

EVENTO DRAG AND DROP:

Arrastrar e Soltar é un dos grandes cambios que incorpora HTML5 é a función de arrastrar e soltar (Drag and Drop) que se pode aplicar a calquera clase de elementos como contedores (divs), áreas de texto, parágrafos, títulos, imaxes, etc. N DROP:

Arrastrar e soltar é unha ferramenta moi potente que permite levar a interacción co usuario a outro nivel que aproveita esta "febre" que existe con dispositivos móbiles intelixentes como tabletas ou teléfonos que contan cunha pantalla táctil que dá lugar a unha combinación. moi atractivo.

Con Drag and Drop simplemente levamos un elemento de A a B "arrastrándoo" dun lado a outro (non só se aplica a HTML, todas as tecnoloxías que o empregan -como Android- usan o mesmo principio); aínda que no momento do desenvolvemento quizais non nos resulte tan sinxelo, sobre todo porque conta (polo menos na API HTML5) cun importante número de métodos, eventos e propiedades que nos permiten explotar ao máximo esta tecnoloxía.

Para crear elementos arrastrables xa non é necesario utilizar un plugin para levar a cabo esta tarefa, utilizando HTML ou mellor dito HTML5 con JavaScript nativo poderemos arrastrar e soltar sobre todos os elementos que queiramos.

DRAG:

Os elementos arrastrables producen 3 eventos.

dragstar()

Este evento desenvólvese cando comezamos a arrastrar un elemento; concretamente neste evento debemos especificar o que estamos arrastrando e establecer os valores correspondentes a través do método JavaScript **setData**; este evento **só se chama unha vez** (ao seleccionar o elemento).

Tamén neste evento **podemos definir un estilo personalizado** para o elemento que se está arrastrando.

Drag():

Este evento desenvólvese inmediatamente despois do método dragstar e **prodúcese cando estamos arrastrando o elemento** (temos o **clic pulsado** sobre o elemento e non o soltamos) e continuará arrastrándoo; o número de veces que se invoca este evento depende do navegador.

Este evento é moi útil para coñecer o estado do elemento arrastrable en cada momento; por exemplo, para coñecer a posición exacta do elemento.

Dragend():

Ocorre cando rematamos de arrastrar o elemento e só se executa unha vez; tamén se chama independentemente de que o elemento sexa "soltado" no recipiente ou non.

Os atributos que poñemos aos elementos son: **ondragstar**, **ondrag** e **ondragend** respectivamente.

DROP:

Neste caso prodúcese 4 eventos.

Dragenter ()

Prodúcese cando comezamos a mover un elemento arrastrable dentro do seu recipiente, pero aínda non se soltou; noutras palabras, este evento desenvólvese cando o elemento "arrastrable" entra na "zona de caída".

Este evento é ideal para cambiar as regras de estilo do contedor que está "dentro" do elemento arrastrable.

Neste caso podemos inspeccionar os datos transferidos (dataTransfer) a través do evento, estes datos foron inicializados mediante o método setData; Ademais, podemos inspeccionar o tipo de datos devoltos.

dragleave ()

Prodúcese cando un elemento arrastrable foi soltado ou sacado do contedor.

Este evento é perfecto para **eliminar calquera estilo aplicado ao noso** recipiente.

dragover()

Prodúcese continuamente cando un elemento arrastrable se move dentro do seu contenedor e só deixa de executarse ata que soltamos o elemento arrastrable dentro do contenedor ou saímos do contenedor; este evento é **perfecto para saber cal é a posición do elemento arrastrable dentro do contedor**.

Do mesmo xeito que o evento de arrastre, é perfecto para determinar a posición exacta do elemento arrastrable porque se chama repetidamente mentres o elemento "arrastrable" está dentro do contedor; o número de veces que se invoca este evento depende do navegador.

Drop()

Ocorre cando un elemento arrastrable se solta nun contedor; neste caso debemos recoller a información do elemento arrastrable mediante o método getData.

Os atributos que poñemos aos elementos son: **ondragent, ondragleave, ondragover e ondrop** respectivamente.

DRAG AND DROP. Atributos en HTML5

Só xestionar eventos non é suficiente para comezar a arrastrar e soltar en HTML5. O atributo **draggable="true"** debe engadirse aos elementos DOM que queremos que sexan arrastrables.

Calquera elemento DOM pódese arrastrar.

Ata agora explicouse que acontecementos ocorren no ciclo de vida de Arrastrar e Soltar; é dicir, como interactúan os arrastrables cos seus contedores; pero...

Como enviamos os datos que se manteñen na operación de arrastre? Como se determina que datos (que estamos arrastrando) ten o elemento que se arrastra?

O obxecto `DataTransfer` en Drag and Drop (HTML5)

Simplemente arrastrar e soltar elementos á esquerda e á dereita non é suficiente a menos que **se vexan afectados os datos que movemos**; este obxecto úsase para **personalizar as operacións de arrastrar e soltar**; por exemplo utilízase para establecer a información do evento arrastrable no evento `dragstart()` e esta información procédese no evento `drop()`, imos ver algúns dos seus métodos máis importantes:

`.setData(formato, datos)`

Con este método debemos inserir a información que queremos gardar do elemento arrastrable (arrastrar) e debe ter un tipo definido, aínda que debes ter en conta o soporte do navegador:

Se vas inserir texto: "texto/sen formato".

Se vas inserir un url: "text/uri-list".

Esta información debe establecerse no evento `dragstar` mediante `event.dataTransfer.setData(type, data)`.

Indicamos o tipo de datos (formato) a inserir ("Datos", para calquera tipo de datos) e os datos.

`.getData(formato)`

Ao contrario do método anterior, este devolve os datos recollidos establecidos mediante o método `setData`.

Para recuperar a información, só debe facerse no evento `drop()` usando `event.dataTransfer.getData(type)`.

`.clearData()`

Borra todos os datos definidos polo método `setData(format, data)` mediante `event.dataTransfer.clearData(type)`.