

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

# HTML Drag and Drop

Difficulty Level : Medium • Last Updated : 13 Dec, 2021

In this article, we will know the **HTML Drag & Drop**, also understand its implementation with the help of examples.

Drag and Drop is a very interactive and user-friendly concept that makes it easier to move an object to a different location by grabbing it. This allows the user to click and hold the mouse button over an element, drag it to another location, and release the mouse button to drop the element there. In HTML 5 Drag and Drop are much easier to code and any element in it is draggable.

**Drag and Drop Events:** There are various Drag and Drop events, some of them are listed below:

- **ondrag**: It is used to use when the element or text selection is being dragged in HTML.
- **ondragstart**: It is used to call a function, `drag(event)`, that specifies what data to be dragged.
- **ondragenter**: It is used to determine whether or not the drop target is to accept the drop. If the drop is to be accepted, then this event has to be canceled.
- **ondragleave**: It occurs when the mouse leaves an element before a valid drop target while the drag is occurring.
- **ondragover**: It specifies where the dragged data can be dropped.
- **ondrop**: It specifies where the drop has occurred at the end of the drag operation.
- **ondragend**: It occurs when the user has finished dragging an element.

**Procedure for Drag and Drop:**



# Start Your Coding Journey Now!

[Login](#)[Register](#)[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

- **Step 1:** Make an object draggable
  - First set the draggable attribute to true for that element to be draggable `<img draggable = "true">`
  - Then, specify what should happen when the element is dragged. The `ondragstart` attribute calls a function, `drag(event)`, that specifies what data to be dragged. The `dataTransfer.setData()` method sets the data type and the value of the dragged data
  - The event listener `ondragstart` will set the allowed effects (copy, move, link, or some combination).
- **Step 2:** Dropping the Object
  - The `ondragover` event specifies where the dragged data can be dropped. By default, data/elements cannot be dropped in other elements. To allow a drop, it must prevent the default handling of the element. This is done by calling the `event.preventDefault()` method
  - Finally, the drop event, which allows the actual drop to be performed

**Example 1:** This example shows the drag & drop of an image in HTML.

## HTML

```
<!DOCTYPE HTML>
<html>
<head>
  <style>
    #getData {
      width: 250px;
```

# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

```
</style>
<script>
function allowDrop(event) {
    event.preventDefault();
}

function drag(event) {
    event.dataTransfer.setData("text", event.target.id);
}

function drop(event) {
    event.preventDefault();
    var fetchData = event.dataTransfer.getData("text");
    event.target.appendChild(document.getElementById(fetchData));
}
</script>
</head>

<body>
<h3>Drag the GeekforGeeks image into the rectangle:</h3>
<div id="getData"
    ondrop="drop(event)"
    ondragover="allowDrop(event)">
</div>
<br>

</body>
</html>
```

## Output:



# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

*Dragging the image into the box*

**The Data Transfer Object:** The data transfer property is used when the whole process of Drag and Drop happens. It is used to hold the dragged data from the source and drop it to the desired location. These are the properties associated with it:

- **dataTransfer.setData(format, data):** It is used to set the data to be dragged.
- **dataTransfer.clearData(format):** It is used for calling this function with no argument that clears all the data. Calling it with a format argument removes only that specific data.
- **dataTransfer.getData(format):** It returns the data of the specified format. If there is

Mega Job-a-thon DSA Data Structures Algorithms Interview Preparation Data Science To

- **dataTransfer.types:** This property returns an array of all the formats that were set in the dragstart event.
- **dataTransfer.files:** It is used to return all the files that are to be dropped.
- **dataTransfer.setDragImage(element, x, y):** It is used to display an existing image as the drag image instead of the default image alongside the cursor. The coordinates specify the drop location.

# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

type(s) of operations are allowed for the user. The property can be set to the following values: none, copy, copyLink, copyMove, link, linkMove, move, all, and uninitialized.

- **dataTransfer.dropEffect(value):** It is used to control the feedback that the user is given during the dragenter and dragover events. When the user hovers over a target element, the browser's cursor will indicate what type of operation is going to take place (e.g. a copy, a move, etc.). The effect can take on one of the following values: none, copy, link, move.

**Example 2:** This example illustrates the use of the *draggable* property of the element.

## HTML

```
<!DOCTYPE HTML>
<html>

<head>
  <title>Drag and Drop box</title>
  <script>
    function allowDrop(ev) {
      ev.preventDefault();
    }

    function dragStart(ev) {
      ev.dataTransfer.setData("text", ev.target.id);
    }

    function dragDrop(ev) {
      ev.preventDefault();
      var data = ev.dataTransfer.getData("text");
      ev.target.appendChild(document.getElementById(data));
    }
  </script>
  <style>
    #box {
      margin: auto;
      width: 50%;
      height: 200px;
      border: 3px solid green;
      padding: 10px;
    }
  </style>
</head>

<body>
  <div id="box">
    <div id="drag">Drag me</div>
  </div>
</body>
</html>
```

# Start Your Coding Journey Now!

[Read](#)
[Discuss](#)
[Courses](#)
[Practice](#)
[Video](#)

```

float: left;
margin: 5px;
padding: 10px;
}

#box1 {
width: 50px;
height: 50px;
background-color: #F5B5C5;
}

#box2 {
width: 100px;
height: 100px;
background-color: #B5D5F5;
}

#box3 {
width: 150px;
height: 150px;
background-color: #BEA7CC;
}

p {
font-size: 20px;
font-weight: bold;
text-align: center;
}

.gfg {
font-size: 40px;
color: #009900;
font-weight: bold;
text-align: center;
}
</style>
</head>

<body>
<div class="gfg">GeeksforGeeks</div>

<p>Drag and drop of boxes</p>

<div id="box">
<div id="box1" draggable="true"

```

# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

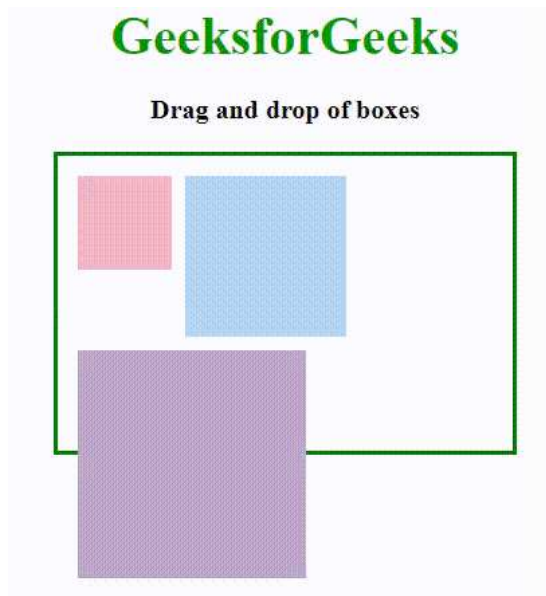
```
</div>
<div id="box3" ondrop="dragDrop(event)"
      ondragover="allowDrop(event)">
</div>
</div>
</body>

</html>
```

## Explanation:

- Start the drag by setting the draggable property of the element to be dragged to true.
- Get the dragged data with the *dataTransfer.getData()* method. This method will return any data that was set to the same type in the *setData()* method.
- Call ***event.preventDefault()*** method to allow the dropping of elements in other elements by preventing the default handling of the element.
- The element is stored in the variable data which we append into the drop element.

## Output:



**Example 3:** This example illustrates the image drag and drop.

TML

# Start Your Coding Journey Now!

[Read](#)
[Discuss](#)
[Courses](#)
[Practice](#)
[Video](#)

```
<script>
function allowDrop(ev) {
    ev.preventDefault();
}

function dragStart(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}

function dragDrop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
</script>
<style>
.div1 {
    width: 240px;
    height: 75px;
    padding: 10px;
    border: 1px solid black;
    background-color: #F5F5F5;
}

p {
    font-size: 20px;
    font-weight: bold;
}

.gfg {
    font-size: 40px;
    color: #009900;
    font-weight: bold;
}
</style>
</head>

<body>
    <div class="gfg">GeeksforGeeks</div>

    <p>Image Drag and Drop in boxes</p>
```

```
<div class="div1"
    ondrop="dragDrop(event)"
    ondragover="allowDrop(event)">
```



# Start Your Coding Journey Now!

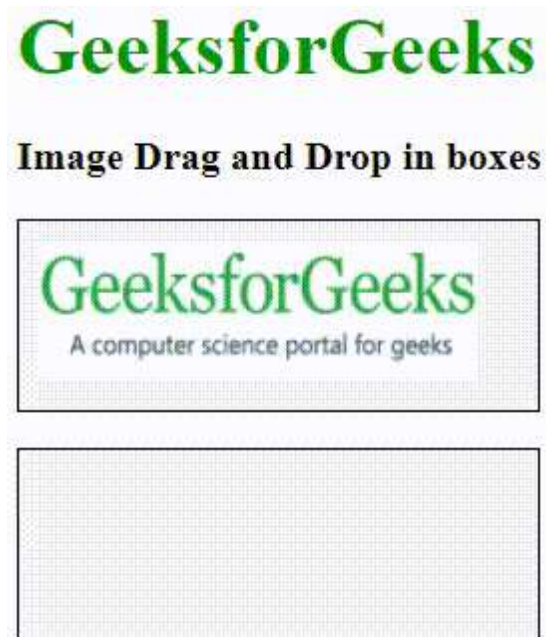
[Read](#)
[Discuss](#)
[Courses](#)
[Practice](#)
[Video](#)

```

<img alt="GeeksforGeeks logo" data-bbox="188 113 479 165" style="width:220px; height:70px;"/>
</div>
<br>
<div class="div1"
      ondrop="dragDrop(event)"
      ondragover="allowDrop(event)">
</div>
</body>
</html>

```

## Output:



*Dragging the image into another box*

**Example 4:** This example demonstrates dragging the text in the rectangular box in HTML.

## HTML

```

<!DOCTYPE HTML>
<html>
  <head>
    <title>Draggable Text</title>
    <style>

```

# Start Your Coding Journey Now!

[Read](#)
[Discuss](#)
[Courses](#)
[Practice](#)
[Video](#)

```

border: 1px solid #272828;
}

h1 {
    color: green;
}
</style>
<script>
function droppoint(event) {
    var data = event.dataTransfer.getData("Text");
    event.target.appendChild(document.getElementById(data));
    event.preventDefault();
}

function allowDropOption(event) {
    event.preventDefault();
}

function dragpoint(event) {
    event.dataTransfer.setData("Text", event.target.id);
}
</script>
</head>

<body>
    <center>
        <h1>GeeksforGeeks</h1>
        <h3>Drag the text in the rectangle</h3>
        <div class="dropbox"
            ondrop="droppoint(event)"
            ondragover="allowDropOption(event)">
        </div>
        <p id="drag1"
            draggable="true"
            ondragstart="dragpoint(event)">
            GeeksforGeeks: A computer science portal for geeks.
        </p>
    </center>
</body>
</html>

```

Output:

# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

GeeksforGeeks: A computer science portal for geeks.



*Dragging the text into the box*

## Supported Browser:

- Google Chrome 93.0
- Microsoft Edge 93.0
- Firefox 92.0
- Opera 78.0
- Safari 14.1



**For Freshers  
& Professionals**

1 Contest | 30+ Opportunities  
[Register Now](#)



Like 8

[Previous](#)[Next](#)

## Related Articles

# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

2. How to Drag and Drop Images using HTML5 ?
3. How to differentiate mouse “click” and “drag” event using JavaScript ?
4. How to Drop fill color to change the image color using HTML and CSS ?
5. How to create a group of related options in a drop-down list using HTML ?
6. How to change font style using drop-down list in JavaScript ?
7. How to define a drop-down list in HTML5 ?
8. Create a Drop Caps Effect using CSS
9. jQuery UI Drop Effect
10. How to set the focus on drop-down list automatically when the page loads in HTML5?

## Article Contributed By :

**prakhar7**

@prakhar7

## Vote for difficulty

Current difficulty : [Medium](#)

Easy

Normal

Medium

Hard

Expert

# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

Article Tags : [HTML-Basics](#), [HTML](#), [Web Technologies](#)

Practice Tags : [HTML](#)

[Improve Article](#)[Report Issue](#)

A-143, 9th Floor, Sovereign Corporate Tower,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

## Company

[About Us](#)[Careers](#)[In Media](#)[Contact Us](#)[Privacy Policy](#)[Copyright Policy](#)[Advertise with us](#)

## News

[Top News](#)[Technology](#)[Work & Career](#)[Business](#)

## Learn

[DSA](#)[Algorithms](#)[Data Structures](#)[SDE Cheat Sheet](#)[Machine learning](#)[CS Subjects](#)[Video Tutorials](#)[Courses](#)

## Languages

[Python](#)[Java](#)[CPP](#)[Golang](#)

# Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

## Web Development

[Web Tutorials](#)[Django Tutorial](#)[HTML](#)[JavaScript](#)[Bootstrap](#)[ReactJS](#)[NodeJS](#)

## Contribute

[Write an Article](#)[Improve an Article](#)[Pick Topics to Write](#)[Write Interview Experience](#)[Internships](#)[Video Internship](#)

@geeksforgeeks , Some rights reserved

