases the mouse button while dragging an object

\* drag and drop process :-

1) Step 1 :- Making an object draggable :-  
First you need to set draggable attribute to true for that element.

Set an event listener for dragstart that stores the dom being dragged.

The even listener dragstart will set the allowed effects (copy, move, link or some combination)

2) Step 2:- Dropping the object :-

The ~~deagenter~~ event, which is used to determine whether or not the drop target is to accept the drop.

The ~~deagover~~ event, which is used to determine what feedback is to be shown to the user.

Finally; the drop event, which allows the actual drop to be performed.

Q. 22) what are raster images & vector images

Ans

1) Raster image :- This image is also known as bitmap, are composed of individual pixels of color. Each color pixel contributes to the overall image.

Raster images might be compared to pointillist paintings, which are composed with a series of individually-colored dots of points. Raster images are capable of rendering complex, multi-colored visuals, including soft color gradients.

There are different types of raster images.

Files : JPG, GIF, and PNG are examples & each file type has its own nuances.

2) Vector image :- Unlike raster graphics which are composed of colored pixels arranged to display an image, vector graphics are made up of paths, each with a mathematical formula that tells the path how it is shaped & what color it is bordered with or filled by.

Since mathematical formulas dictate how the image is rendered, vector images retain their appearance regardless of size. They can be scaled infinitely. Vector images can be created & edited in programs such as Illustrator, CorelDraw & Inkscape.

Raster images	vector images
1) They are composed of pixels.	They are composed of paths.
2) In Raster graphics, refresh process is independent of the complexity of the image.	vector displays flicker when the number of primitives in the image become too large.
3) Graphics primitives are specified in terms of end points & must be scan converted into corresponding pixels.	Scan conversion is not required.
4) Raster graphics can draw mathematical curves, polygons & boundaries of curved primitives only by pixel approximation.	vector graphics draw continuous & smooth lines.
5) Raster graphics cost less.	vector graphics cost more as compared to raster graphics.

Q.23) what are different approaches to make an image responsive.

Ans.

- 1) Device pixel-based method :- This approach allows you to use multiple versions of the same image. It provides multiple versions of the same image with different resolutions & choose the most suitable one to render based on the user's screen resolution. This method is more suitable for devices that don't render high-resolution images.
- 2) Fluid-image method :- By default, images are not fluid. They tend to keep or stay at a fixed size when screen size changes. With the fluid image method, you can insert an image into a responsive layout & give the ability to stretch or shrink as necessary.
- 3) Fluid-image method :- By default, images are not fluid. They tend to keep or stay at a fixed size when screen size changes. With the fluid image method, you can insert an image into a responsive layout & give the ability to stretch or shrink as necessary.

- 4) **Act Direction method** : Act direction is a common issue we face when dealing with different screen sizes. we can address this by altering image content, cropping down the image, or using a different images, or using a different image based on user's screen size.
- 5) **Type-Switching method** :- There are some browsers and devices that don't support modern image types like webp. the type-switching method can be used to switch b/w images types allowing you to serve the best content to the user, based on device & browser compatibility.

Q.24) what is manifest file in HTML5.

Ans:

A manifest file is text file that tells the browsers to cache certain files on webpages so that they can be used even in offline mode. HTML5 cache webpages by specifying manifest attribute in `<html>` tag. all the web pages that contain manifest attributes or are specified in the manifest file will be cached whenever a user visits that page.

It always starts with Cache manifest keyword & contains three sections.

i) Cache :- This section lists all the resources including webpages, CSS style-sheets, Javascripts files, & images that will be cached immediately after their first download. these resources can be used even in offline mode after their first download & don't require a connection to the server.

E.g:- Create a file named Cache.html & add it to be the Cache section of the demo.apache file so as to cache it & use it in offline mode.

```
<!DOCTYPE html>
<html manifest="demo.apache">
<body>
<h1>welcome to my site</h1>
</body>
</html>
```

- 2) Network :- This section lists all the resources that will never be cached. These resources can't be used in offline mode & always requires a connection to server.
- 3) Fallback :- This section lists the fallback resource that will be used in case a page is not accessible. It specifies the primary resource that will be replaced with fallback resource specified next to it in case of server connection failure.

```
<!DOCTYPE html>
<html manifest="demo.appcache">
  <body>
    <h1> welcome </h1>
    <a href="cache.html"> cache file </a>
    <a href="network.html"> network file </a>
    <a href="fallback.html"> offline file </a>
  </body>
</html>
```

Q. 26) Explain web components and its usage.

Ans.

Web components are generally available in all of the major browsers with the exception of Microsoft Edge & Internet Explorer 11, but polyfills exist to fill in those gaps.

Web components consist of three separate technologies that are used together.

1) Custom Element :- Quite simply, these are fully valid HTML elements with custom templates, behaviours & tag names made with a set of JavaScript APIs. Custom elements are defined in HTML living standard specification.

As the name implies, custom elements are HTML element, like `<div>`, `<section>` or `<article>`; but something we can name ourselves are defined via browser API.

E.g:-

HTML :- `<my-component></my-component>`

JS :- class MyComponent extends HTMLElement {  
 connectedCallback () {

this.innerHTML = "`<h1> Hello world</h1>`";

}

}

customElements.define ("my-component",  
 MyComponent);

2) Shadow DOM :- The shadow is an encapsulated version of the DOM. This allows authors to effectively isolate DOM fragments from one another, including anything that could be used as CSS selectors & the style associated with them. When using the light DOM, an element can be selected by using document.querySelector('selector') or by targeting any element's children by using element.querySelector('selector');

E.g:-

HTML :- <div id="e.g"> This will use CSS <div>  
<button id="button"> Not tomato </button>

JS:- const shadowRoot = document.getElementById('e.g').~~attach~~.attachShadow({ mode: 'open' });

shadowRoot.innerHTML = '<style>

button {

background: tomato;

color: white;

}

</style>

<button id="button"> <slot> <slot>

tomato </button>;