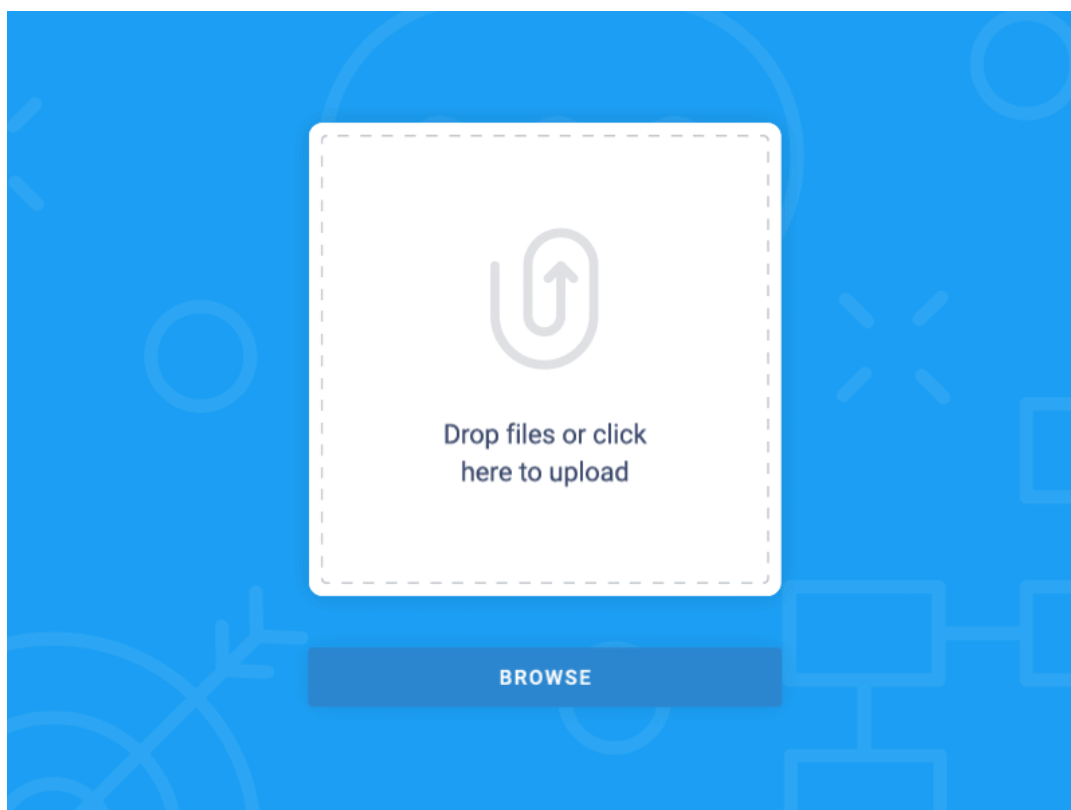


## Какво е Drag and Drop?

Drag and Drop е функция, чрез която потребителите могат да избират обект или текст, да го преместят на желано място и да го пуснат там. Drag and Drop е част от повечето графични потребителски интерфейси, но не се намира във всеки софтуер.

Използването на метода Drag and Drop е предназначено да улесни потребителите да движат и копират елементи. За да се изпълни това действие, потребителят трябва да маркира текст или да избере обект за преместване, след това трябва да натисне и задръжи левия бутон на мишката, за да вземе обекта. След това потребителят премества обекта на желаното място, докато все още държи натиснат бутона на мишката. Когато бутонът на мишката се освободи, той "изпуска" обекта на това място или се движи, или копира, в зависимост от програмата. Пример за това може да видите на Фигура 1, която се намира по-долу.



Фигура 1. Drag and Drop

## Работа с HTML5 Drag and Drop

Традиционният drag-and-drop е бил популярен сред потребителите от дните на оригиналния Apple Macintosh, но днес има компютри и много мобилни устройства с по-сложно drag-and-drop поведение. Drag-and-drop се използва при управление на файлове, прехвърляне на данни, диаграми и много други операции, където движението на обект е по-естествено предвидено с жест, отколкото с клавишна команда. Попитайте разработчиците на улицата какво включва drag-and-drop и вероятно ще получите безброй различни отговори, в зависимост от тяхните любими програми и текущи работни задачи. Попитайте нетехническите потребители за drag-and-drop и те могат да се взират безмълвно пред вас. Сега функцията е толкова вкоренена в изчисленията, че вече често не се нарича по име.

И все пак, HTML не е имал drag-and-drop като основна функция в първите си 15 години от своето съществуване. Въпреки че някои разработчици са използвали вградените способности, за да се справят със събитията, така наречените "events", от ниско ниво на мишката като начин да пробият примитивна версия на тази функция. Тези усилия се пренебрегват в сравнение с типа функции за drag-and-drop, които са били налични в настолните приложения в продължение на десетилетия. С пристигането на добра drag-and-drop функционалност HTML приложенията са напреднали с една стъпка по-близо до достигане на възможностите на техните desktop колеги.

## Историята до момента

Вероятно вече сте виждали примери за drag-and-drop в мрежата и се чудите дали това са употребите на HTML5 с drag-and-drop. Отговорът? Вероятно не.

Причината е, че HTML и DOM са предоставили събития от ниско ниво на мишката още от ранните дни на DOM събитията и това е било достатъчно за творчеството на разработчиците, за да създадат елементарна способност за drag-and-drop. Когато е свързан с позиционирането на CSS, е възможно да се приближи до система за drag-and-drop, чрез създаването на сложни JavaScript библиотеки и солидни познания на DOM събития.

Например, чрез обработване на следните DOM събития е възможно да се преместят елементи в уеб страница, ако кодирате набор от логически стъпки (и някои уговорки):

- **mousedown:** Потребителят започва да работи с мишката. (Дали това е плъзгане или само едно кликване?)
- **mousemove:** Ако мишката все още не е горе, се стартира операция за преместване. (Това е плъзгане или избиране?)
- **mouseover:** Мишката се е преместила над елемент. (Това ли е едно от онези, на които искам да се откажа?)
- **mouseout:** Мишката е оставила елемент, който вече е на място на което няма да може да се пуска друг елемент. (Трябва ли да изготвя обратна връзка?)
- **mouseup:** Мишката се е освободила и е възможно да задейства операцията падане. (Трябва ли падането да е пълно на това място въз основа на това откъде е започнало?)

Въпреки че моделирането на сурова система за drag-and-drop, използвайки събития от ниско ниво е възможно, това има някои съществени недостатъци. Първо, логиката необходима, за да се справите със събитията, свързани с мишките, е по-сложно, отколкото бихте могли да си представите. Тъй като всяко от изброените събития има много крайни случаи, които трябва да бъдат отчетени. По време на тези събития, CSS трябва да се актуализира внимателно, за да предостави обратна информация на потребителя, за възможност за drag-and-drop на всяко определено място.

С ad hoc потребителите могат да извлекат информацията, от която се нуждаят, за да вземат по-добри бизнес решения, без да се налага да включват ИТ отдела. Обикновено един ad hoc отчет е по-визуален от стандартния отчет с данни, което го прави лесен за разбиране и използване от нетехническа аудитория. Но има по-сериозен недостатък, където зависи от тоталния контрол на системата. Ако вие опитате да смесите съдържанието на приложението си с друго съдържание в същата страница, всичко бързо излиза извън контрол и различните разработчици започват да използват събития за собствените си цели. По същия начин, ако се опитате да изтеглите и пуснете съдържание от някой друг код, може да имате проблеми, освен ако не са предварително внимателно координирани и двете. Също така, ad hoc drag-and-drop не взаимодейства с работния плот на потребителя или работата през прозорци.

Новият HTML5 drag-and-drop API е създаден, за да се справи с тези ограничения, вземайки начина, по който е осигурен drag-and-drop в други user interface frameworks.

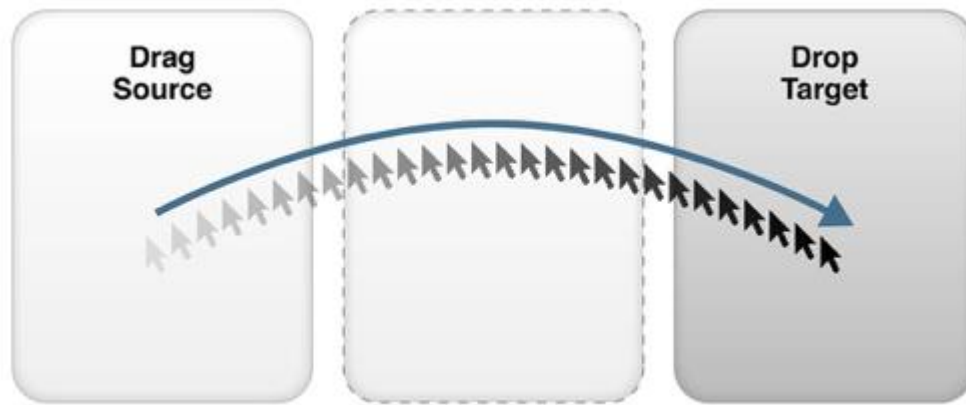
## Преглед на HTML5 Drag and Drop

Ако сте използвали API за drag-and-drop в технологии за програмиране като Java или Microsoft MFC, тогава имате късмет. Новият API за изтегляне и пускане на HTML5 е тясно моделиран от концепциите на тези среди. Първите стъпки са лесни, но овладяването на новата функционалност означава, че ще трябва да се запознаете с нов набор от DOM събития на по-високо ниво на абстракция.

## Голямата картина

Най-лесният начин да научите новия API е да го свържете с понятията, с които вече сте запознати. Ако четете например някаква книга за програмиране за професионалисти в HTML5, ще направим смело предположение, че имате опит с използването на drag-and-drop в ежедневната работа на компютъра. Въпреки това, ето някои стандартни термини върху основните понятия.

Както е показано на Фигура 2, когато (като потребител) стартирате операция с drag-and-drop, започвате с щракване и плъзгане на показалеца. Елементът или регионът, в който сте започнали плъзгането, се наричат **drag source**. Когато пуснете показалеца и завършите операцията, регионът или елементът, към който сте се насочили в края, са известни като **drop target**. Тъй като мишката се движи по страницата, може да преминете през поредица от места за пускане, преди да я пуснете.



Фигура 2. Drag sources и drop targets

Дотук добре. Но просто задържане на мишката и преместването ѝ в друга част от приложението не е това, което представлява drag-and-drop. То е обратната връзка по време на операцията, която прави успешно взаимодействие. Помислете за вашите собствени употреби на drag-and-drop. Тези, които са най-интуитивни са тези, за които системата ви дава постоянни актуализации, за да ви уведоми какво ще се случи, когато в този момент се изпълнява пускането:

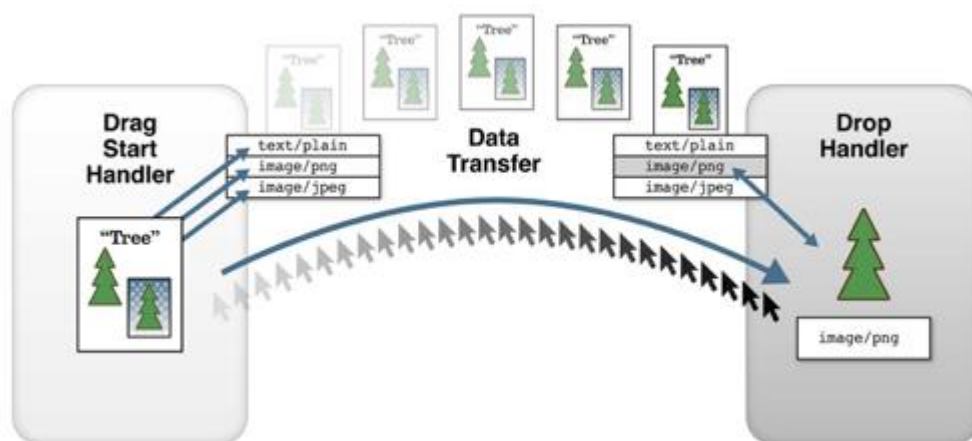
- Показва ли курсорът, че текущата позиция е валидна цел за спускане или означава отхвърляне с „забранен“ индикатор на курсора?
- Курсорът показва ли на потребителя, че операцията ще бъде преместване, връзване или копиране, например с индикатор „плюс“ на курсора?
- Дали областта или целта, върху която премествате, променя външния си вид по никакъв начин, за да покаже, че в момента може да я пуснете там?

С цел да се даде подобна обратна връзка на потребителите в хода на HTML операция drag-and-drop, браузърите излъчват цял куп събития на едно плъзгане. Това се оказва доста удобно, както по време на тези събития ще имаме пълна възможност да променяме DOM и стила на страницата, също така и елементите, които ще дадат точна обратна връзка, която потребителите ще очакват.

Отвъд **drag source** и **drop target** има още една ключова концепция, която да научите в новия API: **data transfer**. Спецификацията описва прехвърлянето на данни като набор от обекти, използвани за излагане на съхранението на данни за плъзгане, което е в основата

на drag-and-drop. Въпреки това, може да е по-лесно просто да мислите за прехвърлянето на данни като за централен контрол на drag-and-drop. Типът на операцията (напр. Преместване, копиране или връзване), изображението, което да се използва като обратната връзка по време на плъзгане и извличането на самите данни са всички управлявани тук на Фигура 3.

По отношение на самите данни, механизмът за прехвърлянето им за завършване на пускането директно адресира една от ограниченията на старите ad hoc drag-and-drop техники описани по-горе. Вместо да принуждава всички drag sources и drop targets да бъдат запознати помежду си, data transfer работи подобно на преговорите по мрежовия протокол. В такъв случай, преговорите се извършват чрез Multipurpose Internet Mail Exchange (MIME) видове.



Фигура 3. Drag and drop договаряне на данни

Представете си например елемент от списък в уеб страница, представляващ човек. Има много различни начини за представяне на данните за един човек. Някои са стандартни, някои не. Когато плъзгането започне в списъка на дадено лице, може да декларира, че данните на човека са в няколко формата, както е показано в Таблица 1.

<b>MIME тип</b>	<b>Резултат</b>
text/plain	Стандартен MIME тип за неформатиран текст. Може да се използва като най-често срещаното представяне, например името на човека.
image/png	Стандартен MIME тип за PNG изображения. Тук той представлява снимката на човека в PNG формат.
image/jpeg	Стандартен MIME тип за JPEG изображения. Може да се използва за прехвърляне на снимката на човека в този формат.
text/x-age	Нестандартен MIME тип (както е посочено от x-префикса). Можем да използваме този формат, за да прехвърлим наш тип информация, като например възрастта на човека.

Таблица 1. Примери за MIME типове в Data Transfer на човек

Когато пускането завърши, drop handler може да поиска списък с налични типове данни. От предоставения списък, drop handler може да избере кой тип е най-подходящ. Drop target с текстов списък може да избере да вземе text/plain „flavor“ на данните, за да извлече името на човека, докато по-напреднал контрол може да избере да извлече и покаже PNG изображението на човека като резултат от пускането. И ако източникът и целта са нестандартни типове, целта може също да извлече възрастта на човека по време на падане(Фигура 3).

Това е процесът на преговори, който позволява drag sources и drop targets да бъдат отделени. Докато drag sources осигуряват данни в избор на MIME типове, drop targets могат да изберат кой формат е подходящ за тях, дори и ако идват от различни разработчици.



## Поток на събитията

Сега, след като проучихме ключовите понятия на drag-and-drop API, нека се фокусираме върху събитията, които могат да бъдат използвани през целия процес. Както ще видите, събитията работят на по-високо ниво от тези на мишките, използвани за модел на системата за drag-and-drop. Въпреки това, drag-and-drop събитията разширяват DOM мишка събитията. Затова все още имате достъп до информацията за мишката на ниско ниво, например координати, ако се нуждаете от нея.

Когато потребител инициира drag-and-drop операция в браузър, поддържащ HTML5, серия от събития се задействат в началото и продължават по време на курса на цялата операция. Тук ще ги разгледаме последователно.

### **dragstart**

Събитието dragstart се изстрелва върху елемент в страницата, когато потребителят започне да го влачи. С други думи, след като мишката е надолу и потребителят се движи при стартиране на мишката. Събитието dragstart е от ключово значение значение, тъй като това е единственото събитие, в което dataTransfer може да има данни, зададени на него, използвайки повикването setData. Това означава, че в dragStart handler, евентуалните типове данни трябва да бъдат настроени така, че да могат да бъдат запитвани при края на пускането, както е описано по-горе.

### **Interception!**

Ако се чудите защо типовете данни могат да се задават само по време на събитието dragStart, всъщност има много добра причина за това. Тъй като drag-and-drop е проектирано да работи през различни прозорци и със съдържание от различни източници, това би било риск за сигурността ако drag event listener може да вмъкне или замени данни, когато влаченето преминава над тях. Представете си, че това е злонамерен раздел от такива listeners, които вмъкват това запитване и заменят drag данните на всяко плъзгане. Това щеше погрешно да представи намеренията на drag source и всички данни след старта ще бъдат забранени.



## **drag**

Drag събитието може да се разглежда като непрекъснато събитие на drag операция. Тъй като потребителят премества курсора на мишката около страницата, drag събитието се извиква многократно върху drag source. Drag събитието ще се задейства няколко пъти всяка секунда по време на операцията. Drag обратна връзка може да бъде променена по време на drag събитие и като dataTransfer данните са изключени.

## **dragenter**

Когато drag преминава в нов елемент на страницата, събитие dragenter се праща на този елемент. Това събитие е подходящо време за задаване на обратна връзка за drop на елемента, според това дали той може да получи drop.

## **dragleave**

Обратното, браузърът ще изпрати dragleave събитие, когато потребителят премести извън елемента, в който dragenter е бил извикан по-рано. Drop обратната връзка може да бъде възстановена в този момент, тъй като мишката вече не е над тази мишена.

## **dragover**

Събитието dragover се извиква на чести интервали, докато мишката се премества над елемент по време на drag операция. За разлика от drag събитието, което е извикано на drag source, това събитие се извиква на текущата мишена.

## **drop**

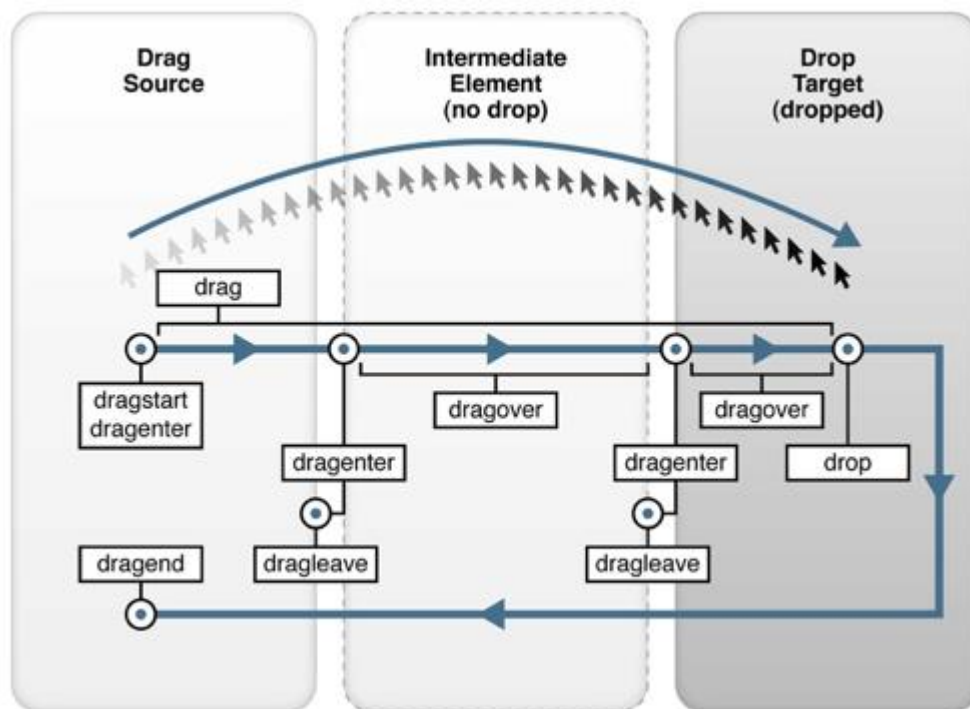
Drop събитието се извиква на текущата мишена, над която потребителят пуска мишката. Въз основа на резултата от dataTransfer обекта, това е мястото, където кодът за обработката на drop трябва да се изпълни.

## **dragend**

Последното събитие във веригата, dragend се изпраща на drag source, показвайки че влаченето е завършило. Това събитие е особено подходящо за почистване на състоянието, използвано по време на

плъзгане, като се извиква, независимо от това дали падането завършва.

Като цяло, има много начини да се правят drag-and-drop операции и да се изпълни нещо. Веригата от drag-and-drop събития е обобщена на Фигура 4.



Фигура 4. Верига от Drag and Drop събития

## Drag участие

Сега, когато видяхте различните събития, които могат да се задействат по време на drag-and-drop операция, може да се чудите какво е необходимо, за да се отбележат елементите в уеб приложението ви като способни на drag. Това е лесно!

Освен няколко елемента - като текстови контроли - елементите в страницата не са способни на drag по подразбиране. За да маркирате конкретен елемент като плъзгащ се, всичко, което трябва да направите е да добавите един атрибут: **draggable**.

```
<div id="myDragSource" draggable="true">
```

Просто като добавите този атрибут, карате браузърът да задейства гореспоменатите събития. След това трябва само да добавите event handlers, които да ги управляват.

## Запознаване с dropzone

Ако смятате, че управлението на всички drag-and-drop събития е сложно, не само вие мислите така. Авторите на спецификацията са проектирали алтернативен, стенографски механизъм за подкрепа на drop събитията: атрибут **dropzone**.

Dropzone предоставя на разработчиците компактен начин да регистрирате това, че елемент е готов да приеме drop, без да пишете дълги event handlers. Атрибутът се състои от няколко разделени с пространството модели, които позволяват на брауъра автоматично да подреди поведението при drop за вас (виж Таблица 2).

Фраза	Резултат
copy, move, link	Допуска се само един от трите вида операции. Ако никой не е посочен, се приема copy.
s:<mime>	Използването на символите s: последвано от MIME тип указва, че данните от този MIME тип могат да бъдат пуснати върху елемента.
f:<mime>	Използването на знаците f: последвано от MIME тип указва, че файловете от този MIME тип могат да бъдат пускани върху елемента.

Таблица 2. Фрази на Dropzone атрибута

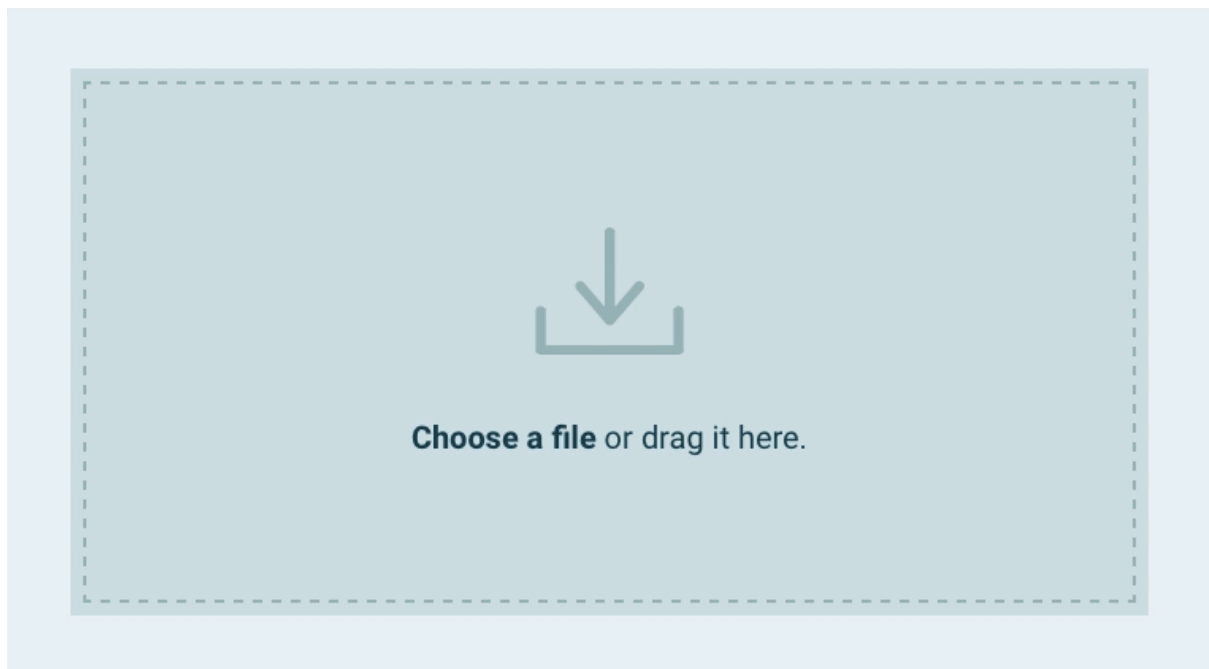
Пример за използване на следния атрибут:

```
<ul id="racers" dropzone="copy s:text/plain s:text/html" ondrop="handleDrop(event)">
```

Това осигурява бърз начин да се каже на брауъра, че copy операциите за елементи, които поддържат или обикновен текст, или HTML формат, са позволени да се пускат.

## Drag and Drop за файлове

Ако някога сте искали по-лесен начин да добавите файлове към уеб приложението си или сте се чудили как някои от най-новите сайтове ви позволяват да плъзгате файлове директно в страницата и да ги качите, отговорът е HTML5 File API. Много браузъри вече поддържат подмножество на стандарта, което позволява файловете да се дърпат в приложения.



Фигура 5. Drag and drop на файл

## Заклучение

Drag-and-drop API може да бъде сложен за овладяване. Включва правилно управление на много събития, някои от които може да се окажат трудни за управление, ако вашето целево оформление е сложно. Въпреки това, ако търсите за drag операции, които пресичат прозорци или браузъри, или дори взаимодействат с десктоп, ще трябва да научите тънкостите на API. По дизайн той комбинира силата на родното приложение drag-and-drop, докато все още работи в рамките на ограниченията за сигурност на среда, в която трябва да има данни, защитени от third-party code.

## Цитирана литература

<http://apress.jensimmons.com/v5/pro-html5-programming/ch9.html>

<https://www.techopedia.com/definition/6918/drag-and-drop>

<https://dribbble.com/shots/3076696-UUPLOAD-Dropzone>

<https://osvaldas.info/drag-drop-file-uploading>