Drag & Drop Elements with JavaScript and HTML | JavaScript Tutorial In Hindi #64

Are you trying to add the drag & drop functionality to your website, but you don't know exactly how to do it? Well, worry not, this tutorial explains how to add this functionality to the website. In this tutorial, we will build a drag-and-drop example using HTML and JavaScript to use the event handlers.

Prerequisites: To complete this tutorial, we need basic HTML and JavaScript knowledge.

As we know, drag and drop is a very common feature. It is when we "grab" an object and drag it to a different location. With JavaScript event handlers, we can turn any element into a draggable item or an item that can be dropped into. Drag and drop is a great interface solution. Taking something and drag and drop it is a simple way to do many things, from copying and moving documents as in file managers or dropping items into a cart.

Drag Events:-

A typical drag operation begins when a user selects a draggable element, drags the element to a droppable element, and then releases the dragged element. During drag operations, several event types are fired, and some events might fire many times, such as the drag and dragover events. Here are some drag events

Event	Description
drag	It is fired when a *dragged item* is dragged.
dragend	It is fired when a drag operation ends, such as releasing a mouse button or hitting the Esc key.
dragenter	It is fired when a dragged item enters a valid drop target.
dragexit	It is fired when an element is no longer the drag operation's immediate selection target.
dragleave	It is fired when a dragged item leaves a valid drop target.
dragover	It is fired when a dragged item is being dragged over a valid drop target, every few hundred milliseconds.

Event	Description
dragstart	It is fired when the user starts dragging an item.
drop	It is fired when an item is dropped on a valid drop target.

Code index.html as described/written in the video

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Drag and Drop - Js Tutorial 64</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div class="whiteBox">
        <div class="imgBox" draggable="true"></div>
    </div>
    <div class="whiteBox"></div>
    <div class="whiteBox"></div>
    <div class="whiteBox"></div>

</body>
<script src="index.js"></script>
</html>
```

Code index.js as described/written in the video

```
console.log('This is drag and drop utility');

const imgBox = document.querySelector('.imgBox');
const whiteBoxes = document.getElementsByClassName('whiteBox');

// Event listeners for draggable element imgBox
imgBox.addEventListener('dragstart', (e) => {
```

```
    console.log('DragStart has been triggered');
    e.target.className += ' hold';
    setTimeout(() => {
        e.target.className = 'hide';
    }, 0);

});

imgBox.addEventListener('dragend', (e) => {
    console.log('DragEnd has been triggered');
    e.target.className = 'imgBox';
});

for (whiteBox of whiteBoxes) {
    whiteBox.addEventListener('dragover', (e) => {
        e.preventDefault();
        console.log('DragOver has been triggered');
    });

    whiteBox.addEventListener('dragenter', (e) => {
        console.log('DragEnter has been triggered');
        e.target.className += ' dashed';
    })

    whiteBox.addEventListener('dragleave', (e) => {
        console.log('DragLeave has been triggered');
        e.target.className = 'whiteBox'
    })

    whiteBox.addEventListener('drop', (e) => {
        console.log('Drop has been triggered');
        e.target.append(imgBox);
    })
}
```

Code style.css as described/written in the video

Copy

```
body{
```

```
    background: darkslateblue;
}

.imgBox{
    background-image: url('photo.png');
    position: relative;
    top: 7px;
    left: 5px;
    height: 145px;
    width: 145px;
    cursor: pointer;
}

.whiteBox{
    display: inline-block;
    height: 155px;
    width: 155px;
    margin: 10px;
    background-color: white;
    border: 3px black solid;
}

.hold{
    border: solid red 4px;
}

.dashed{
    background: rgb(214, 206, 206);
    border-style: dashed;
}

.hide{
    display: none;
}
```

Previous Next

CodeWithHarry