



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

AULA:

Drag and Drop

Disciplina: Programação de Sistemas para Internet

Alba Lopes, Profa.

<http://docentes.ifrn.edu.br/albalopes>
alba.lopes@ifrn.edu.br

Agenda

- ▶ Ação de Drag and Drop
 - ▶ Introdução
 - ▶ ondragstart
 - ▶ ondragover
 - ▶ ondrop
 - ▶ Exercícios



Drag and Drop

- ▶ É a ação de arrastar um determinado elemento e soltar esse elemento em outro lugar
- ▶ Drag and drop é parte do padrão HTML 5
- ▶ Qualquer elemento HTML pode se tornar arrastável



Propriedade draggable

- ▶ Para que seja possível arrastar um determinado elemento, deve-se atribuir **true** à propriedade **draggable**, dizendo que o elemento agora é “arrastável”.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>DOM</title>
    <script src="meuscript.js"></script>
    <style>
      img{
        width: 200px;
      }
      .retangulo{
        width: 250px;
        height: 250px;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <h3>Arraste o logo do HTML 5 para o retângulo </h3>
    <div class="retangulo">
      
    </div>
    <div class="retangulo">
      </div>
  </body>
</html>
```



Evento ondragstart

- ▶ É preciso também definir que ação deve ser executada quando o elemento começar a ser arrastado.
- ▶ O evento ondragstart é que monitora esse evento.
- ▶ Dessa forma, deve-se criar uma função e definir qual a ação que deve ser feita no momento que o elemento começar a ser arrastado.
- ▶ O objeto **event** deve ser passado por parâmetro para que seja possível enviar e receber dados.

HTML

```
<body>
  <h3>Arraste o logo do HTML 5 para o retângulo </h3>
  <div class="retangulo">
    
  </div>
  <div class="retangulo">
    </div>
</body>
```



Evento ondragstart

- ▶ O objeto **event** possui algumas propriedades úteis ao drag and drop
- ▶ É possível recuperar elemento de onde o evento foi disparado, acessando a propriedade **target**.
- ▶ Nesse caso, o target é o objeto img.

Javascript

```
<head>

  <script>
    function arrastar(ev) {
      alert (ev.target.id) ;
    }
  </script>
</head>
```



Evento ondragstart

- ▶ Outra propriedade do objeto **event** é possibilidade de atribuir valores a serem transferidos de uma origem até o destino (de onde foi iniciado o “arraste” até onde ocorrerá a “soltura”).
- ▶ Isso é feito através da propriedade **dataTransfer** do objeto **event**, onde é possível atribuir o valor que se deseja transferir. O valor a ser transferido é definido através do método **setData**:

Javascript

```
<head>

  <script>
    function arrastar(ev) {
      //alert(ev.target.id);
      ev.dataTransfer.setData("text", ev.target.id);
    }
  </script>
</head>
```



Evento ondragover

- ▶ Assim como por padrão os elementos não podem ser arrastados, também não é permitido soltar elementos em outros elementos.
- ▶ Para tanto, deve-se modificar essa propriedade padrão permitindo que determinado elemento possa ser “depositado” em outro.
- ▶ Isso é feito através do evento ondragover a ser monitorado no elemento destino, através a ação **preventDefault**.

HTML

```
<div class="retangulo" ondragover="permitirSoltar(event)">  
</div>
```

Javascript

```
function permitirSoltar(ev) {  
    ev.preventDefault();  
}
```



Evento ondrop

- ▶ Para monitorar o momento em que o elemento será solto no destino, utiliza-se o evento **ondrop**. Em geral, função invocada no evento ondrop é a que vai de fato executar a ação.

HTML

```
<body>
  <h3>Arraste o logo do HTML 5 para o retângulo </h3>
  <p>
    
  </p>
  <div class="retangulo" ondrop="soltar(event)">
  </div>
</body>
```

Javascript

```
function soltar(ev) {
}

```



Evento ondrop

- ▶ Assim como na função ondragover, também temos que permitir a ação, utilizando o **preventDefault()**
- ▶ Para recuperar o dado que está sendo transferido no evento, utiliza-se o método **getData** da propriedade **dataTransfer**
- ▶ Como a ação desejada é de mover o elemento **img** para o segundo quadrado, utilizamos a ação **appendChild** para mover a imagem como “filha” da **div** de destino.

Javascript

```
function soltar(ev) {  
    ev.preventDefault();  
    var idOrigem = ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(idOrigem));  
}
```



Exercícios

1. Crie uma página contendo dois parágrafos. O primeiro parágrafo deve possuir um texto. Ao arrastar o primeiro parágrafo para o segundo, o conteúdo do segundo parágrafo deve passar a ser o mesmo do primeiro. Mantenha o conteúdo também no primeiro parágrafo.
2. Crie uma página contendo uma lista com várias palavras pertencentes a uma frase. Cada palavra deve ser arrastável. Sua página deve também possuir um parágrafo que irá receber as palavras da lista de forma a montar a frase. Alterne a cor do parágrafo para verde quando as palavras forem arrastadas na sequência correta.



REFERÊNCIAS

- ▶ [1] W3C School. JavaScript Tutorial. Disponível em:
<http://www.w3schools.com/js/>
- ▶ http://www.w3schools.com/html/html5_draganddrop.asp
- ▶ [2] MORISSON, Michael. Java Script - Use a Cabeça. Ed. 2. Rio de Janeiro: Altabooks
- ▶ [3] Manzano, José; Toledo, Suely. Guia de Orientação e Desenvolvimento de Sites - HTML, XHTML, CSS e JavaScript / Jscript. 2a. Edição

