**Drag Event**

É o evento que ocorre quando um elemento está sendo arrastado e solto.

Todos os eventos drag-n-drop pertecem ao objeto Drag Event e herda todos os métodos e propriedades dos objetos:

* **Event**
* **MouseEvent**

**Propriedades Próprias**

**dataTransfer:** um objeto que representa o dado que é arrastado.

**Propriedades do objeto dataTransfer**

**dataTransfer.dropEffect:** determina qual efeito visual será exibido no evento drag-n-drop, como o tipo de cursor que será usado para os diferentes tipos de operações.

Aceita os seguintes valores:

* **None(padrão):** quando poder ser arrastado, mas não pode ser solto
* **Copy:** quando o item puder ser copiado para um nova localização
* **Link:** quando algum tipo de conexão é criada entre a localização atual e a nova
* **Move:** quando o item puder ser movido para uma nova localização

**dataTransfer.allowedEffect:** determina quais efeitos podem ser usados no arrasto. Valores aceitos:

* **None:** o item não pode ser dropado
* **Copy:** o item pode ser copiado
* **move:** o item pode ser movido
* **copyMove:** o item pode ser copiado ou movido
* **link:** um link pode ser estabelecido entre a localização atual e a nova
* **linkMove:** junção de link e move
* **all:** qualquer operação pode ser feita
* **uninitialized:** quando nenhum efeito foi definido, é o mesmo que all

**dataTransfer.items:** é um objeto chamado **DataTransferItemlist** que lista o item sendo arrastado.

**dataTransfer.types:** array que contém uma lista dos tipos de dados definidos no evento dragstart.

* A ordem da lista será a mesma da chamada do método setData()
* Cada item do array é uma string MIME

**Métodos do objeto dataTransfer**

**dataTransfer.setData(tipo, dado):** configura o tipo e o dado a ser arrastado.

Onde:

* **Tipo:** string que especifica o tipo de dado a ser arrastado, geralmente em formato MIME. **Ex.:** text/plain
* **Dado:** string representando o dado a ser arrastado

**Ex.:** dataTransfer.setData(‘text’, idDoElemento)

**Nota1:** os tipos de dados que podem ser arrastados são texto, html, links e arquivos

**Nota2:** os tipos de dados serão adicionados à propriedade *dataTransfer.types*

**Nota3:** dados diferentes e do mesmo tipo serão adicionados ao final da lista; dados iguais e do mesmo tipo serão substituídos

**dataTransfer.setDragImage(image|elemento, xOffset, yOffset):** define uma imagem ou elemento que irá acompanhar o ponteiro do mouse enquanto estiver arrastando o dado.

* **Image|elemento:** um elemento <image> ou <canvas> ou um arquivo de imagem
* **xOffset:** deslocamento horizontal em realação ao ponteiro
* **yOffset:** desclocamento vertical em relação ao ponteiro

**dataTransfer.getData(tipo):** retorna o dado para determinado tipo

**Eventos Drag-n-drop**

**Alguns conceitos:**

* **item:**  o dado que está sendo arrastado
* **drop zone:** local onde o item arrastado será colocado

**Eventos relacionados com o elemento**

**Ondragstart:** quando o item começar a ser arrastado. Características:

* **Target:** o dado que está sendo arrastado
* Ação padrão: inicia a operação de drag-n-drop

**Ondrag:** quando o item estiver sendo arrastado

* **Target:** o dado que está sendo arrastado
* **Ação padrão:** continua a ação de drag-n-drop

**Ondragend:** quando o item for solto em qualquer lugar, mesmo na drop zone

* **Target:** ao dado que foi movido
* **Ação padrão:** varia

**Eventos referentes com a drop zone**

**Ondragover:** quando o item estiver sobre a drop zone.

* **Target:** a drop zona atual
* **Ação padrão:** resetar o atributo dropEffect para ‘none’

**Ondgragenter:** quando o item entrar na drop zone.

* **Target:** a drop zona ou elemento body
* **Ação padrão:** rejeitar a drop zona como potencial válida

**Ondraleave:** quando o item sair da drop zone.

* **Target:** a última drop zone que o dado estava
* **Ação padrão:** nenhuma

**Ondrop:** quando o item for solto na drop zone.

* **Target:** a drop zone atual
* **Ação padrão:** vari

**Criando um Drop-n-Drap**

**Primeiro:** temos que determinar se o item pode ser arrastado, caso não possa:

* Se for um elemento html, basta colocar o atributo **draggable=true** nele
* Imagens, textos e links são arrastáveis por padrão

**Segundo:** precisamos adicionar um listener para o evento **dragstart.**

Neste evento:

* Podemos definir as propriedades **efeito permitido(allowedEffect)**, **dado e tipo arrastado(setData)** e **imagem de feedback(setDragImage)**
* **Apenas o dado arrastado é obrigatório**

**Definindo o tipo e o dado arrastado**

Precisamos definir o tipo e dado a ser arrastado com o método **dataTransfer.setData(tipo, dado).**

**Para textos:**

* **Tipo:** text/plain
* **Dado:** vai ser o próprio texto selecionado

**Nota:** você pode encontrar os tipos ser *text/unicode* ou apenas *text*  em códigos antigos

**Ex.:** *dataTransfer.setData(‘text/plain’, ‘algum texto’)*

**Para links:**

* **Tipo:** tanto text/uri-list e text/plain devem ser usados
* **Dado:** o próprio link

**Ex.:**

*dataTransfer.setData(‘text/uri-list’, ‘www.example.com’)*

*dataTransfer.setData(‘text/plain’, ‘www.example.com’)*

**Caso seja múltiplos links, separamos cada link em ‘text/uri-list’ com CRLF(\r\n)**

**Ex.:**

*dataTransfer.setData(‘text/uri-list’, ‘www.example.com****\r\n****www.novoexemplo.com’)*

**Para html:**

* **Tipo:** text/html ou text/plain
* **Dado:** o conteúdo html, como o innerHTML d o elemento

Ex.:

*dataTransfer.setData(‘text/html, elemento.innerHTML )*

*ou dataTransfer.setData(‘text/plain’, ’Bem-vindo, <strong>João</strong>’)*

**Terceiro:** cancelando os eventos ***dragover*** e ***dragdrop****.*

* A maioria dos elementos HTML não permitem que possamos dropar alguma coisa dentro
* Para que consigamos dropar, precisamos cancelar os eventos ***dragenter*** e ***dragover*** porque suas ações padrões impedem o efeito drop

Podemos cancelar de duas formas:

* Colocando **return false**
* Usando o método **preventDefault()** do objeto Event

**Ex. 1:** *<div ondragover="return false"> <div ondradenterr="return false">*

**Ex. 2:** *<div ondragover="event.preventDefault()"> <div ondragenter="event.preventDefault()">*

**Quarto:** definindo listener para o evento drop.

* Precisamos pegar o dado que foi definido no evento ***dragstart***
* Inserir o dado recuperado na drop zone

**Quinto:** finalizando o efeito drag-n-drop com o evento ***dragend***. Se o atributo dragEffect, no evento dragend:

* For none, quer dizer que a operação foi cancelada
* For qualquero outro valor, a operação foi realizada com sucesso