

# Drag and drop images or any HTML element within a document

We saw the main principles of HTML5 drag and drop in the previous sections. There are other interesting uses that differ in the way we copy and paste things to/from the clipboard. The clipboard is accessed through the `dataTransfer` property of the different events:

```
event.dataTransfer.setData("Fruit",event.target.dataset.value);  
...  
var data =event.dataTransfer.getData("Fruit");
```

## <IMG> ELEMENTS ARE ALL DRAGGABLE BY DEFAULT!

Normally, to make an element draggable, you must add the `draggable=true` attribute. `<img>` elements are an exception: they are all draggable by default! The next example shows how to drag and drop images from one location in the document to another one.


### Example: move images as an HTML subtree

- Adapted from [braincracking.org](https://braincracking.org)

Try this example in your browser below or [play with it at CodePen](#):

HTMLCSSJSResultEdit

Drag and drop browser images in a zone:



Cool Web browsers

Code from the example:

```

<html lang="en">
<head>
  <style>
    .box {
      border: silver solid;
      width: 256px;
      height: 128px;
      margin: 10px;
      padding: 5px;
10.     float: left;
    }
  </style>
  <script>
    function drag(target, evt) {
      evt.dataTransfer.setData("Text",target.id);
    }
    function drop(target, evt) {
      var id =evt.dataTransfer.getData("Text");
      target.appendChild(document.getElementById(id));
20.     // prevent default behavior
      evt.preventDefault();
    }
  </script>
</head>
<body>
Drag and drop browser images in a zone:<br/>

  <imgsrc="http://html5demo.braincracking.org/img/logos/chrome1.png" id="cr"

      ondragstart="drag(this, event)"alt="Logo Chrome">

  <imgsrc="http://html5demo.braincracking.org//img/logos/firefox1.png" id="ff"
      ondragstart="drag(this, event)"alt="Logo Firefox">

  <imgsrc="http://html5demo.braincracking.org//img/logos/ie1.png" id="ie"
      ondragstart="drag(this, event)"alt="Logo IE">
33.
  <imgsrc="http://html5demo.braincracking.org//img/logos/opera1.png" id="op"
34.     ondragstart="drag(this, event)"alt="Logo Opera">

  <imgsrc="http://html5demo.braincracking.org//img/logos/safari1.png" id="sf"
      ondragstart="drag(this, event)"alt="Logo Safari"><br/>
  <div class="box" ondragover="return false"ondrop="drop(this, event)">
    <p>Good web browsers</p>
  </div>
  <div class="box" ondragover="return false"ondrop="drop(this, event)">

```

```
        <p>Bad web browsers</p>
    </div>
</body>
45. </html>
```

The trick here is to only work on the DOM directly. We used a variant of the event handler proposed by the DOM API. This time, we used handlers with two parameters (the first parameter, `target`, is the element that triggered the event, and the second parameter is the event itself). In the `dragstart` handler we copy just the `id` of the element in the DOM (*line 15*).

In the `drop` handler, we just move the element from one part of the DOM to another (under the `<div>` defined at *line 38*, that is the drop zone). This occurs at *line 18* (get back the `id` from the clipboard), and *line 19* (make it a child of the div. Consequently, it is no longer a child of the `<body>`, and indeed we "moved" one `<img>` from its previous location to another location in the page).