

Events

Handling events via html attributes

```
<!DOCTYPE html>
<html>
  <head>
    <title>handling events via html attributes</title>
  </head>
  <body>
    <a href="somepage.html" onclick="return myHander()">Click Me</a>
    <script>
      function myHander() {
        alert("You Clicked?");
        return true;
      }
    </script>
  </body>
</html>
```

True goes to nowhere.html.
False aborts the current action.

HTML attribute

Displaying random Images with htmL attribute Event handlers

```
<!DOCTYPE html>
<html>
  <head>
    <title>Displaying random Images with htML attribute Event
handlers </title>
  </head>
  <body>
    
    
    <script>
      var myImages = [
        "coco.jpg",
        "bobo.jpg",
        "dodo.jpg",
        "haha.jpg"
      ];
      function changeImg(that) {
        var newImgNumber = Math.round(Math.random() * 3);
        while (that.src.indexOf(myImages[newImgNumber]) != -1) {
          newImgNumber = Math.round(Math.random() * 3);
        }
        that.src = "img/" + myImages[newImgNumber];
      }</script>
  </body>
</html>
```



Using an event object

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using Event Object</title>
  </head>
  <body>
    <p ondblclick="handle(event)">Paragraph</p>
    <h1 onclick="handle(event)">Heading 1</h1>
    <span onmouseover="handle(event)">Special Text</span>
    <script>
      function handle(e) {
        alert(e.type);
      }
    </script>
  </body>
</html>
```

Displaying random Images with html attribute Event handlers

```
<!DOCTYPE html>
<html>
  <head>
    <title>Displaying random Images with html attribute Event handlers
  </title>
  </head>
  <body>
    
    
    <script>
      var myImages = [
        "coco.jpg",
        "bobo.jpg",
        "dodo.jpg",
        "haha.jpg"
      ];
      function changeImg(e) {
        var e1 = e.target;
        var newImgNumber = Math.round(Math.random() * 3);
        while (e1.src.indexOf(myImages[newImgNumber]) != -1) {
          newImgNumber = Math.round(Math.random() * 3);
        }
        e1.src = "img/" + myImages[newImgNumber];
      }
    </script>
  </body>
</html>
```

Using an event object, you can
get more information about the event.

Handling events using object properties

```
<!DOCTYPE html>
<html>
  <head>
    <title>Displaying random Images with HTML attribute Event handlers </title>
  </head>
  <body>
    
    
    <script>
      var myImages = [
        "coco.jpg",
        "bobo.jpg",
        "dodo.jpg",
        "haha.jpg"
      ];
      function changeImg(e) {
        var e1 = e.target;
        var newImgNumber = Math.round(Math.random() * 3);
        while (e1.src.indexOf(myImages[newImgNumber]) != -1) {
          newImgNumber = Math.round(Math.random() * 3);
        }
        e1.src = "img/" + myImages[newImgNumber];
      }
      document.getElementById("img0").onclick = changeImg;
      document.getElementById("img1").onclick = changeImg;
    </script>
  </body>
</html>
```

Assign ID for getElementById() call

Hook up the event handler to handle the click event

Using Window object's onload event

```
<!DOCTYPE html>
<html>
  <head>
    <title>Displaying random Images with HTML attribute Event handlers </title>
    <script>
      var myImages = [
        "coco.jpg",
        "bobo.jpg",
        "dodo.jpg",
        "haha.jpg"
      ];
      function changeImg(e) {
        var e1 = e.target;
        var newImgNumber = Math.round(Math.random() * 3);
        while (e1.src.indexOf(myImages[newImgNumber]) != -1) {
          newImgNumber = Math.round(Math.random() * 3);
        }
        e1.src = "img/" + myImages[newImgNumber];
      }
      window.onload = function() {
        document.getElementById("img0").onclick = changeImg;
        document.getElementById("img1").onclick = changeImg;
      }
    </script>
  </head>
  <body>
    
    

  </body>
</html>
```

You must execute these statements inside the onload event handler that is called when the page is loaded.

The Standard Event Model

- Define a standard way of adding and removing listeners for an event on the target. Every element node in the DOM is an **EventTarget**.

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Standard Way</title>
    <script>
      function myHandler() {
        alert("You Clicked?");
        return true;
      }
      window.onload = function() {
        var link = document.getElementById("link");
        link.addEventListener("click", function (e)
        {
          alert("This link is going nowhere");
          e.preventDefault();
        });
      };
    </script>
  </head>
  <body>
    <a href="nowhere.html" id="link">Click Me</a>
  </body>
</html>
```

The link is an **EventTarget** object that provides addEventListener() to add a listener.

The **Event** object provides a set of APIs. E.g. the preventDefault() prevents the default action of a link from happening.

Adding and Removing Multiple Event Listeners

```
<!DOCTYPE html>
<html>
  <head>
    <title>Adding and Removing Multiple Event Listeners </title>
    <script>
      var myImages = ["coco.jpg", "bobo.jpg", "dodo.jpg",
      "haha.jpg"];
      function changeImg(e) {
        var el = e.target;
        var newImgNumber = Math.round(Math.random() * 3);
        while (el.src.indexOf(myImages[newImgNumber]) != -1)
        {
          newImgNumber = Math.round(Math.random() * 3);
        }
        el.src = "img/" + myImages[newImgNumber];
      }
      function updateStatus(e) {
        var el = e.target;
        var status = document.getElementById("status");
        status.innerHTML = "The image changed to " + el.src;
        if (el.src.indexOf("haha") != -1) {
          el.removeEventListener("click", changeImg);
          el.removeEventListener("click", updateStatus);
        }
      }
      window.onload = function () {
        var img = document.getElementById("img1");
        img.addEventListener("click", changeImg);
        img.addEventListener("click", updateStatus);
      };
    </script>
  </head>
  <body>
    
    
  </body>
</html>
```

Removing
event listeners

Adding multiple
event listeners

Case Study - Finding NPU (1)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Finding NPU</title>
    <script>
      // Get a random number
      var getRandomNumber = function (size) {
        return Math.floor(Math.random() * size);
      };

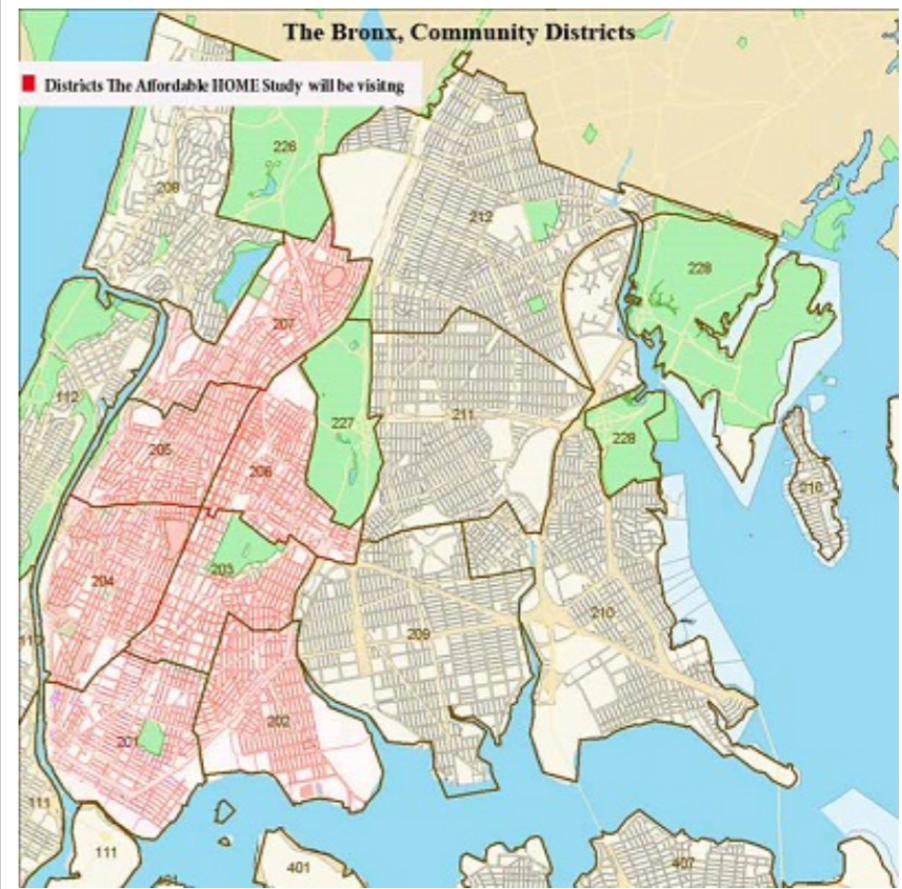
      // Calculate the distance between click event and target
      var getDistance = function (event, target) {
        var diffX = event.offsetX - target.x;
        var diffY = event.offsetY - target.y;
        return Math.sqrt(diffX * diffX + diffY * diffY);
      };

      // Get a string representing the distance
      var getDistanceHint = function (distance) {
        if (distance < 10) {
          return "Less tan 10 miles!";
        } else if (distance < 20) {
          return "Less tan 20 miles!";
        } else if (distance < 40) {
          return "Less tan 40 miles!";
        } else if (distance < 80) {
          return "Less tan 80 miles!";
        } else if (distance < 160) {
          return "Less tan 160 miles!";
        } else if (distance < 320) {
          return "Less tan 320 miles!";
        } else {
          return "Too far away!";
        }
      };

      // Declare the variables
      var width = 400;
      var height = 400;
      var clicks = 0;

      // Create a random target location
      var target = {
        x: getRandomNumber(width),
        y: getRandomNumber(height)
      };
    </script>
  </head>
  <body>
    <div style="text-align: center; margin-top: 20px;">
      <img alt="A map of The Bronx showing community districts. Some districts are shaded red or green, while others are grey. A legend in the top right corner indicates that red areas represent 'Districts The Affordable HOME Study will be visiting'. A callout box in the bottom right corner says 'Less tan 320 miles!'." data-bbox="600 460 940 900"/>
      <p>Less tan 320 miles!</p>
    </div>
  </body>
</html>
```

Finding NPU



Less tan 320 miles!

Case Study - Finding NPU (2)

```
window.onload = function() {
    // Add a click handler to the img element
    var img = document.getElementById("map");
    img.addEventListener("click", function(e) {
        clicks++;

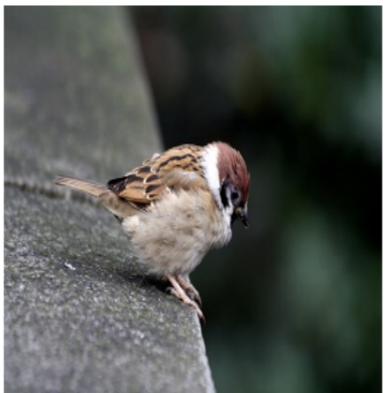
        // Get distance between click event and target
        var distance = getDistance(e, target);
        // Convert distance to a hint
        var distanceHint = getDistanceHint(distance);
        // Update the #distance element with the new hint
        document.getElementById("distance").innerHTML = distanceHint;
        // If the click was close enough, tell them they won
        if (distance < 8) {
            alert("Found NPU in " + clicks + " clicks!");
        }
    ) );
};

</script>
</head>
<body>
    <h1 id="heading">Finding NPU</h1>
    
    <p id="distance"></p>
</body>
</html>
```

Case Study - Raffle Version 1

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Raffle</title>
    <link rel="stylesheet" href="style.css" />
    <script>
      window.onload = function() {
        var images = document.getElementsByClassName("guess");
        for(var i = 0; i < images.length; i++ ) {
          images[i].addEventListener("click", function(e) {
            // Generate a random price between 100 and 500
            var prize = Math.floor((Math.random() * 401) + 100);
            var newNode = document.createElement("P");
            newNode.innerHTML = "<strong>Your prize is " + prize + "</strong>";
            this.appendChild(newNode);
          });
        };
      };
    </script>
    <style>
      div {
        width: 20%;
        height: 200px;
        margin: 5px;
        float: left;
      }
      img {
        width: 100%;
        height: 100%;
      }
    </style>
  </head>
  <body>
    <header>
      <h1>Raffle</h1>
    </header>
    <section>
      <div class="guess"></div>
      <div class="guess"></div>
      <div class="guess"></div>
      <div class="guess"></div>
    </section>
  </body>
</html>
```

Raffle



Your prize is 381

Case Study - Raffle Version 2 (1)

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Raffle</title>
    <link rel="stylesheet" href="style.css" />
    <script>
      window.onload = function () {

        function getRandom(num) {
          var randomNum = Math.floor(Math.random() * num);
          return randomNum;
        }

        var hidePrize = function () {
          var numRand = getRandom(4);
          var images = document.getElementsByClassName("guess");
          for (var i = 0; i < images.length; i++) {
            if (numRand == i) {
              var newNode = document.createElement("SPAN");
              newNode.id = "hasPrize";
              images[i].appendChild(newNode);
              break;
            }
          }
        }

        hidePrize();
      }
    </script>
  </head>
  <body>
    <div class="guess"></div>
    <div class="guess"></div>
    <div class="guess"></div>
    <div class="guess"></div>
  </body>
</html>
```

Case Study - Raffle Version 2 (2)

```
function checkForPrize() {
    var message;
    var node = document.getElementById("hasPrize");
    if (this == node.parentNode) {
        var prize = Math.floor((Math.random() * 401) + 100);
        var newNode = document.createElement("P");
        newNode.innerHTML = "<strong>Your prize is " + prize + "</strong>";
        this.appendChild(newNode);
    } else {
        var newNode = document.createElement("P");
        newNode.innerHTML = "<strong>Sorry, no prize this time.</strong>";
        this.appendChild(newNode);
    }

    // unblind all the click event listeners
    var images = document.getElementsByClassName("guess");
    for (var i = 0; i < images.length; i++) {
        images[i].removeEventListener("click", checkForPrize);
    }
}

var images = document.getElementsByClassName("guess");
for (var i = 0; i < images.length; i++) {
    images[i].addEventListener("click", checkForPrize);
}

};

</script>
<style>
div {
    width: 20%;
    height: 200px;
    margin: 5px;
    float: left;
}
img {
    width: 100%;
    height: 100%;
}
</style>
</head>
<body>
<header>
    <h1>Raffle</h1>
</header>
<section>
    <div class="guess"></div>
    <div class="guess"></div>
    <div class="guess"></div>
    <div class="guess"></div>
</section>
</body>
</html>
```

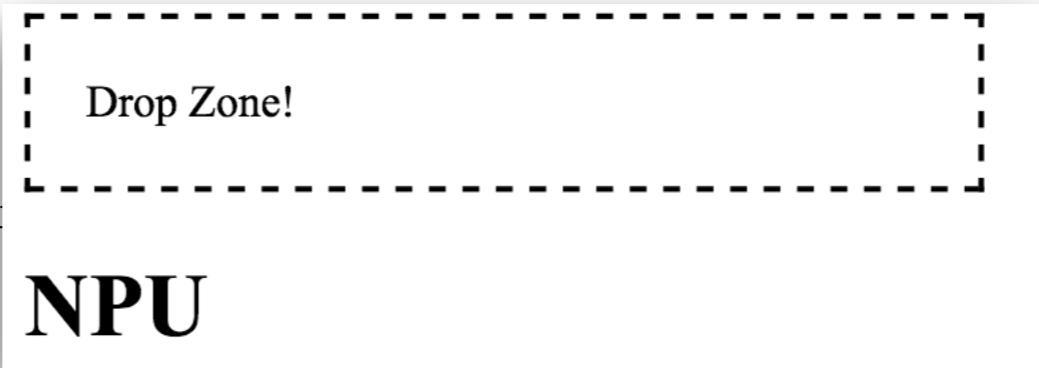
Raffle



Sorry, no prize this time.

Making Content Draggable

```
<!DOCTYPE html>
<html>
  <head>
    <title>Draggable Attr:</title>
    <style>
      .drop-zone {
        width: 300px;
        padding: 20px;
        border: 2px dashed #000;
      }
    </style>
    <script>
      window.onload = function () {
        var dropZone = document.getElementById("dropZone");
        var dropStatus = document.getElementById("dropStatus");
        dropZone.addEventListener("dragenter", function handleDragEnter(e) {
          dropStatus.innerHTML = "You're dragging something!";
        });
      };
    </script>
  </head>
  <body>
    <div id="dropZone" class="drop-zone">Drop Zone!</div>
    <div id="dropStatus"></div>
    <div>
      <h1 draggable="true">NPU</h1>
    </div>
  </body>
</html>
```



Shows the status when the draggable content enters the drop zone.

This is how we can make `<h1>` Draggable using the `draggable` attribute

Transferring data

- The drag and drop specification defines a **DataTransfer** object that is used to hold the data that is being dragged during a drag - and - drop operation.

Example of using DataTransfer object

```
<script>
    window.onload = function () {
        var dropZone = document.getElementById("dropZone");
        var dropStatus = document.getElementById("dropStatus");
        var dragBox = document.querySelector("[draggable]");

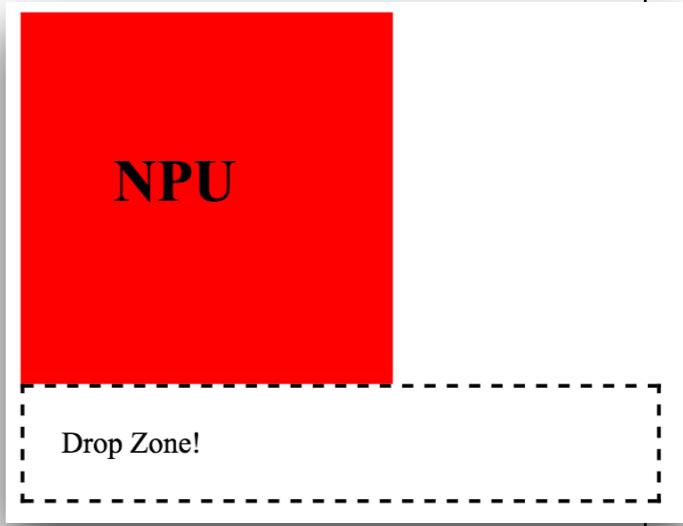
        function dragStartHandler(e) {
            var redBox = document.getElementsByClassName("red")[0];
            e.dataTransfer.setData("text", redBox.innerText);
        }

        function dragDropHandler(e) {
            e.preventDefault();
            if (e.type === "dragover") {
                dropStatus.innerHTML = "You're dragging over the drop zone!";
            } else {
                dropStatus.innerHTML = e.dataTransfer.getData("text");
            }
        }

        dropZone.addEventListener("dragover", dragDropHandler);
        dropZone.addEventListener("drop", dragDropHandler);
        dragBox.addEventListener("dragstart", dragStartHandler);
    },
</script>
</head>
<body>
    <div draggable="true" class="box red"><h1>NPU</h1></div>
    <div id="dropZone" class="drop-zone">Drop Zone!</div>
    <div id="dropStatus"></div>
</body>
</html>
```

Draggable box

NPU



Example of Moving Pictures Around

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Raffle</title>
    <link rel="stylesheet" href="style.css" />
    <script>
      function handleDragStart(e) {
        e.dataTransfer.setData("text", this.id);
      }
      function handleDragEnterLeave(e) {
        if (e.type == "dragenter") {
          this.className = "drag-enter";
        } else {
          this.className = "";
        }
      }
      function handleOverDrop(e) {
        e.preventDefault();
        if (e.type != "drop") {
          return;
        }
        var draggedId = e.dataTransfer.getData("text");
        var draggedEl = document.getElementById(draggedId);
        if (draggedEl.parentNode == this) {
          return;
        }
        draggedEl.parentNode.removeChild(draggedEl);
        this.appendChild(draggedEl);
        this.className = "";
      }
      window.onload = function () {
        var draggable = document.querySelectorAll("[draggable]");
        var targets = document.querySelectorAll("[data-drop-target]");
        for (var i = 0; i < draggable.length; i++) {
          draggable[i].addEventListener("dragstart", handleDragStart);
        }
        for (i = 0; i < targets.length; i++) {
          targets[i].addEventListener("dragover", handleOverDrop);
          targets[i].addEventListener("drop", handleOverDrop);
          targets[i].addEventListener("dragenter", handleDragEnterLeave);
          targets[i].addEventListener("dragleave", handleDragEnterLeave);
        }
      };
    </script>
```

Store the image id

Remove the image from the current drop zone. And then append it to this drop zone

Example of Moving Pictures Around

```
<style>
    img {
        width: 100%;
        height: 100%;
    }
    .panel {
        height: 220px;
        width: 90%;
        margin: 2px;
        background-color: gainsboro;
        float: left;
    }
    [data-drop-target] {
        height: 200px;
        width: 23%;
        margin: 5px;
        background-color: lightblue;
        float: left;
    }
    .drag-enter {
        border: 2px dashed #000;
    }
</style>
</head>
<body>
    <header>
        <h1>Moving Pictures Around</h1>
    </header>
    <section>
        <div class="panel">
            <div data-drop-target="true"></div>
            <div data-drop-target="true"></div>
            <div data-drop-target="true"></div>
            <div data-drop-target="true"></div>
        </div>
        <div class="panel">
            <div data-drop-target="true"></div>
            <div data-drop-target="true"></div>
            <div data-drop-target="true"></div>
            <div data-drop-target="true"></div>
        </div>

    </section>
</body>
</html>
```

Moving Pictures Around

