

HTML5 en pratique

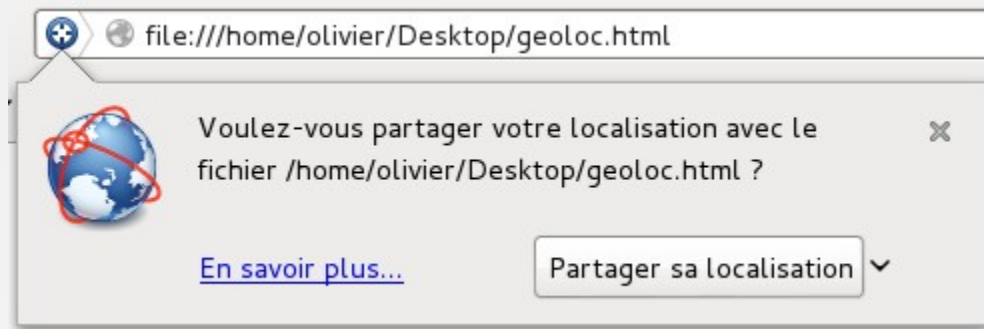
Applications interactives



- Introduction
- Les balises
- CSS 3
- Javascript, le langage du web
- Vers des application plus interactives
- Gestion des données
- Multimédia
- Conclusion

- Géolocalisation
- Drag'n'drop

- La géolocalisation permet de connaître la localisation de l'utilisateur
 - Marketing de proximité
 - Communautés
- Nécessite l'accord de l'utilisateur



- La localisation est déterminée grâce
 - au GPS des appareils mobiles
 - à une triangulation GSM/3G
 - l'adresse IP du point de connexion
 - triangulation wifi
- Elle est accessible via l'objet `navigator.geolocation`
 - Pas toujours disponible !



```
var geoloc = navigator.geolocation;  
if (geoloc) {  
    ...  
}
```

- Deux méthodes pour récupérer la position
 - `getCurrentPosition()`
Détermine la position courante de l'utilisateur
 - `watchPosition()`
Suit le déplacement de l'utilisateur en temps réel
- Les deux méthodes acceptent les mêmes paramètres
 - une fonction *callback* en cas de succès
 - une fonction *callback* en cas d'échec
 - des options, sous forme d'une *map*
 - `timeout` (en ms)
 - `maximumAge` (en ms)
 - `enableHighAccuracy` (`true` | `false`)

- En cas de succès, les fonctions renvoient la même structure `Position` contenant les coordonnées (attribut `coords`) et un `timestamp` :
 - `coords.latitude`
 - `coords.longitude`
 - `coords.accuracy`
 - `coords.altitude`
 - `coords.altitudeAccuracy`
 - `coords.heading` (direction par rapport au nord)
 - `coords.speed` (en mètres / seconde)
 - `timestamp`

```
function onSuccess(position) {  
    var lat = position.coords.latitude;  
    var lng = position.coords.longitude;  
    window.alert(lat + ", " + lng);  
};  
  
function onError(error) {  
    switch(error.code) {  
        case error.TIMEOUT: (...); break;  
        case error.PERMISSION_DENIED: (...); break;  
        case error.POSITION_UNAVAILABLE: (...); break;  
        case error.UNKNOWN_ERROR: (...); break;  
    }  
};  
  
navigator.geolocation  
    .getCurrentPosition(onSuccess, onError, {timeout:500});
```


- Affichage de la position sur une carte statique Google Maps
 - Carte statique, sans contrôles
 - Facile à intégrer

```
function onSuccess(pos) {  
    var lat = pos.coords.latitude;  
    var lng = pos.coords.longitude;  
    var latlng = lat + "," + lng;  
  
    var mapUrl =  
        "http://maps.googleapis.com/maps/api/staticmap"+  
        "?center="+latlng+  
        "&zoom=14&size=400x300&sensor=false"  
  
    var mapHolder = document.getElementById("map");  
    mapHolder.innerHTML = "<img src='"+mapUrl+"' />";  
};
```

- Pour une carte dynamique, il faut utiliser les API Google Maps

```
<script src="http://maps.google.com/maps/api/js?sensor=true"></script>
```

```
navigator.geolocation.getCurrentPosition(function(pos) {  
    var lat = pos.coords.latitude;  
    var lng = pos.coords.longitude;  
    var myPosition = new google.maps.LatLng(lat,lng);  
  
    var opts = {  
        zoom      : 8,  
        center    : myPosition,  
        mapTypeId : google.maps.MapTypeId.ROADMAP  
    };  
  
    var mapHolder = document.getElementById("mapCanvas")  
    var map = new google.maps.Map(mapHolder, opts);  
});
```

- Compatibilité

Geolocation - Candidate Recommendation

Method of informing a website of the user's geographical location

*Usage stats:	Global
Support:	84.47%
Partial support:	0.03%
Total:	84.5%

Show all versions	IE	Firefox	Chrome	Safari	Opera
		3.6			
		12.0			
		14.0			
		15.0	21.0		
		16.0	22.0		
	6.0	17.0	23.0		
	7.0	18.0	24.0		
	8.0	19.0	25.0	5.0	
	9.0	20.0	26.0	5.1	
Current	10.0	21.0	27.0	6.0	12.1
Near future	11.0	22.0	28.0		15.0
Farther future		23.0	29.0		

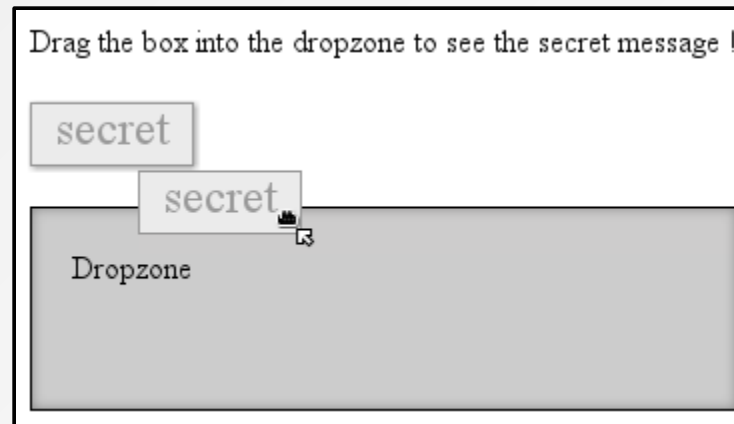
- Le drag'n'drop ("glisser-déposer" selon la terminologie officielle) est un geste naturel
- Deux cas d'utilisation
 - Déplacement d'éléments au sein du navigateur
 - Déplacement d'éléments externes vers le navigateur



- Tout élément peut être déplacé
 - Il suffit de lui ajouter l'attribut "**draggable**"

```
<div draggable="true"> ... </div>
```

- Tout élément peut servir de zone de réception ("*drop*")



- Événements relatifs à l'élément déplacé
 - **dragstart** : début de l'action
 - **drag** : action en cours
 - **dragend** : fin de l'action
- Événements relatifs à l'élément récepteur
 - **dragenter** : début de survol
 - **dragover** : survol en cours
 - **dragleave** : fin de survol
 - **drop** : élément lâché

```
<element>.on<event> = function() { ... }
```

Drag'n'drop

Événements – exemple

- Configuration de l'élément déplacé

```
var item = document.getElementById("item");

// Activer le drag'n'drop
item.setAttribute('draggable', true);

// Changement d'opacité pendant le déplacement
item.ondragstart = function(e) {
    this.style.opacity = '0.4';
};

item.ondragend = function(e) {
    this.style.opacity = '1';
};
```

- Configuration de la zone de réception
 - Par défaut, les éléments ne supportent pas le "drop" ; il faut désactiver ce comportement sur l'événement ondragover

```
var dropzone = document.getElementById("dropzone");

zone.ondragover = function(e) {
    return false;    // ou e.preventDefault();  OBLIGATOIRE !
};

zone.ondrop = function(e) {
    // ...
};
```


- Compatibilité

# Drag and Drop - Working Draft					*Usage stats:	Global
<i>Method of easily dragging and dropping elements on a page, requiring minimal JavaScript.</i>					Support:	60.42%
					Partial support:	20.47%
					Total:	80.89%
Show all versions	IE	Firefox	Chrome	Safari	Opera	
		3.6				
		12.0				
		14.0				
		15.0	21.0			
		16.0	22.0			
	6.0	17.0	23.0			
	7.0	18.0	24.0			
	8.0	19.0	25.0	5.0		
	9.0	20.0	26.0	5.1		
Current	10.0	21.0	27.0	6.0	12.1	
Near future	11.0	22.0	28.0		15.0	
Farther future		23.0	29.0			

Drag'n'drop

Transfert de données intra-navigateur

- Il est possible d'associer des méta-données à l'élément déplacé
- Propriété `dataTransfer` de l'événement de *drag'n'drop*
 - `setData(mime-type, value)`
 - `getData(mime-type)`
 - `clearData(mime-type)`
 - `types` : mime-types des données transmises (tableau)

```
item.ondragstart = function(e) {  
    e.dataTransfer.setData('text/plain', this.innerText);  
};  
  
zone.ondrop = function(e) {  
    e.preventDefault();  
    this.innerHTML += e.dataTransfer.getData('text/plain');  
};
```

- La propriété `dataTransfer.files` contient les descripteurs des fichiers posés dans la zone de réception
 - `lastModifiedDate`
 - `name` : nom court, sans le chemin
 - `size` : taille en octets
 - `type` : type mime

```
zone.ondrop = function(e) {  
    e.preventDefault();  
    var files = e.dataTransfer.files;  
    this.innerHTML = "";  
    for (var f=0; f<files.length; f++) {  
        var file = files[f];  
        this.innerHTML += file.name + "<br/>";  
    }  
};
```

Drag'n'drop

Transfert de fichiers vers le navigateur

- Pour obtenir davantage d'informations sur les fichiers, il faut utiliser l'API **FileReader**
 - `readAsDataURL(file)` : renvoie l'URL complète du fichier
 - `readAsBinaryString(file)` : lit le fichier en binaire
 - `readAsText(file, [charset])` : lit le fichier en UTF8
- La progression est monitorée par des méthodes *callback*
 - `onabort()`, `onerror()`, `onload()`, `onprogress()`
 - Paramètre de type `ProgressEvent` contenant le résultat

```
reader.onload = function(e) {  
    var url = e.target.result;  
    var progress = e.loaded + "/" + e.total;  
}
```

Drag'n'drop

Transfert de fichiers vers le navigateur

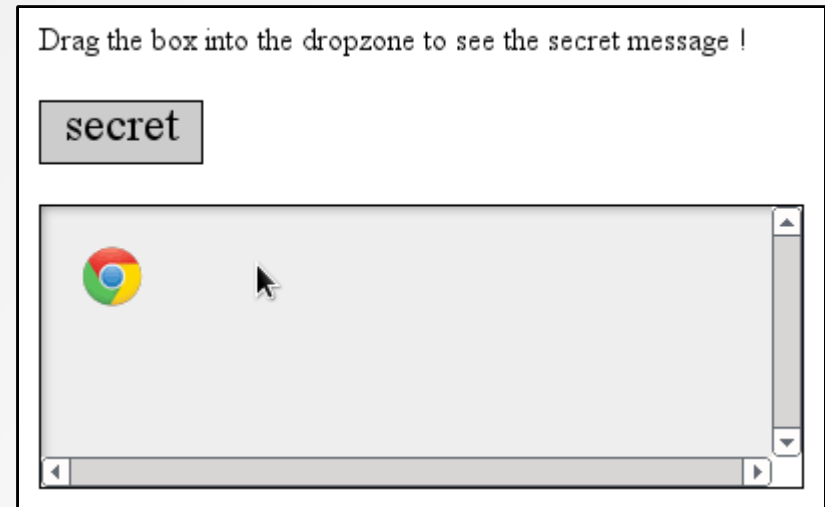
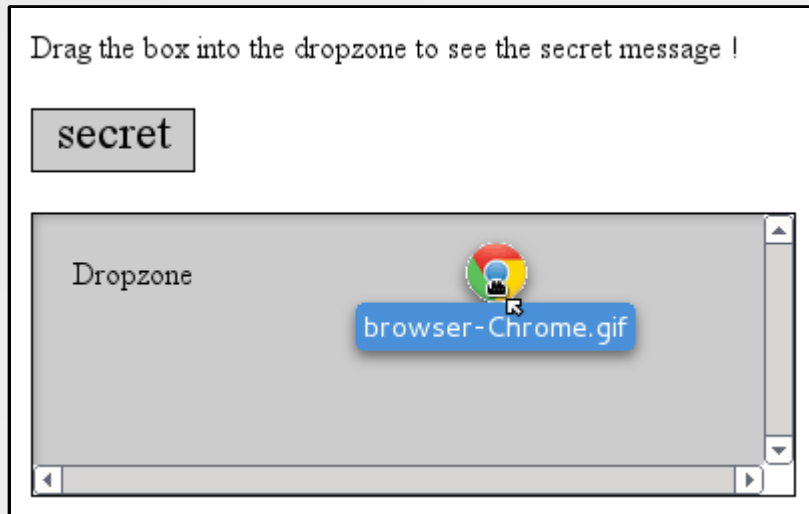
```
zone.ondrop = function(e) {
    e.preventDefault();

    var files = e.dataTransfer.files;
    this.innerHTML = "";
    for (var f=0; f<files.length; f++) {
        var file = files[f];
        var reader = new FileReader();
        reader.onload = function (event) {
            var img = document.createElement("img");
            img.setAttribute('src', event.target.result);
            this.appendChild(img);
        };
        reader.readAsDataURL(file);
    }

    return false;
};
```

Drag'n'drop

Transfert de fichiers vers le navigateur



Drag'n'drop

Transfert de fichiers vers le navigateur

- Compatibilité

# File API - Working Draft					*Usage stats:	Global
<i>Method of manipulating file objects in web applications client-side, as well as programmatically selecting them and accessing their data.</i>					Support:	64.55%
					Partial support:	4.72%
					Total:	69.27%
Show all versions	IE	Firefox	Chrome	Safari	Opera	
		3.6				
		12.0				
		14.0				
		15.0	21.0			
		16.0	22.0			
	6.0	17.0	23.0			
	7.0	18.0	24.0			
	8.0	19.0	25.0	5.0		
	9.0	20.0	26.0	5.1		
Current	10.0	21.0	27.0	6.0	12.1	
Near future	11.0	22.0	28.0		15.0	
Farther future		23.0	29.0			

Drag'n'drop

Transfert de fichiers vers le navigateur

- Compatibilité

# FileReader API - Working Draft					*Usage stats:	Global
Method of reading the contents of a File or Blob object into memory					Support:	67.82%
Show all versions	IE	Firefox	Chrome	Safari	Opera	
		3.6				
		12.0				
		14.0				
		15.0	21.0			
		16.0	22.0			
	6.0	17.0	23.0			
	7.0	18.0	24.0			
	8.0	19.0	25.0	5.0		
	9.0	20.0	26.0	5.1		
Current	10.0	21.0	27.0	6.0	12.1	
Near future	11.0	22.0	28.0		15.0	
Farther future		23.0	29.0			



