



HTML5 et CSS3 pour le multimédia

Remarques générales



Structure du code

Modèle MVC

Modèle

- Données de l'application
- Maintien l'état de l'application
 - e.g. détails utilisateur, état panier

Vue

- HTML: ce qui est affiché à l'écran + contrôles pour interagir avec l'utilisateur
- e.g. récupère l'état d'un panier à partir de la couche Modèle et affichage de son contenu à l'écran

Contrôleur

- Gère la logique et les différents traitements de l'application
- e.g. clic sur un élément de la couche Vue, envoi d'un message à la couche Contrôleur pour ajouter un nouvel élément dans le panier géré par la couche Modèle.

Librairies JavaScript utiles

Modernizr

Modernizr



<https://modernizr.com/download?setclasses>

Transitions CSS (CSS3) + Canvas + HTML5 Audio (HTML5) + Modernizr.load (Extra) + Modernizr.prefixed (Extensibility)

Chargement de scripts

Détection des fonctionnalités disponibles dans le navigateur

jQuery



Détection et réponse à des événements

Envoi des requêtes AJAX au serveur

Accès et manipulation des éléments du DOM

`document.getElementById("game").getAttribute("id") - > $("#game").attr("id")`

Modernizr

- Librairie JavaScript permettant de charger des scripts et de détecter si une fonctionnalité spécifique est prise en charge par un navigateur
- Exemple détection du support du canvas:
 - Sans Modernizr
 - `var element = document.createElement("canvas");`
 - `var canvasSupported = !!element.getContext;`
 - Avec Modernizr
 - `var canvasSupported = Modernizr.canvas;`
- Modernizr contient un ensemble de propriétés dont les valeurs sont instanciées au chargement d'une page à VRAI ou FAUX, selon les fonctionnalités supportées par le navigateur
 - Permet d'avoir des routines de détection plus rapides

Introduction à jQuery

- **Librairie JavaScript open-source et gratuite**
 - ▣ Fournit de nombreuses méthodes pour faciliter la vie des développeurs JavaScript
 - ▣ Peut être considérée comme l'une des 4 technologies à maîtriser par un développeur Web: HTML, CSS, JavaScript et jQuery 😊
- **Les fonctions jQuery sont codées et testées afin d'assurer une compatibilité entre navigateurs**
 - ▣ A partir de sa version 2.0, jQuery n'assure plus la compatibilité avec IE 6, 7 et 8
- **Plusieurs manières de charger jQuery:**
 - ▣ soit à partir d'un fichier local,
 - ▣ soit à partir de Google CDN

Chargement de jQuery à partir de Google CDN (1)

- Utiliser le service CDN (Content Delivery Network) de Google afin de réduire le temps de chargement
 - <https://developers.google.com/speed/libraries/>
- Avantages:
 - Téléchargement de la librairie à partir d'un serveur proche géographiquement
 - Serveur différent du serveur du jeu – téléchargement plus rapide du fichier
 - Augmentation du nombre de fichiers pouvant être téléchargés en simultané
 - Réduction du temps de téléchargement
 - D'autres sites utilisent la même référence – il est possible que l'utilisateur (de par sa navigation Web) ait déjà téléchargé ce fichier sur son ordinateur
- Utiliser Modernizr pour tester l'éventuelle indisponibilité du service et charger une copie locale de jQuery si nécessaire

Chargement de jQuery à partir de Google CDN (2)

- ❑ Téléchargement de jQuery: <http://jquery.com/download/>
- ❑ Code permettant de:
 - ❑ Tester un chargement à partir de Google CDN
 - ❑ Charger la version locale de jQuery si le chargement en ligne échoue

```
<script src="JS/modernizr-custom.js"></script>
<script src="JS/jquery-1.12.1.min.js"></script>
<script>
Modernizr.load({
load : "//ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js",
complete : function() {
if (!window.jQuery) {
Modernizr.load("JS/jquery-1.12.1.min.js");
}
}
});
</script>
```

Si l'objet `window.jQuery` n'existe pas, on charge la version locale de jQuery.

Chargements multiples avec Modernizr.load


- Nous souhaitons utiliser Modernizr pour charger plusieurs scripts
- Pour ajouter le chargement d'un fichier supplémentaire avec Modernizr:
 - ▣ Enveloppez le premier appel à la méthode `Modernizr.load` entre [...]
 - ▣ Ajoutez une nouvelle entrée:

```
Modernizr.load([  
  load : "//ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js",  
  complete : function() {  
    if (!window.jQuery) {  
      Modernizr.load("JS/jquery-1.12.1.min.js");  
    }  
  },  
  {  
    load: "JS/jeu.js"  
  }  
]);
```

- Il est ainsi possible de charger de nouveaux fichiers, en les ajoutant comme éléments du tableau passé en paramètre de `Modernizr.load`

Fonctionnement de Modernizr.load (2)

- La ligne
 - `Modernizr.load("_js/jquery-1.8.2.min.js");`
- Est équivalente à la ligne
 - `Modernizr.load({load : "_js/jquery-1.8.2.min.js"});`



La 1^{ère} ligne utilise un nom de fichier au format *String* pour charger juste ce fichier, sans préciser d'autres options de configuration.

Fonctionnement de Modernizr.load (1)

```
Modernizr.load({  
  load :  
    "//ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js",  
  complete : function() {... Modernizr.load("JS/jquery-  
1.12.1.min.js");}  
});
```

Modernizr est basé sur la librairie yepnope.js pour les chargements de scripts.

- Types de paramètres pris en compte:
 - Un fichier passé en tant que chaîne de caractères,
 - Un objet paires [clé] [valeur]
 - Un tableau contenant soit des string soit des objets
- Il est donc possible de charger plusieurs fichiers avec un seul appel à `Modernizr.load`
 - Premier appel, on passe un objet avec des propriétés "load" et "complete"
 - Deuxième appel, on passe une chaîne de caractères

Sélecteurs dans jQuery (1)

- Les sélecteurs représentent un concept de base dans jQuery
 - Vu que cette librairie est principalement utilisée pour manipuler le DOM
- Permettent la sélection d'un nœud HTML du DOM en utilisant un des éléments suivants:
 - Identifiant de l'élément
 - Classe CSS
 - Type de balise
 - D'autres combinaisons: éléments précédents, position donnée dans une liste, relations parent-enfant
- Possibilité de récupérer des valeurs de propriétés CSS:
 - `var top = $("#game").css("top");`
 - `var width = $("#game").width();`
 - `var divs = $("div");`

Sélecteurs dans jQuery (2)

Comment coder les sélecteurs jQuery ?

- Par type d'élément: `$("h2")`
- Par identifiant d'élément: `$("#monId")`
- Par attribut de classe: `$(".maClasseCSS")`

Comment appeler une méthode jQuery ?

- Récupérer la valeur d'une TextBox:
`var email = $("#email").val () ;`
- Spécifier le texte d'un élément:
`$("#emailErr").text ("Adresse email requise.") ;`
- Spécifier le texte du prochain élément de même type:
`$("#email").next () .text ("Adresse email requise.") ;`

Sélecteurs dans jQuery (3)

Travailler avec els événements

- Tester quand le DOM a fini de charger

```
$(document).ready(function() {  
    alert("Le DOM a fini de charger.");  
});
```

- Gérer le clic sur tous les éléments de type h2

```
$("h2").click(function() {  
    alert("Clic sur h2");  
});
```

- Utiliser la gestion du clic à l'intérieur de la méthode `ready`

```
$(document).ready(function() {  
    $("h2").click(function() {  
        alert("Clic sur h2");  
    }); //fin du gestionnaire de clic  
}); //fin du gestionnaire ready
```

Exemple application

Rollover avec JS et jQuery

Code HTML

```
<script src="jquery-2.1.3.js"></script>
<script src="rollovers.js"></script>
<main>
  <h1>Ram Tap Combined Test</h1>
  <ul id="image_rollovers">
    <li></li>
    <li></li>
    <li></li>
  </ul>
</main>
```

Ram Tap Combined Test

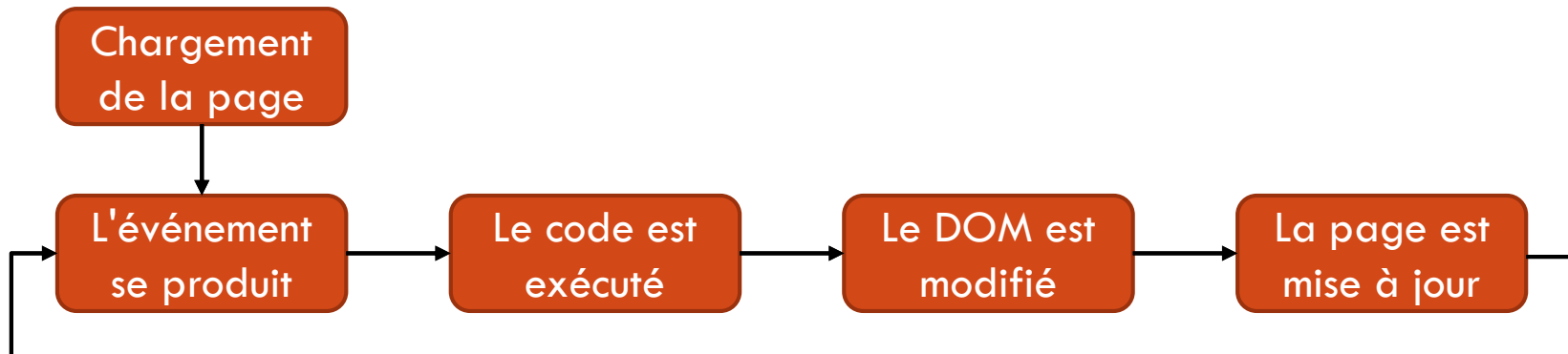


Contenu de rollover.js

```
$(document).ready(function() {
  $("#image_rollovers img").each(function() {
    var oldURL = $(this).attr("src"); // gets the src attribute
    var newURL = $(this).attr("id"); // gets the id attribute
    var rolloverImage = new Image();
    rolloverImage.src = newURL;
    $(this).hover(
      function() {
        $(this).attr("src", newURL); // sets the src attribute
      },
      function() {
        $(this).attr("src", oldURL); // sets the src attribute
      }
    ); // end hover
  }); // end each
}); // end ready
```

Gestion d'événements avec JavaScript

- Un événement représente une action de l'utilisateur:
 - ▣ Clic sur un bouton, chargement d'une page Web
- Un gestionnaire d'événement JavaScript permet de spécifier le code à exécuter lorsque un événement donné se produit
- Cycle de vie des événements au sein du DOM: exemple `onload`



JavaScript modulaire

- Nous allons utiliser une approche modulaire pour notre code JavaScript
 - Minimise le nombre de variables globales
 - Limite les conflits de nommage sur les noms de variables
- Un module permet d'englober les objets et variables d'une application au sein d'un même nom de domaine
 - Le nom de domaine est **NomApplication**
 - Ce module va contenir plusieurs classes **ClasseX**
 - On peut accéder à cette classe à partir de n'importe quel endroit de l'application en écrivant **NomApplication.ClasseX**
- Utiliser un nom de domaine est plus sécuritaire:
 - Si on ajoute dans notre application une autre librairie JavaScript contenant une variable appelée **Jeu**, les 2 peuvent exister simultanément sans générer de conflit

Similaire à l'utilisation d'un nom de domaine dans un langage comme Java ou C#. L'ensemble des classes seront propriétés de l'objet **NomApplication**.

Exemple

Ajout de code dans jeu.js

- `jeu.js` représente notre module jeu

```
var TirBulles window.TirBulles || {};
```

Placer cette ligne en haut de chacun de nos classes, évite à nous soucier de l'ordre de chargement des scripts.

```
TirBulles.Jeu = (function($) {
```

```
    var Jeu = function() {};  
    return Jeu;
```

```
})(jQuery);
```

Test si l'objet TirBulles existe.
Si non, le créer en tant qu'objet vide.

Définition de l'objet TirBulles.Jeu

La fonction est créée, exécutée et détruite instantanément, mais pas avant d'avoir retourné l'objet Jeu.

La fonction entre (...) est à l'intérieur d'un contexte. Dès la fin de son exécution, on peut accéder à la classe Jeu à partir du contexte globale via TirBulles.Jeu

Expressions de fonctions IIFE

- Le fait de placer une fonction entre parenthèses (...) est une approche commune en JavaScript
- Cette structure utilise une expression de fonction immédiatement exécutée
 - IIFE (Immediately Invoked Function Expression)
 - On crée une fonction qui est exécutée tout de suite
 - Le résultat est associé à une variable locale
 - Sans polluer le contexte global
- Les IIFE ont une durée de vie courte et ne sont pas associées en tant que propriétés d'un objet donné
 - Grâce aux fermetures JavaScript, il est possible d'utiliser des variables privées avec la fermeture – une variable existe à l'intérieur du bloc code où elle a été définie et elle persiste en son intérieur
 - La fonction `debutJeu` définie en tant qu'IIFE dans le module `Jeu` peut être utilisée par d'autres fonctions à l'intérieur de `Jeu`, mais ne peut pas être accédée par des fonctions en dehors de ce contexte
 - <https://developer.mozilla.org/fr/docs/Web/JavaScript/Closures>

Fonction jQuery \$

- Traditionnellement les interactions client-serveur sont déclenchées à la suite de l'événement `window.onload`
 - ▣ Signifie que le code HTML a été chargé, le DOM est prêt et toutes les images sont chargées
- Attendre le chargement des images laisse souvent les utilisateurs trop longtemps devant des écrans avec lesquels ils ne peuvent pas interagir
 - ▣ Alternative à préférer: permettre aux utilisateurs d'interagir avec l'application dès que le DOM est prêt
 - ▣ Problème: déterminer quand le DOM est prêt implique souvent du code spécifique à chaque navigateur, qui évolue d'une version à l'autre des navigateurs
- La fonction `$` jQuery uniformise ce fonctionnement à travers les navigateurs
 - ▣ Dans notre cas, l'ensemble des opérations à effectuer dans `init()` doit avoir lieu après le chargement de jQuery et lorsque le DOM est prêt

Théorie

Méthode jQuery fadeOut

- La méthode `fadeOut` de jQuery pour faire disparaître tout élément HTML ayant une classe CSS donnée
 - On passe en paramètre le nombre de millisecondes
- Que fait jQuery dans l'appel à `fadeOut` ?
 - Réduction de l'opacité CSS d'un pas réduit et fixe – répétée dans une boucle durant 300ms
 - A la fin, l'opacité de l'élément considéré devrait être nulle
 - Définition de la propriété CSS `display` de l'élément considéré à `none`
- Méthode `fadeOut` incluse dans jQuery, mais il est possible d'obtenir le même effet en utilisant des sélecteurs et en manipulant les éléments du DOM
 - Plusieurs appels à des fonctions `setTimeout`
 - Plus rapide avec jQuery 😊

Plugins jQuery

- Doc officielle: <http://learn.jquery.com/plugins/basic-plugin-creation/>
- Utile pour ne pas encombrer la logique de l'application avec des éléments d'affichage
 - Créer un nouveau fichier `jquery.nomPlugin.js` dans votre répertoire avec les fichiers `*.js`
 - Cette syntaxe pour le nom permet de reconnaître qu'il s'agit d'un plug-in jQuery
 - Charger ce fichier dans l'application

Illustration plugin kaboom

jquery.kaboom.js

- Structure de base d'un plugin jQuery:

```
1 (function(jQuery) {  
2     jQuery.fn.kaboom = function(parametres)  
3     {  
4  
5     };  
6 })(jQuery);
```

- Cette structure permet les appels de la forme `$ (...).kaboom()`, en passant un paramètre supplémentaire `settings`
- La méthode `kaboom()` sera appelée pour chaque objet (jQuery)
 - N'aura aucune connaissance des objets du jeu, et travaillera uniquement avec des éléments du DOM
- Ce plugin sera appelé à partir de `faireTomberBulles(...)`, remplacez l'appel à la méthode `animate` par un appel au plugin ci-dessus:
 - `bulle.getSprite().kaboom();`

2^e
partie

Animations avec JS/jQuery

Animations

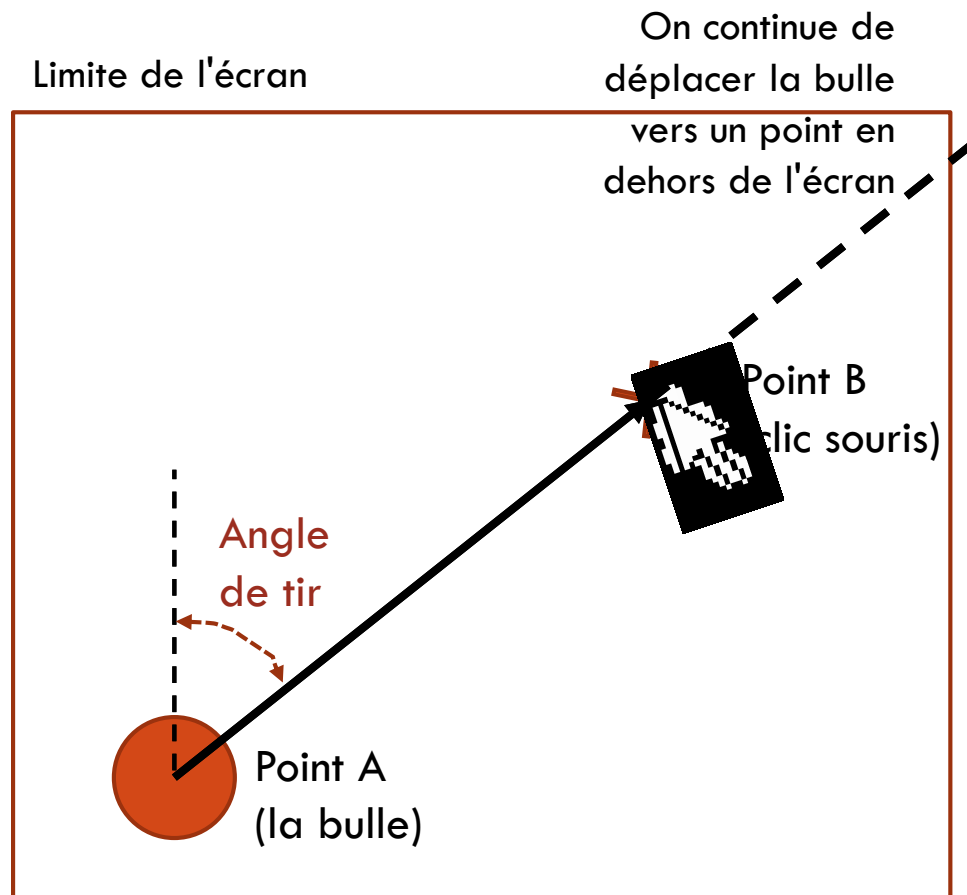
- Implémenter des animations est la tâche la plus courante dans le développement de jeux
 - Les principes appris dans le développement de ce jeu simple pourront être appliqués à la plupart des jeux
- Jeux HTML5 – beaucoup de discussions autour de l'élément `canvas`
 - On peut faire des traitements similaires en utilisant des techniques (HTML, CSS et JavaScript) plus traditionnelles
 - Jeux basés sur le DOM – plus grande compatibilité avec les navigateurs
 - On utilisera `canvas` plus tard

Mouvement d'une bulle

- Buts:
 - 1) Déplacer une bulle d'un point A vers un point B, en ligne droite, à vitesse constante
 - 2) Déterminer la position exacte des points A et B
 - Le joueur lance des bulles à partir de la même position – le point A aura toujours les mêmes coordonnées pour chaque nouvelle bulle tirée
 - Le point B correspond au point à l'écran où l'utilisateur a cliqué – il faut récupérer les coordonnées de la souris lors de clic de l'utilisateur dans la zone de jeu
- On va commencer par implémenter le mouvement d'un point A vers un point B
 - On doit calculer l'angle de tir en se basant sur les positions relatives de points A et B
 - Avec l'angle de tir, on peut déplacer la bulle dans une direction donnée, aussi loin que nécessaire – calculé avec précision plus tard
 - Pour l'instant ce sera vers un point situé en dehors de l'écran

Remarques sur le mouvement des bulles

- Dans la version finale, la bulle ne s'arrêtera pas là où l'utilisateur a cliqué
 - ▣ Elle va continuer jusqu'à rencontrer d'autres bulles
 - ▣ On va gérer les collisions par la suite
- Une fois qu'on aura implémenté le mouvement point-à-point, il sera plus facile de l'extrapoler et continuer à déplacer la bulle dans la même direction



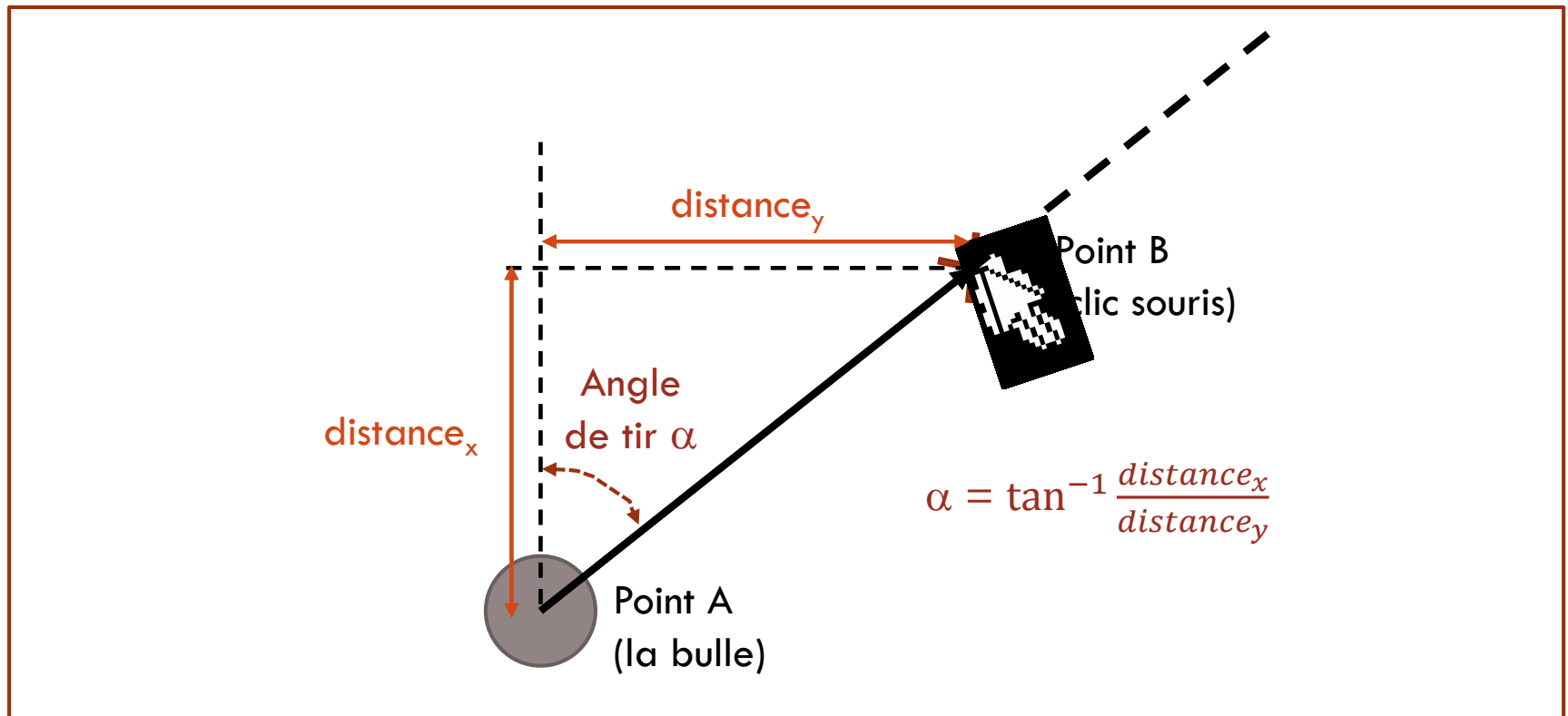
Théorie

Les sprites CSS

- Un *sprite* CSS est un élément 2D du jeu qui peut se déplacer à l'intérieur d'une zone du jeu
 - Fait partie d'une scène plus grande
 - Son mouvement n'affecte pas les données en arrière-plan
 - Dans notre cas, la bulle au point A est notre seul *sprite*
- Un *sprite* (dans cette approche orientée DOM) est un élément HTML (souvent un ensemble de `<div>`) avec un style CSS
- Le déplacement d'un tel élément sans modifier le reste de l'écran est facile
 - Un élément HTML ayant une position absolue définie via le CSS est affiché indépendamment des éléments HTML l'entourant
 - Le navigateur affiche les différents éléments à l'écran et gère leurs intersections – si on supprime un élément, le navigateur sait qu'il faut afficher l'élément qui se trouve en dessous

Rappels trigonométrie

- On peut calculer l'angle de tir d'une bulle en utilisant la fonction arctan (tangente inverse)



Déplacer un sprite CSS

- Pour déplacer un élément en manipulant le DOM, on doit seulement modifier les valeurs de ses propriétés CSS `top` et `left`
 - ▣ Pour chaque étape intermédiaire composant le mouvement
- Il est possible d'animer des *sprites* de 2 manières:
 - ▣ En utilisant des animations JavaScript
 - ▣ En utilisant les transitions CSS3 (plus tard)
- La méthode `animate` de jQuery permet d'animer le déplacement d'un objet
 - ▣ Calcule la différence entre les valeurs de départ et d'arrivée
 - ▣ Modifie les valeurs des propriétés `top` et `left` pour les étapes intermédiaires
 - ▣ Remarque:
 - Cette méthode ne fonctionne pas avec des propriétés CSS ne prenant pas des valeurs numériques (e.g. faire une transition de couleur sur un arrière-plan)

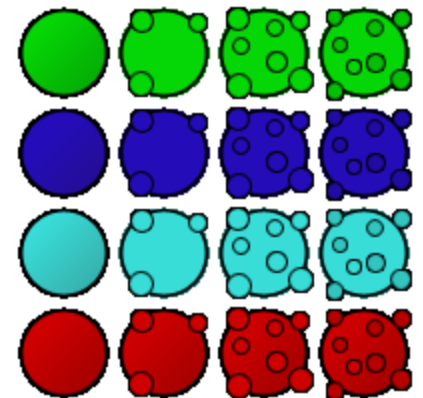
Théorie

La méthode animate de jQuery

- Arguments de la méthode animate:
 - Propriétés CSS à animer
 - Souvent top et left, mais toute propriété ayant des dimensions simples en pixels peut être utilisée (e.g. font-size, width, height, border-width, margin-left)
 - Durée en millisecondes de l'animation
 - Par défaut elle est fixée à 1s, avec une vitesse de 1000 pixels par seconde
 - La distance de déplacement d'une bulle dépendra de l'état du jeu, mais aussi des éventuels objets avec lesquels la bulle va entrer en collision
 - Pour l'instant, la durée telle qu'elle est définie suffit
 - Easing: définit comment un objet transite de son état de départ à son état final
 - Permet de faire varier l'accélération et la décélération par rapport à un tracé
 - Pour le mouvement:
 - linear résulte en une vitesse constante du début à la fin
 - swing ajoute une accélération au début et une décélération à la fin

Ajout de sprites

- Soit l'image suivante définissant des bulles
- On va utiliser 4 couleurs de bulles et chaque bulle pourra avoir 4 états
 - ▣ Pour aller plus vite, on utilise un seul fichier PNG contenant les 16 images possibles
- Via le CSS on va modifier la position de l'image d'arrière-plan
 - Plus rapide que de placer les images dans une balise ``



Animations CSS type explosion (1)

- Déplacer les *sprites* sur l'écran est un type d'animation
- Nous voulons maintenant modifier la manière dont les *sprites* sont affichés
 - ▣ Nous allons utiliser les autres images d'états de bulles présentes dans le fichier `bulles.png`
- Technique: pour afficher une animation type explosion, afficher les 4 images à la suite, en déplaçant à chaque fois l'image d'arrière-plan de 50px vers la gauche
- Au lieu d'appliquer l'effet explosion à un groupe de bulles, nous allons l'appliquer à chaque bulle individuellement

Animations CSS type explosion (2)

- Pour faire exploser chaque bulle individuellement, on doit modifier le corps de la fonction `exploderBulles(...)` dans `jeu.js`

```
var exploderBulles = function(listeBulles, temps) {  
    $.each(listeBulles, function() {  
        var bulle = this;  
        setTimeout(function() {  
            bulle.animerExplosion();  
        }, temps);  
        etatJeu.exploderBullePos(this.getLignes(), this.getColonnes());  
        setTimeout(function() {  
            bulle.getSprite().remove();  
        }, temps + 200);  
        temps += 60;  
    });  
};
```

`animerExplosion()` reste à définir dans `bulle.js`

L'explosion de la 1^{ère} bulle doit débiter dès que la collision est détectée. Les explosions des autres bulles du groupe seront retardées de 60ms.

Animations CSS type explosion (3)

- Ajoutez la fonction `animerExplosion(...)` dans `bulle.js`

```
etatjeu.js  index.html  bulle.js ×  jeu.js
23     this.setColonne = function(colonneDonnee) {colonnes = colonneDonnee;};
24
25     this.animerExplosion = function() {
26         var haut = type * that.getSprite().height();
27         this.getSprite().css(Modernizr.prefixed("transform"), "rotate(" + (Math.random() * 360) + "deg)");
28         setTimeout(function() {
29             that.getSprite().css("background-position", "-50px -" + haut + "px");
30         }, 125);
31         setTimeout(function() {
32             that.getSprite().css("background-position", "-100px -" + haut + "px");
33         }, 150);
34         setTimeout(function() {
35             that.getSprite().css("background-position", "-150px -" + haut + "px");
36         }, 175);
37         setTimeout(function() {
38             that.getSprite().remove();
39         }, 200);
40     };
41 }
```

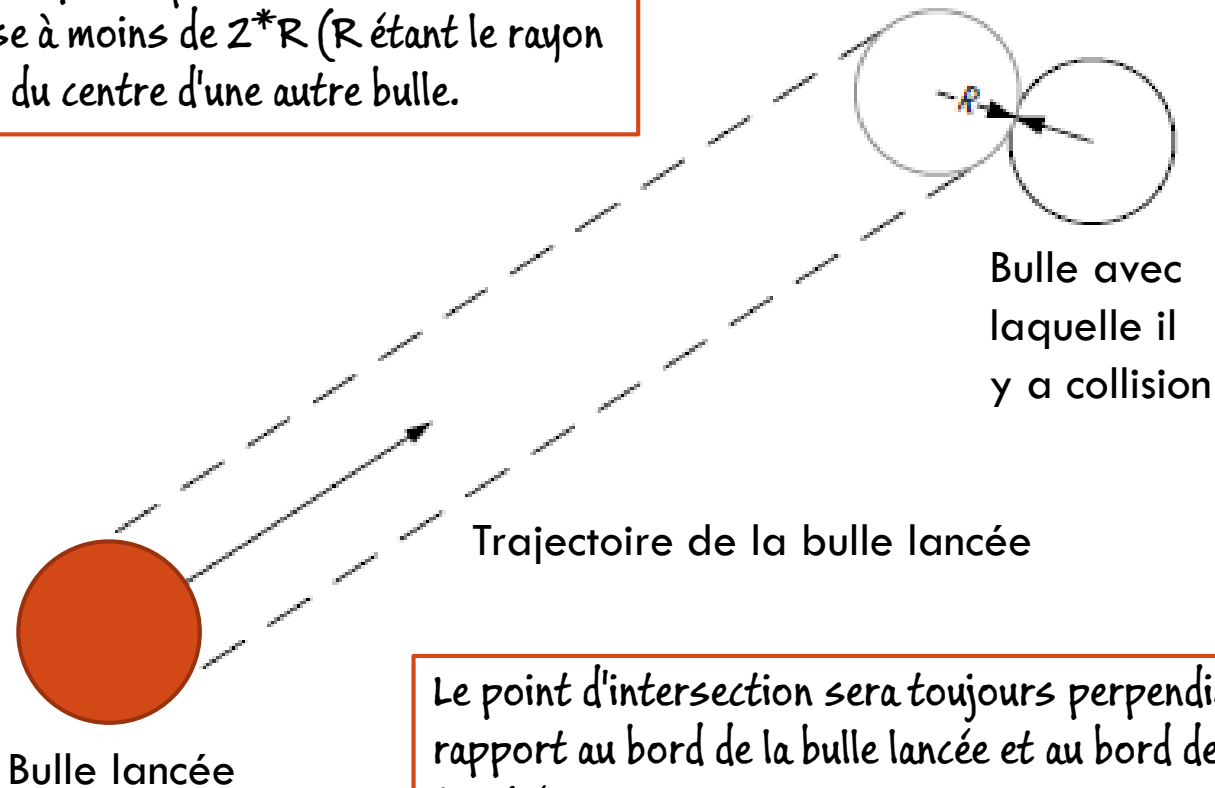
En se basant sur le type de bulle, la fonction calcule la valeur de la position haute de l'image d'arrière-plan. On ajoute ensuite une transformation CSS (rotation) pour faire tourner la bulle d'un angle aléatoire. La fonction permettant de décaler l'image d'arrière-plan est appelée 4 fois, chaque appel étant retardé de 25ms. Enfin le sprite est supprimé du DOM.

Détecter des collisions

- On calcule les collisions de 2 manières:
 - ▣ Option 1: on déplace un *sprite* de quelques pixels pour chaque image et on essaie de déterminer s'il y a superposition avec d'autres *sprites*
 - Si oui, on sait qu'on a touché une autre bulle
 - ▣ Option 2: on utilise la géométrie pour calculer où il peut y avoir une collision entre un *sprite* et notre bulle, avant même qu'on la déplace
- La 1^e option peut convenir pour les jeux d'arcade rapides
 - ▣ Les objets se déplacent vite, les détections de collisions se produisent après que l'objet se soit déplacé de plusieurs pixels (depuis la dernière détection)
- La 2^e option est plus précise si faisable
 - ▣ C'est le cas pour notre jeu, mais pas pour des jeux où les *sprites* vont avoir des formes plus complexes

Géométrie d'une collision

La bulle lancée provoquera une collision si son centre passe à moins de $2 \cdot R$ (R étant le rayon d'une bulle) du centre d'une autre bulle.

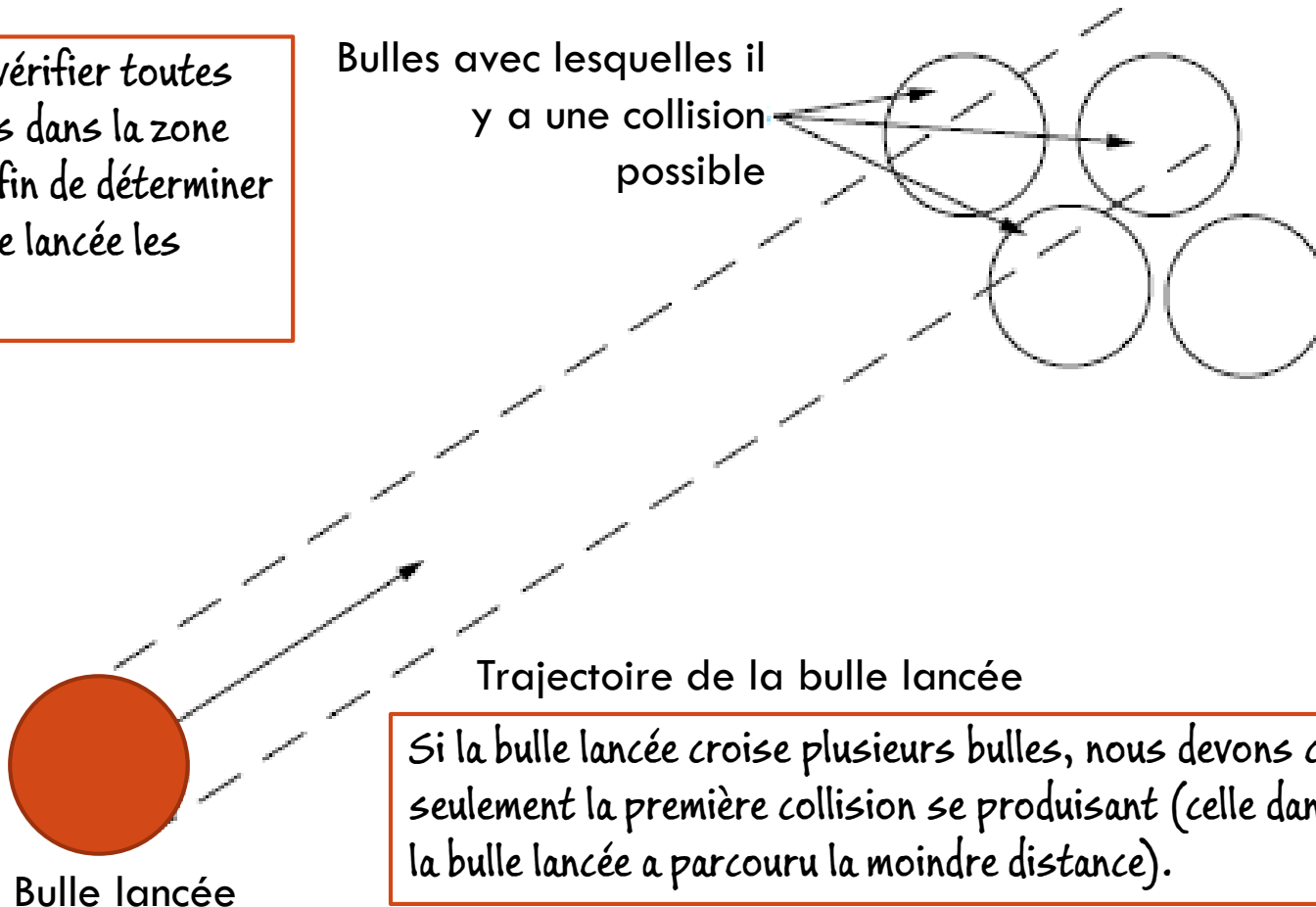


Le point d'intersection sera toujours perpendiculaire par rapport au bord de la bulle lancée et au bord de la bulle touchée.

Déterminer la bonne collision

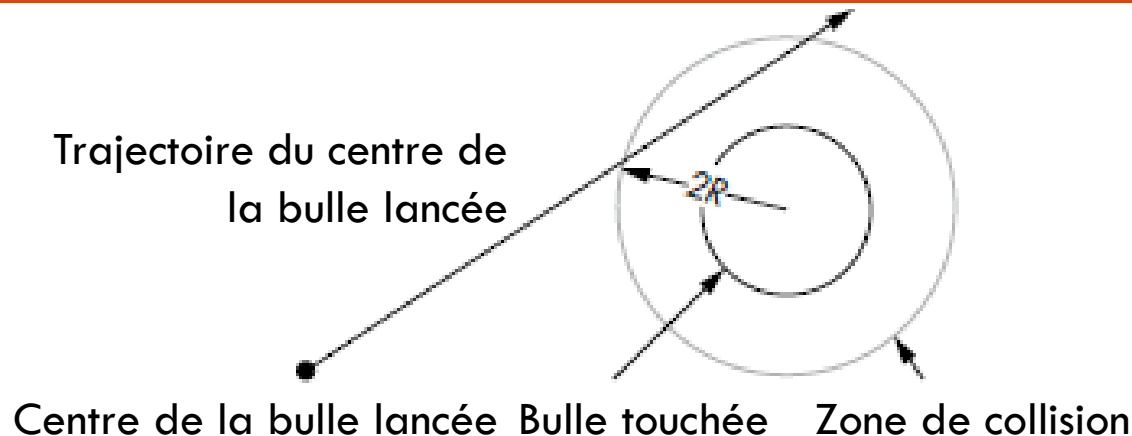
On doit vérifier toutes les bulles dans la zone de jeu, afin de déterminer si la bulle lancée les croise.

Bulles avec lesquelles il y a une collision possible



Détecter une collision

Détecter une collision revient à détecter lorsqu'un vecteur partant du centre de la bulle lancée va couper un cercle de rayon $2R$. Ce cercle constitue la zone de collision d'une bulle affichée dans la grille du jeu.



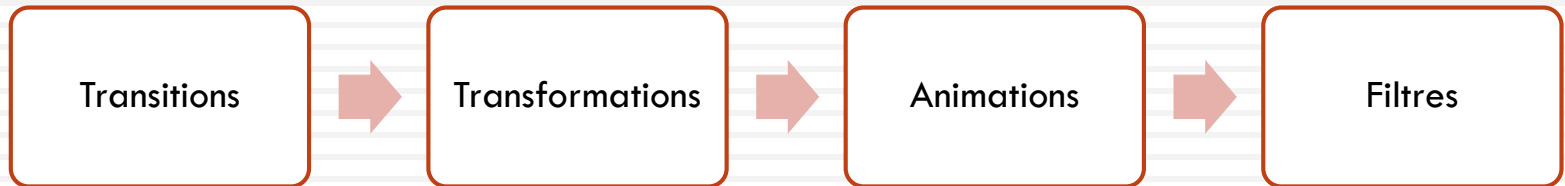
La figure est transformée en une formule mathématique utilisant des vecteurs.

Le code sera dans un fichier à part `detecteurCollision.js`

Calcul d'une collision

- Calcul du vecteur de la bulle du joueur
 - Equation du cercle: $(x - c_x)^2 + (y - c_y)^2 = r^2$
 - x et y sont des points sur la circonférence du cercle
 - c_x et c_y sont les coordonnées du centre du cercle
 - r est le rayon du cercle
 - Equation de la bulle lancée:
$$\begin{cases} p_x = e_x + td_x \\ p_y = e_y + td_y \end{cases}$$
 - p_x et p_y sont des points sur la trajectoire du centre de la bulle
 - Calculés dans la méthode `animate` de jQuery (équation standard pour un point se déplaçant sur une ligne)
- Calcul du point de cette trajectoire le plus proche du centre de la bulle à tester pour collision

Animer le contenu avec CSS



Avantages de CSS

- Ensemble d'attributs pour les transformations et transitions permettant d'animer les modifications apportées aux propriétés CSS
 - e.g. coordonnées `top` ou `left` d'un élément
- Au lieu d'utiliser JavaScript pour créer des animations image par image, les transitions CSS sont définies dans les feuilles de style associées au document
 - Une animation est débutée suite à une seule modification d'une propriété CSS
 - Au lieu d'avoir plusieurs modifications incrémentales d'une propriété (comme le veut JavaScript)
- Les animations CSS sont gérées par le moteur d'affichage du navigateur
 - Pas par l'interpréteur JavaScript – libère du temps processeur pour exécuter du code JavaScript par exemple
 - Si processeur graphique, les effets sont entièrement gérés par celui-ci – plus d'images par seconde lors de l'affichage des animations

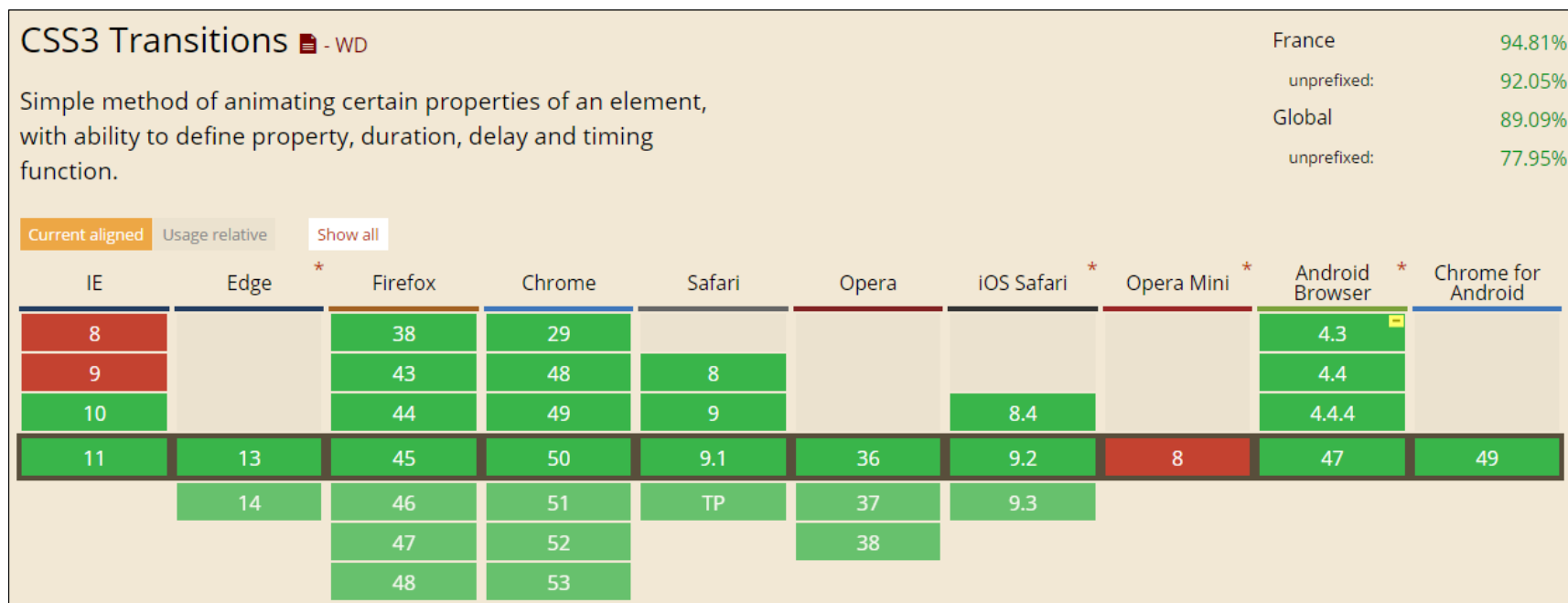


3.1 TRANSITIONS CSS

Les transitions CSS

- Une transition CSS permet de faire varier une ou plusieurs propriétés CSS d'un élément sur une période de temps donnée
 - La modification de la propriété `left` d'un élément du DOM de 50px à 500px aura pour effet de changer immédiatement sa position à l'écran
 - En appliquant une transition, on peut rendre cette modification plus graduelle
- Une transition CSS spécifie les éléments suivants:
 - La (ou les) propriété(s) CSS à animer
 - Toutes propriété CSS prenant une valeur numérique (e.g. `left`, `color`)
 - Pas pour les propriétés CSS prenant des valeurs booléennes (e.g. `visibility:hidden` et `visibility:visible`) – résolu si on anime `opacity` de 0 à 1
 - Liste de l'ensemble des propriétés CSS pouvant être animées:
https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties
 - Comment l'animation doit se produire
 - Combien de temps doit durer l'animation

Prise en charge des transitions CSS3



Source: <http://caniuse.com/#feat=css-transitions>

Transitions CSS de base

- 5 propriétés CSS peuvent être utilisées pour les transitions:

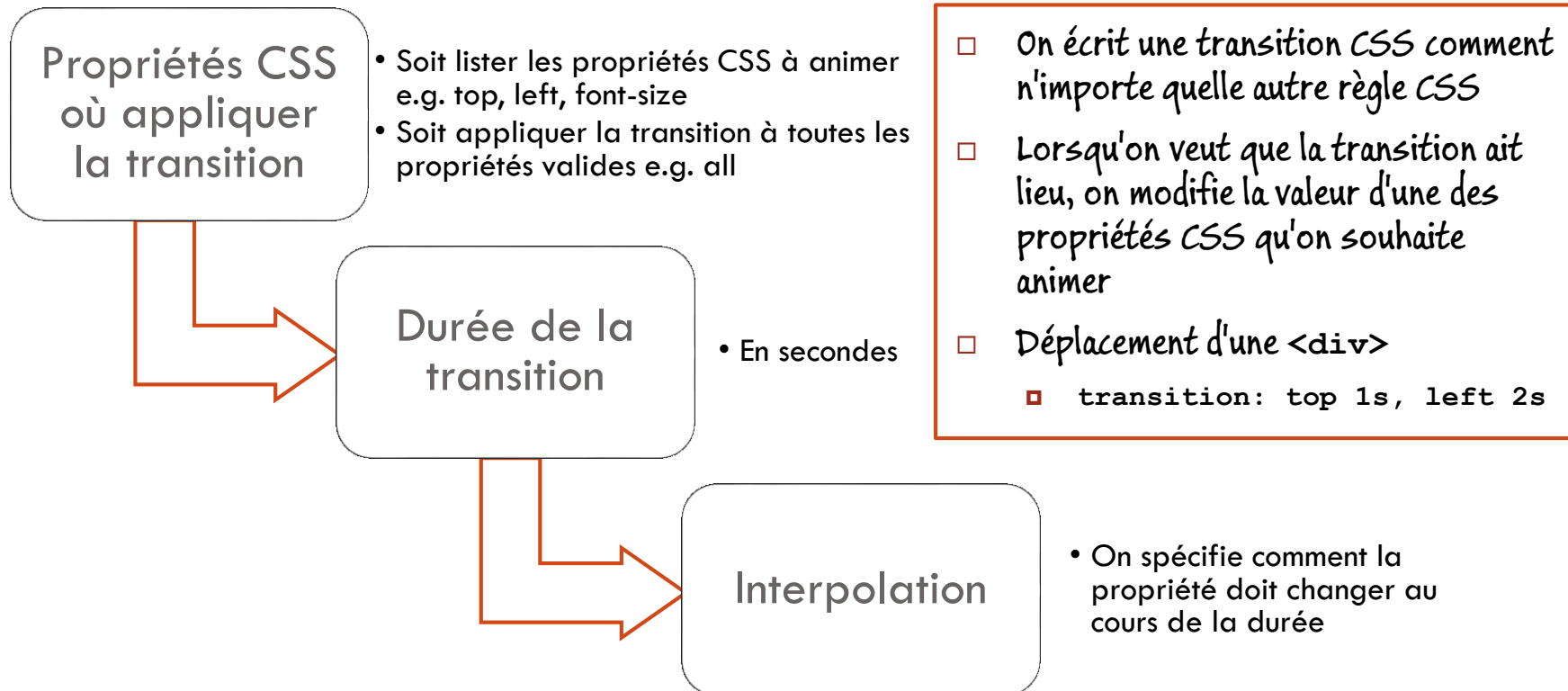
Propriété CSS	Description
<code>transition</code>	Raccourci pour spécifier les propriétés suivantes
<code>transition-property</code>	Spécifier la (ou les) propriété(s) CSS à laquelle (auxquelles) s'applique la transition
<code>transition-duration</code>	La durée de la transition en secondes ou millisecondes
<code>transition-timing-function</code>	Le type de courbe pour l'interpolation. Valeurs possibles: <code>ease</code> , <code>linear</code> , <code>ease-in</code> , <code>ease-out</code> , <code>ease-in-out</code> , <code>cubic-bezier</code>
<code>transition-delay</code>	Le délai à appliquer avant le début de la transition (en secondes ou millisecondes)

- Les moteurs récents implémentent la spécification
- Sur certaines des anciennes versions, tous les mots-clés spécifiques sont préfixés:
 - `webkit-` sous WebKit (Chrome, Safari), de `-moz-` chez Gecko (Mozilla), `-ms-` pour Microsoft, et `-o-` pour Opera.

Comment écrire une transition ?

Syntaxe pour la propriété transition:

transition: [propriété] [durée] [fonction_interpolation] [retard]



Exemple 1

Transition lors du survol d'un élément HTML

- Avant la transition

Hover over this heading to see its transition

- Après la transition

Hover over this heading to see its transition

- Code HTML de l'élément à animer

- `<h1>Hover over this heading to see its transition</h1>`

- Code CSS pour appliquer la transition sur les 2 propriétés considérées

- ```
h1 { font-size: 120%;
 color: blue;
 transition: font-size 2s ease-out 1s,
 color 2s ease-in 1s; }

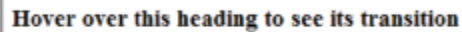
h1:hover {
 cursor: pointer;
 font-size: 180%;
```



# Exemple 2

## Transition lors du survol d'un élément HTML

- Avant la transition

A screenshot of a heading element with the text "Hover over this heading to see its transition". The text is in a standard font size and weight, and the background is white with a thin border.

- Après la transition

A screenshot of the same heading element after a transition. The text "Hover over this heading to see its transition" is now larger and bolder. A mouse cursor is visible over the text, indicating it is being hovered over.

- Code HTML de l'élément à animer

- `<h1>Hover over this heading to see its transition</h1>`

- Code CSS pour appliquer la transition sur une propriété (`font-size`)

- ```
h1 { font-size: 120%;  
    transition: font-size 2s ease-out;  
    -webkit-transition: font-size 2s ease-out; }
```

```
h1:hover { font-size: 180%; }
```

- Code CSS pour appliquer la transition sur toutes les propriétés

- `transition: all 2s ease-out 1s`

Exemple 3

Un accordéon

□ Code HTML de l'élément à animer

```
□ <h1>jQuery FAQs</h1>
  <div id="accordion">
    <h3><a href="#Q1">What is jQuery?</a></h3>
    <div id="Q1">Contents for first panel</div>
    <h3><a href="#Q2">Why is jQuery becoming so popular?</a></h3>
    <div id="Q2">Contents for second panel</div>
    <h3><a href="#Q3">Which is harder to learn: ...?</a></h3>
    <div id="Q3">Contents for third panel</div>
  </div>
```

□ Code CSS pour appliquer la transition sur toutes les propriétés

```
□ #accordion div {
  overflow: hidden;
  height: 0;
  transition: height 2s ease-in-out;
  -webkit-transition: height 2s ease-in-out;
}
#accordion div:target {
  height: 120px;
}
```

jQuery FAQs

What is jQuery?

Why is jQuery becoming so popular?

Three reasons:

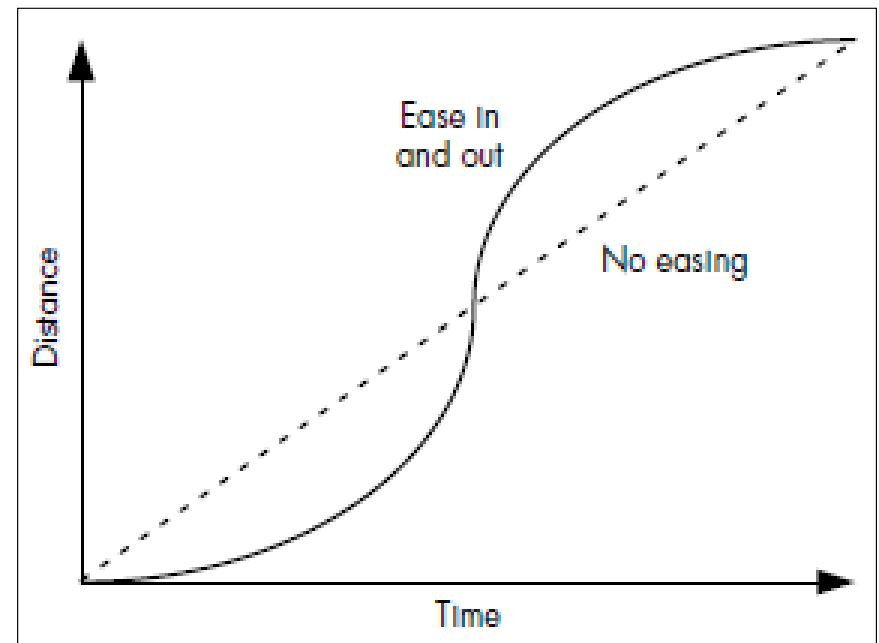
- It's free.
- It lets you get more done in less time.
- All of its functions are cross-browser compatible.

Which is harder to learn: jQuery or JavaScript?

Lorsque l'utilisateur clique sur un élément `<a>` on modifie la valeur de la hauteur (de 0 à 120px) pour la `<div>` cible (pseudo-classe CSS `div:target`).

Remarques sur l'interpolation

- Permet de spécifier comment (à quelle vitesse) doit la valeur d'une propriété doit être modifiée
 - ▣ Rythme constant, ou accéléré au début et ralenti à la fin
- S'applique à toutes les propriétés qu'on souhaite animer
 - ▣ Y compris la couleur



Exemple d'application à Tir Bulles

Déplacement du bouton "nouveau jeu"

- Nous voulons déplacer le bouton "Nouveau jeu" vers le bas de l'écran
- Ajoutez le code suivant à `main.css` dans la classe `.btn_debut_jeu`

```
79     transition: all .8s ease-in-out 1s;  
80     -moz-transition: all .8s ease-in-out 1s;  
81     -webkit-transition: all .8s ease-in-out 1s;  
82     -ms-transition: all .8s ease-in-out 1s;
```

- Ligne 79:
 - `all` spécifie à quelles propriétés on applique la transition CSS (ici toutes)
 - `.8s` spécifie la durée de la transition en secondes
 - `ease-in-out` précise le type d'interpolation utilisé (accélération initiale et décélération finale)
 - `1s` spécifie un délai de une seconde avant le début de l'animation
- Lignes 80-82 on spécifie la même transition pour tous les types de navigateurs

Changer la couleur des boutons

- Nous allons définir une transition CSS pour animer la couleur d'un bouton
- Ajoutez le code suivant à **main.css**

```
transition: background-color .3s ease-in-out;  
-moz-transition: background-color .3s ease-in-out;  
-webkit-transition: background-color .3s ease-in-out;  
-ms-transition: background-color .3s ease-in-out;
```

```
background-color: #900;
```

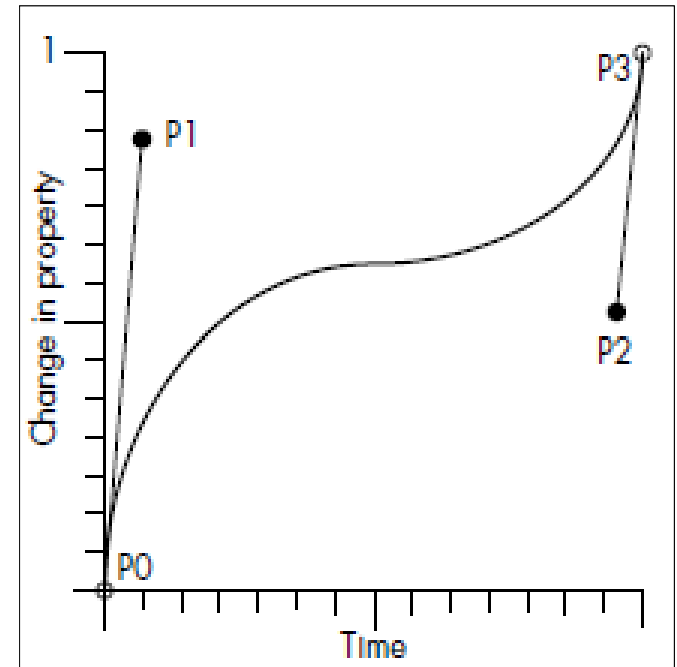
Classe
.btn_debut_jeu

Pseudo-classe
.btn_debut_jeu:
hover

- Nous utilisons une interpolation **ease-in-out**
 - ▣ Possibilité d'utiliser **ease**, **linear**, **ease-in** ou **ease-out**
- Toutes sont des alias pour des définitions spécifiques de **cubic-bezier**
 - ▣ **ease-in-out** correspond à **cubic-bezier(0.42, 0, 0.58, 1.0)**

Remarques sur les courbes de Bézier

- Les courbes de Bézier sont spécifiées en définissant les coordonnées de 2 points (P_1 et P_2)
 - ▣ Représentent les lignes tangentes à la courbe à son début (segment P_0P_1) et à sa fin (segment P_3P_2)
- Les coordonnées spécifiées dans le CSS correspondent aux points P_1 et P_2
 - ▣ Toujours comprises entre 0 et 1
 - ▣ P_0 correspond à l'origine (0,0) et P_3 correspond à la fin (1,1)
- L'angle entre l'axe vertical et P_1 et P_2 représente la pente de la courbe
- La longueur des segments P_0P_1 et P_2P_3 détermine l'importance de la courbe
- <http://cubic-bezier.com>





3.2 TRANSFORMATIONS CSS DE BASE

Introduction aux transformations CSS

- Une transformation CSS permet de manipuler la forme d'un objet
 - Translation de coordonnées 2D (x,y) ou 3D (x,y,z)
 - Mise à l'échelle de dimensions selon les axes x, y et z
 - Rotation d'un angle donné sur un des axes x, y et z
 - Inclinaison selon les axes x ou y
 - Ajout d'une perspective 3D
- Les transformations peuvent être animées avec des transitions
- Dans la plupart des navigateurs il est possible de modifier un objet en 2D/3D
 - Nous allons nous concentrer sur les transformations 2D (seuls les navigateurs webkit supportent les transformations 3D)
 - Pour les transformations 3D: <http://desandro.github.io/3dtransforms/>
- Référence:
 - <https://developer.mozilla.org/en-US/docs/Web/CSS/transform>
 - <https://www.w3.org/TR/css-transforms-1/>

Prise en charge de CSS3 Transforms

□ 2D – source: <http://caniuse.com/#feat=transforms2d>

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8		38	29					4.3	
9		43	48	8				4.4	
10		44	49	9		8.4		4.4.4	
11	13	45	50	9.1	36	9.2	8	47	49
	14	46	51	TP	37	9.3			
		47	52		38				
		48	53						

□ 3D – source: <http://caniuse.com/#feat=transforms3d>

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8		38	29					4.3	
9		43	48	8				4.4	
¹ 10		44	49	² 9		8.4		4.4.4	
¹ 11	13	45	50	² 9.1	36	9.2	8	47	49
	14	46	51	² TP	37	9.3			
		47	52		38				
		48	53						

Propriétés CSS pour les transformations 2D

□ Propriétés CSS

Propriété CSS	Description
<code>transform</code>	Permet d'appliquer une ou plusieurs transformations à un élément HTML
<code>transform-origin</code>	Spécifie le point d'origine pour la transformation considérée

□ Remarques:

- Par défaut, le point d'origine d'une transformation est placé au milieu de l'élément à transformer
 - 50% de la largeur, et 50% de la hauteur de l'élément

Méthodes CSS pour les transformations 2D

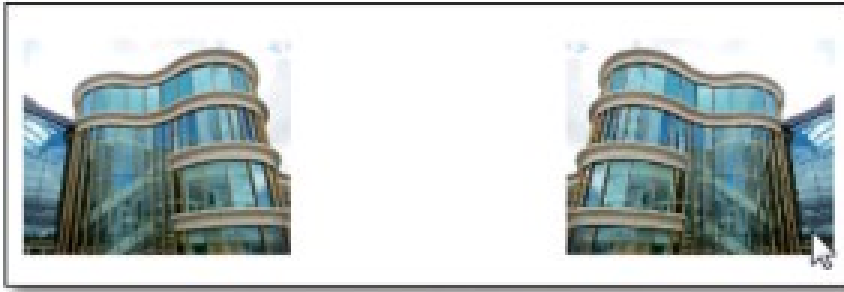
Méthode CSS	Description
<code>rotate (angle)</code>	Rotation d'un élément de l'angle spécifié
<code>rotateX (angle)</code> <code>rotateY (angle)</code>	Rotation d'un élément à la verticale, respectivement à l'horizontale
<code>scaleX (valeur)</code> <code>scaleY (valeur)</code>	Mise à l'échelle horizontale (modifie la largeur), respectivement verticale (modifie la hauteur)
<code>scale (valeur-x, valeur-y)</code>	Mise à l'échelle d'un élément en modifiant sa largeur et son hauteur
<code>skewX (angle)</code> <code>skewY (angle)</code>	Rotation d'un élément sur l'axe X et respectivement Y
<code>skew (angle-x, angle-y)</code>	Rotation d'un élément sur les axes X et Y
<code>translateX (valeur)</code> <code>translateY (valeur)</code>	Déplacement d'un élément horizontalement (à gauche ou à droite) et respectivement verticalement (en haut et en bas)
<code>translate (valeur-x, valeur-y)</code>	Déplacement d'un élément haut/bas et droite/gauche
<code>matrix (a,b,c,d,e,f)</code>	6 valeurs de la matrice pour appliquer les transformations précédentes avec une seule méthode

Exemple

Appliquer un effet miroir à une image

- Transformation souhaitée

- Appliquer une rotation avec une origine différente à l'image initiale



On affiche 2 copies de la même image côte à côte. Lorsque l'utilisateur survole l'image, elle est retournée de 180° (effet miroir). La rotation est appliquée à partir du côté droit de la 2^e image, d'où l'espace blanc entre les images.

- Code HTML de l'élément à transformer

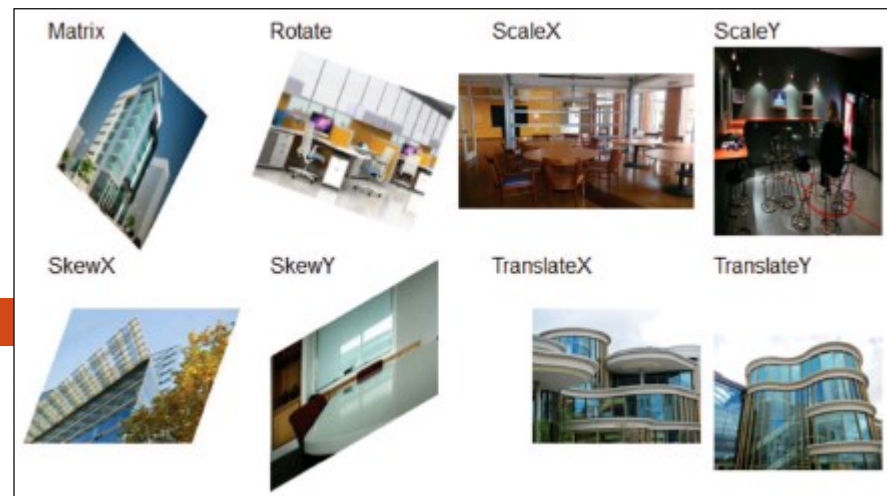
- `<p></p>`

- Code CSS pour appliquer la transformation souhaitée sur l'image

- ```
.image1 {
 transition: 2s; }
.image1:hover {
 transform: rotateY(180deg);
 transform-origin: right; }
```

On définit une transition de 2s pour toutes les modifications apportées aux élément ayant pour classe .image1.

# Exemple Galerie d'images



## □ Code HTML pour les images

### □ <ul>

```
<p>Matrix</p>
<p>Rotate</p>
<p>ScaleX</p>
<p>ScaleY</p>
<p>SkewX</p>
<p>SkewY</p>
<p>TranslateX</p>
<p>TranslateY</p>

```

## □ Code CSS pour appliquer la transformation souhaitée sur l'image

```
.image1 {transform: matrix(0.5,0.5,-0.5,1,0,0);}
.image2 {transform: rotate(20deg);}
.image3 {transform: scaleX(1.4);}
.image4 {transform: scaleY(1.4);}
.image5 {transform: skewX(-20deg);}
.image6 {transform: skewY(-30deg);}
.image7 {transform: translateX(30px);}
.image8 {transform: translateY(20px);}
```

# Remarques additionnelles

- Appliquer l'effet `ease-in-out` à toutes les propriétés CSS pouvant recevoir des transitions, avec la même durée:
  - ▣ `transition: all .3s ease-in-out`
- Si on souhaite avoir des effets et des durées différentes:
  - ▣ `transition: background-color .2s ease-in-out, transform 0.2s linear;`
  - ▣ Cette solution permet de réduire les effets de bord, lorsque vous modifiez les valeurs d'autres propriétés CSS que vous ne souhaitez pas animer
- Lorsqu'on applique des transitions à des propriétés `transform` individuelles, on doit quand même définir les transitions associées pour les autres navigateurs
  - ▣ `transition: background-color .3s ease-in-out, transform .2s ease-in-out;`
  - ▣ `-moz-transition: background-color .3s ease-in-out, -moz-transform .2s ease-in-out;`
  - ▣ `-webkit-transition: background-color .3s ease-in-out, -webkit-transform .2s ease-in-out;`
  - ▣ `-ms-transition: background-color .3s ease-in-out, -ms-transform .2s ease-in-out;`

*Appliquer ces corrections dans main.css*

# Exemple de transformation 3D (1)

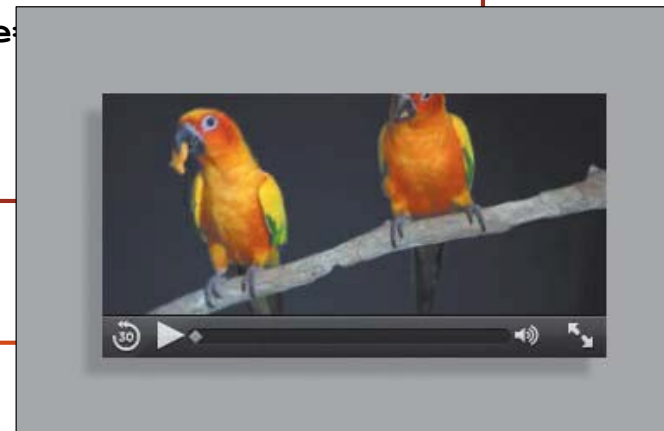
- Soit le code HTML et le code CSS ci-dessous

```
<div class="main">
 <div class="bg">
 <video controls preload="metadata">
 <source src="parrots-small.mp4" type="video/mp4" />
 </video>
 </div>
</div>
```

```
div.main {
 position: absolute;
 top: 50%;
 left: 50%;
 margin-left: -350px;
 margin-top: -233px;
}
```

```
div.bg {
 height: 246px;
 width: 380px;
 background-color: #aaa;
}
```

```
video {
 width: 292px;
 position: absolute;
 top: 50px;
 left: 50px;
 -webkit-box-shadow: -10px 10px 5px #888;
}
```



- Affichage: (pas de 3D)

# Exemple de transformation 3D (2)

- Si on modifie le code CSS comme indiqué ci-dessous

```
div.main {
position:absolute;
top:50%;
left:50%;
margin-left:-350px;
margin-top:-233px;
-webkit-perspective:1000;
}
```

```
div.bg {
height:246px;
width:380px;
background-color:#aaa;
-webkit-transform:rotate3d(0,1,0, 40deg);
}
```

- On obtient l'affichage suivant:





# Inconvénients des transitions CSS

Plus on ajoute des animations à un élément, plus les transitions CSS sont difficiles à composer

- Exemple: déplacer un élément de 100px sur 1s et augmenter sa taille de 10px sur 2s
  - Nécessite la définition de transitions CSS différentes pour chacune des propriétés à animer
  - Il faut pouvoir faire continuer la mise à échelle de l'élément après son déplacement (très difficile si vous souhaitez déplacer cet élément à nouveau)

L'interpolation peut altérer la manière dont le mouvement est perçu, mais ce mouvement est toujours effectué en ligne droite

- Un mouvement selon une courbe est une suite de mouvements sur des petits segments – dans ce cas, on peut tout aussi bien utiliser JavaScript

Une fois démarrée, il est impossible d'interroger et de modifier une transition CSS

- Le navigateur gère la transition et la mise à jour de la position de l'élément, dès que la propriété CSS à animer est modifiée
- L'élément peut n'être affiché que jusqu'à la moitié du trajet, mais le DOM dira qu'il a terminé son déplacement – il faut attendre la fin d'une animation pour pouvoir récupérer la position d'un élément



## 3.3 ANIMATIONS CSS3

# Animations CSS3

- Les animations CSS3 vous permettent de créer des animations image par image
  - ▣ Similaire au fonctionnement de Flash
  - ▣ Peuvent être simples ou complexes, selon le nombre de propriétés à animer
- Syntaxe pour définir une animation CSS3
  - ▣ `animation: [nom] [durée] [vitesse] [retard] [itérations] [direction];`

# Propriétés CSS3 à utiliser pour définir une animation

Propriété CSS	Description
<code>animation</code>	Propriété raccourci pour spécifier en une seule ligne des valeurs pour toutes les propriétés qui suivent.
<code>animation-name</code>	Nom d'un sélecteur <code>@keyframes</code> définissant les images-clé à utiliser pour l'animation.
<code>animation-duration</code>	Durée de l'animation en secondes ou millisecondes
<code>animation-delay</code>	Durée en secondes ou millisecondes à respecter avant de débiter l'animation
<code>animation-iteration-count</code>	<code>infinite</code> soit une valeur précisant le nombre de répétitions
<code>animation-timing-function</code>	La courbe de vitesse à utiliser pour l'animation
<code>animation-direction</code>	Spécifie la direction de l'animation. Valeurs possibles: <code>normal</code> , <code>reverse</code> ou <code>alternate</code> .

# Exemple – pas de définition des images clés

## Déplacer un titre et changer sa couleur

- Avant la transition – on débute en bleu, avec une marge gauche à 20%

This text will animate.

- Après la transition – on finit en rouge, avec une marge gauche à 60%

This text will animate.

- Code HTML de l'élément à animer

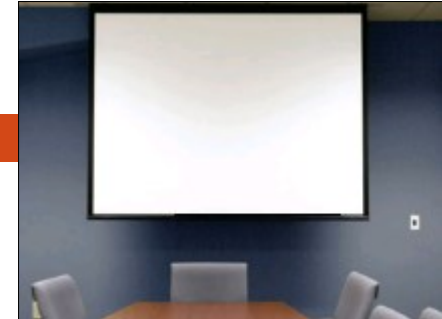
- `<h1>This text will animate.</h1>`

- Code CSS pour appliquer la transition sur les 2 propriétés considérées

- ```
h1 {animation: moveright 3s 2s infinite ease-in-out alternate; }
@keyframes moveright {
  from { margin-left: 20%;
        color: blue; }
  to {   margin-left: 60%;
        color: red; }
}
```

Exemple – définition des images clés

Diaporama



□ Code HTML de l'élément à animer

```
□ <ul>
  □ <li>
    <h2>Front of Building</h2>
    
  </li>
  ...
</ul>
```

□ Code CSS des éléments ul et li

```
□ ul { list-style: none;
      width: 50%;
      position: relative; }
ul li { width: 20%;
      float: left; }
```

□ Code CSS pour appliquer la transition sur les 2 propriétés considérées

```
□ ul { animation: slideshow 15s infinite alternate; }
  □ @keyframes slideshow {
    0% {left: 0%;}
    10% {left: 0%;}
    20% {left: -100%;}
    30% {left: -100%;}
    40% {left: -200%;}
    50% {left: -200%;}
    60% {left: -300%;}
    70% {left: -300%;}
    80% {left: -400%;}
    90% {left: -400%;}
    100% {left: -400%;}
  }
```



3.4 FILTRES CSS3

Filtres CSS3

- Les filtres CSS3 permettent de changer l'apparence d'une image
 - ▣ Après qu'elle ait été chargée dans le navigateur, sans modifier le fichier d'origine
 - ▣ Impossible à coupler avec des transitions
- Syntaxe
 - ▣ `filter: filtermethod(valeur)`
- Premier navigateur à prendre en charge les filtres CSS3: Firefox 35 (2015)

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8		38	29					4.3	
9		43	48	8				4.4	
10		44	49	9		8.4		4.4.4	
11	13	45	50	9.1	36	9.2	8	47	49
	14	46	51	TP	37	9.3			
		47	52		38				
		48	53						

Source: <http://caniuse.com/#feat=css-filters>

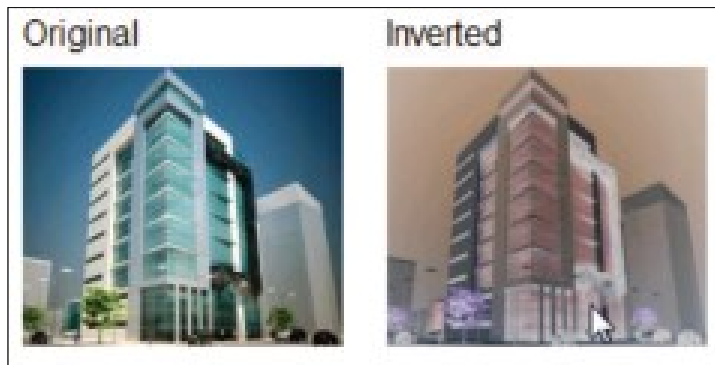
Méthodes CSS3 à utiliser pour définir un filtre

Méthode CSS	Description
<code>blur (valeur)</code>	Applique un flou gaussien. Valeur en pixels
<code>brightness (valeur)</code>	Ajuste la luminosité, de 0% (noir) à 100% (non modifié). Les valeurs supérieures à 100% résultent en une image plus lumineuse.
<code>contrast (valeur)</code>	Ajuste le contraste, de 0% (noir) à 100% (non modifié).
<code>drop-shadow (valeurs)</code>	Similaire à la propriété <code>box-shadow</code> . Affiche une ombre portée
<code>grayscale (valeur)</code>	transforme l'image en niveaux de gris. 100% totalement en niveaux de gris, 0% laisse l'image inchangée
<code>hue-rotate (angle)</code>	Ajuste la teinte de l'image, selon une rotation sur le cercle des couleurs.
<code>invert (valeur)</code>	Inverse les couleurs de l'image (100% complètement inversé, 0% laisse l'image inchangée).
<code>opacity (valeur)</code>	Rend l'image transparente: 0 pour une image totalement transparente, 1 pour une image inchangée.
<code>saturate (valeur)</code>	Variation de la saturation de l'image (0% saturée, 100% inchangée)
<code>sepia (valeur)</code>	Image en sépia (100% sépia, 0% inchangée)

Exemple 1

Inverser les couleurs d'une image

- Résultat attendu



- Code HTML de l'élément à animer

- ```
Original

Inverted

```

- Code CSS du filtre

- ```
.image1:hover { filter: invert(.80); }
```

Exemple 2

Tous les filtres



Code HTML de l'élément à animer

```
<ul>  
  <li>Blur<br>    </li>  
  <li>Brightness<br>    </li>  
  <li>Contrast<br>    </li>  
  <li>Drop Shadow<br>    </li>  
  <li>Grayscale<br>    </li>  
  <li>Hue Rotate<br>    </li>  
  <li>Invert<br>    </li>  
  <li>Opacity<br>    </li>  
  <li>Saturate<br>    </li>  
  <li>Sepia/blur<br>    </li>  
</ul>
```

Code CSS du filtre

```
.image1 { filter: blur(2px); }  
.image2 { filter: brightness(50%); }  
.image3 { filter: contrast(50%); }  
.image4 { filter: drop-shadow(2px 2px 5px #333); }  
.image5 { filter: grayscale(50%); }  
.image6 { filter: hue-rotate(90deg); }  
.image7 { filter: invert(.8); }  
.image8 { filter: opacity(.50); }  
.image9 { filter: saturate(30%); }  
.image10 { filter: sepia(100%) blur(1px); }
```

4^e
partie

HTML5 pour le multimédia



4.1 FORMATS ET CODECS VIDÉO ET AUDIO

Définitions

- Un **type de média** est un conteneur pour plusieurs composants, y compris:
 - Une piste vidéo encodée
 - Une ou plusieurs pistes audio encodées
 - Des métadonnées
 - Des marqueurs
- Pour lire un type de média, un navigateur nécessite une **lecteur média** pour ce type
- Pour des types de médias spécifiques, les navigateurs nécessitent des **plugins** spéciaux
- Un navigateur supporte nativement un type de média s'il n'a pas besoin d'un plugin pour lire ce type de média

Types de médias

- Exemples: MPEG, AVI, MP3, etc.
- Les **types de médias** représentent des conteneurs d'informations/données qui sont exploités par les **lecteurs médias** afin de lire leur contenu
 - Exemple: un fichier MPEG va contenir une piste vidéo, mais aussi une ou plusieurs pistes son
- Un type de média peut aussi contenir:
 - des marqueurs – pour synchroniser la piste vidéo avec la (ou les) piste(s) audio
 - des métadonnées – titre de la vidéo, images en lien avec la vidéo, longueur totale, droits numériques associés à l'œuvre, etc.
- Pour certains types de médias, le navigateur aura besoin d'un **plugin**
 - Souvent édités par les éditeurs du lecteur média – Apple fournit le plugin QuickTime, Adobe fournit le Flash Player plugin, etc.
- Si un navigateur supporte un type de média sans nécessiter de plugin, on dit que le navigateur supporte ce type "nativement"

Types de médias vidéo

Type média	Description
MPEG-4	Extensions *.mp4 ou *.m4v. Basé sur le type média *.mov (Apple QuickTime). Nouveau format de fichier utilisé par iTunes pour fournir des vidéos.
Flash Video	Extensions *.flv ou *.f4v. Initialement conçu par Macromedia, puis racheté par Adobe, la vidéo Flash fut le format le plus populaire pour fournir des vidéos durant la dernière décennie.
Ogg	Extension *.ogg, open-source et open-standard, nativement pris en charge par Firefox, Chrome et Opera. Ogg utilise le codec Theora pour la compression du flux vidéo.
WebM	Type de fichier relativement nouveau (*.webm), pris en charge nativement par Firefox, Chrome, Opera et IE9+. Permet la lecture de fichiers vidéo directement dans le navigateur (HTML5). Adobe a annoncé que les futures versions de Flash supporteraient le WebM. Repose sur le codec VP8.
ASF	Advanced Systems Format correspond à l'extension *.asf. Type de média propriétaire de Microsoft, spécifiquement conçu pour le streaming. Le flux vidéo d'un ASF est typiquement du WMV (Windows Media Video)
AVI	Audio Video Interleave (*.avi) est aussi un format propriétaire de Microsoft (1992).

Types de médias audio

Type média	Description
MP3	MPEG-1 Audio Layer 3 est un des types de médias les plus utilisés pour les fichiers audio.
AAC	Advanced Audio Coding (AAC) est le format utilisé par Apple pour fournir de l'audio sur le iTunes Store. AAC a été initialement conçu pour offrir une meilleure qualité audio que le MP3.
Ogg	Ogg est le nom du format conteneur pour l'audio, la vidéo et les métadonnées de Xiph.org. Vorbis est le nom d'un système de compression audio spécifiquement développé pour être contenu dans des conteneurs Ogg (possibilité d'intégrer d'autres formats e.g. FLAC ou Speex). Vorbis est comparable aux autres formats utilisés pour stocker et lire de la musique numérique, comme MP3, VQF, AAC. Il est différent de ces formats car il est complètement gratuit, ouvert, et non breveté.
WMA	Windows Media Audio (WMA) correspond aux extensions *.wma. C'est le système de compression propriétaire de Microsoft, utilisé pour encoder le flux audio des conteneurs ASF.

Etapes du processus de lecture d'un type de média par un lecteur média



Etape 1

Détermine le type de média que l'utilisateur souhaite lire



Etape 2

Détermine s'il est capable de décoder les flux vidéo et audio contenus.



Etape 3

Décode la vidéo et affiche le résultat à l'écran



Etape 4

Décode l'audio et envoie le résultat au périphérique de lecture.



Etape 5

Interprète les éventuelles métadonnées et rend le résultat disponible.

Codecs vidéo (1)

- ❑ A l'intérieur d'un type de média, la piste vidéo est encodée avec un type spécifique de code
- ❑ codec = algorithme de COmpression et DECompression de la vidéo (ou COdage et DECodage)
- ❑ Pas de codec vidéo précis prévu dans la spécification HTML 5
- ❑ Plusieurs codecs sont supportés par les navigateurs:

VP8/9	MPEG 4	H.264	Theora	Opus	Dalaa
<ul style="list-style-type: none">• Initialement développé par On2 Technologies, puis racheté par Google.• Open source, gratuit, il est équivalent en qualité à H.264.• Aussi connu sous le nom WebM	<ul style="list-style-type: none">• Standard avec diverses implémentations. E.g. Xvid est open source mais cela n'empêche pas l'existence de brevets sur le codec.	<ul style="list-style-type: none">• Version de 1993 de MPEG 4 pour les appareils moins performants• Fournit une haute compression pour une qualité réduite.• Utilisé par Youtube pour la HD et par Blu-Ray.	<ul style="list-style-type: none">• Codec libre et open source.• Ne supporte pas l'accélération matérielle comme le fait H.264.	<ul style="list-style-type: none">• Codec totalement libre et gratuit pour remplacer H.264.	<ul style="list-style-type: none">• Codec en cours de définition par Xiph et Mozilla• Écrit pour être supérieur à H.265 et donc nettement plus efficace que H.264.

Codecs vidéo implémentés par la balise <video>

Theora

Ogg reconnu par la balise <video>, l'extension du fichier suffit à identifier le format

Liens utiles: <http://firefogg.org/>, <http://v2v.cc/~j/ffmpeg2theora/>

webM / VP8 / VP9

WebM a un conteneur Matroska, comme Theora est celui de Ogg. Des plugins WebM sont disponibles pour IE et Safari, les autres navigateurs n'en ayant pas besoin, pas plus que les logiciels comme VLC ou Miro.

Liens utiles: <http://www.webmproject.org/>

H.264

Soutenu par Apple et Microsoft, qui font partie des ayants droit, largement utilisé dans le monde de la vidéo. Son utilisation est payante dans la plupart des cas, d'où une implémentation restreinte dans les navigateurs. Léger avantage sur WebM quand à la qualité de l'image mais cela tiendrait à son implémentation plus qu'au format. Son successeur H.265 est encombré de brevets.

Thor

Lancé par Cisco en tant qu'alternative à H.265, totalement gratuit et open-source

Liens utiles: <https://github.com/cisco/thor>

Codecs audio (1)

- Il existe des douzaines de codecs audio disponibles
 - ▣ Il y a plus de codecs audio que de codecs vidéo – en lien avec le marché de la musique en ligne (mobiles, smart TVs, consoles, etc.)
- Un codec audio réalise les tâches suivantes:

Décoder une portion
d'un fichier audio ou
vidéo,



La transformer en
une onde sonore
audible,



L'envoyer au
périphérique de
lecture de son du
terminal (onde
transformée en son)

- Différence entre audio et vidéo: l'audio peut stocker plusieurs canaux, permettant au son d'être transmis à différentes enceintes au même moment.
 - ▣ La plupart des fichiers audio contiennent 2 canaux (gauche/droite)
 - ▣ Il existe des types de média vidéo qui ont plusieurs canaux (5.1 ou 7.1)

Codecs audio (2)

- Les codecs audio ont souvent le même nom que le type de média auquel ils sont associés

AAC	FLAC	MP3	Vorbis	WMA
<ul style="list-style-type: none">• Standard d'encodage pour le type média *.aac.• Utilisé par Apple sur tous ses produits• Aussi sur Nintendo Dsi, Wii et 3DS, Sony PS3 et Portable• Mobiles: Sony Ericsson, Nokia, Android et WebOS	<ul style="list-style-type: none">• Free Lossless Audio Codec• Gratuit, open-source• Grand taux de compression – réduction de la taille des fichiers audio d'environ 60%	<ul style="list-style-type: none">• MPEG-1 Audio Layer 3• Un des types de fichiers audio les plus utilisés	<ul style="list-style-type: none">• Typiquement contenu dans l'extension *.ogg• Appelé Ogg Vorbis• Supporté nativement par la plupart des installations Linux• Aussi par les navigateurs Chrome, Firefox et Opera.	<ul style="list-style-type: none">• Windows Media Audio• Contenu dans l'extension *.wma• Utilisé pour encoder les flux audio des fichiers *.asf

Support des codecs audio et vidéo dans les navigateurs (1)

- Le problème lorsqu'on ajoute de l'audio ou de la vidéo dans un site Web c'est qu'il n'existe pas une combinaison unique de types audio et vidéo qui fonctionne sur l'ensemble des navigateurs aujourd'hui

Navigateur	Types audio				Types vidéo		
	Ogg Vorbis	MP3	AAC	WebM	Ogg Theora	MP4	VP8 (WebM)
Chrome	Oui	Oui	Oui	Oui	Oui	Oui	Oui
IE	Non	9.0+	Non	Non	Non	9.0+	Non
Firefox	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Safari	Non	Oui	Oui	Non	Non	Oui	Non
Opera	Oui	Oui	Oui	Oui	Oui	Oui	Oui

Support des codecs audio et vidéo dans les navigateurs (2)

- Une solution à ce problème est d'encoder vos médias au format Flash (*.swf) et d'utiliser le plugin Flash Player du navigateur pour lire le média
 - Problème 1: vous forcez les utilisateurs à utiliser un plugin pour lire vos médias
 - Même si le plugin est installé sur environ 97% des navigateurs, ce plugin consomme des ressources et il reste quand même 3% d'utilisateurs qui ne pourront pas lire le média
 - Problème 2: cela n'est pas adapté pour les terminaux mobiles ne supportant pas Flash (e.g. iPhone)
 - Segment représentant plus de 50% du trafic Internet, mais pourtant ignoré
 - Problème 3: le lecteur Flash n'a pas un comportement stable dans tous les navigateurs (crashe parfois)
 - En effet, les éléments audio et vidéo de HTML5 permettent d'éliminer le besoin de Flash


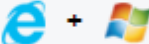

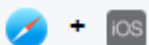
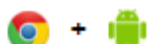
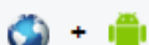
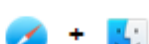

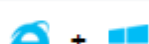
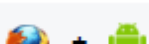
Codecs vidéo et audio pour HTML5

- Jusqu'à récemment, le principal challenge pour la vidéo avec HTML5 était la division au niveau de la prise en charge des codecs vidéo:



- Problème résolu par Firefox en supportant H.264 sous Windows (22+), Linux (26+) et Mac (35+)
- H.264 peut maintenant être lu partout, même sous IE8 et Opera en utilisant un conteneur Flash
- Pour les futurs codecs, la situation est inversée:
 - VP9 open-source est pris en charge par plusieurs navigateurs, mais pas HEVC
 - Or la combinaison VP9/HEVC permet d'avoir la même qualité vidéo que H.264/VP8 en utilisant la moitié de la bande passante

Codecs vidéo et audio pour HTML5

VIDEO CODECS	BROWSER	AUDIO CODECS
H.264, VP8, VP9		AAC, MP3, Vorbis, Opus
H.264		AAC, MP3
H.264, VP8, VP9		AAC, MP3, Vorbis, Opus
H.264		AAC, MP3
H.264, VP8, VP9		AAC, MP3, Vorbis
H.264, VP8		AAC, MP3, Vorbis
H.264		AAC, MP3
VP8, VP9		Vorbis, Opus
H.264		AAC, MP3
H.264, VP8, VP9		AAC, Vorbis, Opus

Source:
<https://www.jwplayer.com/html5/#browser-market-shares>

Prise en charge de l'audio et de la vidéo par les navigateurs actuels

Navigateur	H264	H265	VP8	VP9	AAC	MP3	VORBIS	OPUS
Chrome Desktop	30+	-	30+	30+	30+	30+	30+	33+
Internet Explorer (Windows)	9+	-	-	-	9+	9+	-	-
Firefox Desktop	22+ ¹	-	20+	28+	22+ ¹	22+ ¹	20+	20+
Safari (Mac)	3+	-	-	-	3+	3+	-	-
Opera Desktop	-	-	11+	16+	-	-	11+	12+
Safari (iOS)	3+	-	-	-	3+	3+	-	-
Stock Browser (Android)	2.3+	-	4.0+ ²	-	2.3+	2.3+	4.0+ ³	-
Chrome (Android)	30+	- ⁴	30+ ²	-	30+	30+	30+	-
Internet Explorer (WinPhone)	7.5+	-	-	-	7.5+	7.5+	-	-
Firefox (Android)	20+	-	20+ ²	28+ ²	20+	-	20+	20+

1. Firefox supporte H264/AAC/MP3 sur OSX à partir de la version 35.
2. Décodage logiciel seulement, impacte la performance et la durée de vie de la batterie.
3. `canPlayType()` ne retourne rien sur Vorbis, mais lecture de l'audio
4. Android Lollipop contient un décodeur logiciel qui lit le H265 dans Chrome.

Semestre Printemps 2017

Source: <https://www.jwplayer.com/html5/formats/>

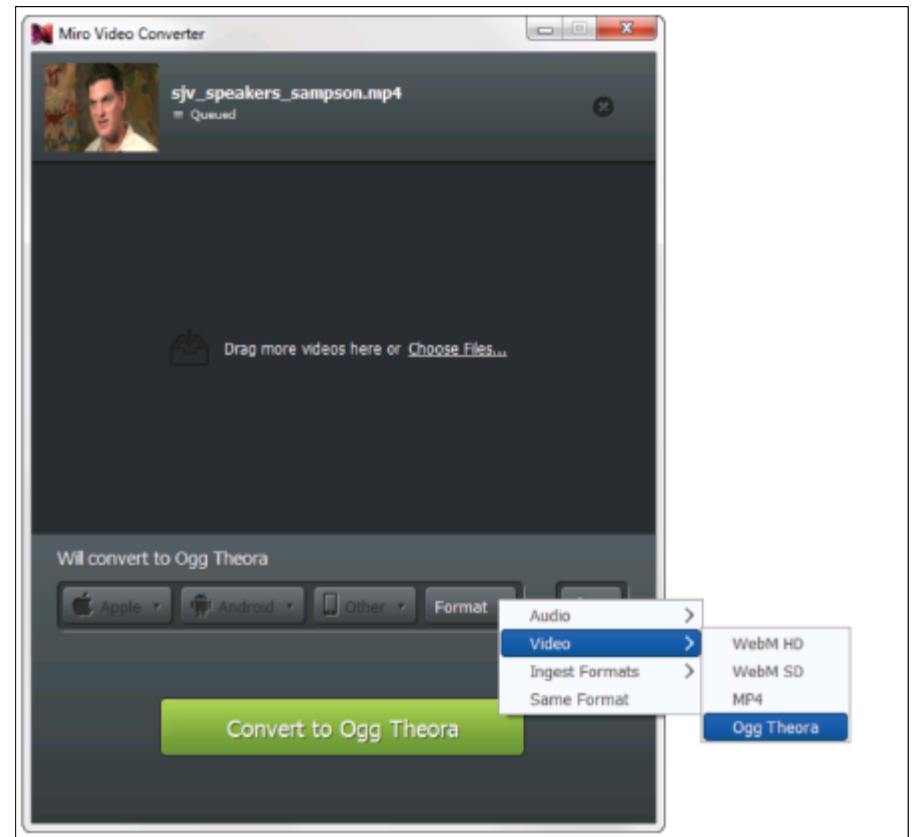
Types MIME

- Les types MIME (Multipurpose Internet Media Extension) permettent d'identifier le contenu d'un fichier audio ou vidéo
 - ▣ Un type MIME est composé d'au moins deux parties : un type et un sous-type et d'un ou plusieurs autres champs au besoin – e.g. `application/javascript`
- Pour s'assurer que votre serveur peut lire les types de fichiers associés, il faut modifier le contenu du fichier `.htaccess` (ou l'équivalent si serveur différent de Apache) et ajouter les types MIME associés

Type média	Type MIME	Codec
MP3	<code>audio/mpeg3</code> , <code>audio/x-mpeg-3</code>	
Ogg Vorbis	<code>audio/ogg</code>	
VP8 / WebM	<code>video/webm</code>	theora, vorbis
Ogg Theora	<code>video/ogg</code>	theora, vorbis
Flash	<code>application/x-shockwave-flash</code>	
AVI	<code>application/x-troff-msvideo</code> , <code>video/avi</code> , <code>video/msvideo</code> , <code>video/x-msvideo</code>	

Encoder des médias

- ❑ Pour encoder des fichiers audio et vidéo dans les formats dont vous avez besoin, vous pouvez utiliser un encodeur
- ❑ Miro Video Converter, iTunes, QuickTime, Windows Media Encoder, Adobe Media Encoder, Firefogg, Ffmpeg ou Handbrake





4.2 LES BALISES HTML TRADITIONNELLES

Éléments HTML historiques

- De par le passé, plusieurs éléments HTML permettaient d'intégrer des plugins dans vos pages Web:

applet

- Pour intégrer des applets Java, déprécié à partir de HTML 4.01

embed

- Introduit avec Netscape 2.0 pour permettre l'intégration d'objets data quelconques, à l'intérieur d'un document Web – fait partie de la spécification HTML5
- Si le plugin nécessaire n'est pas présent, un message d'alerte est affiché, souvent avec une boîte vide à côté

object

- Remplacement des balises applet et embed dans HTML 4.01, se veut plus générique – peut contenir des applets, des fichiers audio et vidéo
- Accepte de nombreux paramètres (à spécifier avec param)
- Fait partie de la spécification HTML5



4.2.1 <OBJECT>

Balise <object>

- Durant des années, ce fut la balise standard pour intégrer des médias dans une page Web
- Un élément <object> a généralement 4 attributs:
 - L'attribut **type** spécifie le type MIME
 - L'attribut **data** précise l'URL du fichier ou du flux à lire
 - Les attributs **width** et **height** spécifient la largeur et respectivement la hauteur soit du type média soit du lecteur média
 - Pour un fichier Flash, ces attributs font référence au fichier. Pour les autres types de médias, ils s'appliquent au lecteur média
 - Pour un fichier qui n'a pas de visuel, ces attributs peuvent être omis, ou définis à 0
 - Par exemple, pour la lecture en arrière-plan d'un fichier audio

Balise <object>

Élément <param>

- Pour chaque élément <object> il est possible de déclarer plusieurs éléments <param> afin de spécifier des valeurs pour le lecteur média
 - Si un lecteur média ne prend pas en charge un certain paramètre, il va simplement l'ignorer
- L'élément <param> dispose de 2 principaux attributs:
 - L'attribut **name** spécifie le nom du paramètre pour un lecteur média en particulier
 - L'attribut **value** permet de préciser la valeur du paramètre nommé précédemment

```
<object type="application/x-shockwave-flash"  
      data="media/coucou.swf"  
      width="400" height="400">  
  <param name="autoplay" value="true">  
</object>
```

Problèmes avec la balise <object>

- Cette balise ne permet pas de préciser le lecteur média à utiliser
 - ▣ On dépend des lecteurs que l'utilisateur a installé sur son poste
 - ▣ Parmi ceux-ci encore faut-il qu'il y en ait un capable de lire le type MIME précisé par la balise <object>
- Pour cette raison, l'utilisation de la balise <object> est sur le déclin
- La plupart du temps, cela représente une solution pour encapsuler les types médias tels que Flash, pour les navigateurs ne prenant pas en charge les balises HTML5 <audio> et <video>



4.2.2 <EMBED>

Balise `<embed>`

- Pas spécifié dans les standards HTML précédents, mais supporté par la plupart des navigateurs aujourd'hui et intégré dans la spécification HTML5
- Un élément `<embed>` a généralement 4 attributs (comme pour `<object>`):
 - L'attribut `type` spécifie le type MIME
 - Si pas de type MIME précisé, le fichier sera ouvert avec le lecteur média par défaut du navigateur, selon l'extension du fichier
 - L'attribut `src` précise l'URL du fichier ou du flux à lire
 - Les attributs `width` et `height` spécifient la largeur et respectivement la hauteur soit du type média soit du lecteur média
- D'autres attributs spécifiques à différents lecteurs médias sont pris en charge

Balise <embed>

- Principale différence avec <object> - tous les paramètres sont codés en tant qu'attributs

```
<embed type="audio/mp3"
       src="media/printemps.mp3"
       width="300" height="25"
       autoplay="true">
```

- Comme pour <object> on ne précise pas le lecteur média à utiliser
 - D'où une utilisation sur le déclin aussi
- Exception: utilisation de <embed> pour lire des vidéos YouTube
 - Possible d'utiliser l'élément <video>, mais l'utilisation de <embed> est bien plus simple (pas besoin de JavaScript)

```
<embed src="https://www.youtube.com/embed/NkfJWJn1"
       width="560" height="350">
```



4.3 MULTIMÉDIA NATIF

- La prise en charge native d'éléments multimédia par les navigateurs offre les avantages suivants:
 - Plus besoin de plugins
 - Gain en rapidité – tout ce qui sera pris en compte nativement par un navigateur sera toujours plus rapide qu'une solution tierce
 - Contrôles natifs fournis par le navigateur
 - Intégration automatique des contrôles clavier

Nouvelles balises HTML5 pour le multimédia

`<audio>`

`<video>`

`<source>`

`<track>`



4.3.1 LA BALISE <AUDIO>

Balise <audio> présentation

- Elle permet d'encapsuler dans une page Web des fichiers ou des flux audio
- Utilisation:

```
<audio src="sayHello.mp3" controls></audio>
```
- La balise <audio> comprend des attributs supplémentaires comme **preload**, **autoplay**, **loop** et **controls**
- Un élément <audio> n'affichera rien sur la page, à moins que les contrôles soient affichés
 - Même si les contrôles ne sont pas affichés à l'écran, il est possible de les implémenter grâce à l'**API Media Elements** – utile lorsque vous souhaitez que votre site ait du son indépendamment des contrôles utilisateur
- Implémente des balises **source** en tant qu'éléments enfants
 - Tout élément enfant qui n'est pas de type **source** sera traité comme une alternative pour les navigateurs ne prenant pas en charge cette balise

Prise en charge de la balise <audio>

Audio element - LS

France 95.48%

Global 89.65% + 0.01% = 89.66%

Method of playing sound on webpages (without requiring a plug-in).

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8		38	29					4.3	
9		43	48	8				4.4	
10		44	49	9		8.4		4.4.4	
11	13	45	50	9.1	36	9.2	8	47	49
	14	46	51	TP	37	9.3			
		47	52		38				
		48	53						

Source: <http://caniuse.com/#feat=audio>

Attributs de la balise <audio>

Attribut	Description
src	Fournit l'adresse de la ressource média (fichier ou flux audio). Cet attribut doit contenir une URL valide vers la source audio.
preload	<p>Utilisé pour indiquer au navigateur comment pré-charger la source audio. Représente seulement une suggestion pour le navigateur, ce dernier fera ce qu'il veut (à savoir respecter les préférences utilisateur).</p> <p>Trois réglages possibles:</p> <ol style="list-style-type: none">1) none: précise au navigateur de ne pas pré-charger la source audio. L'utilisateur n'est pas censé lire l'audio, pas besoin de pré-charger. Le chargement débutera dès que l'utilisateur cliquera sur un bouton "play".2) metadata: précise au navigateur de pré-charger les métadonnées associées à la source audio (e.g. durée).3) auto: c'est le navigateur qui décide s'il télécharge que les métadonnées ou la source en intégralité, ou rien du tout ! C'est le réglage par défaut de l'attribut preload, si rien n'est précisé.

Attributs de la balise <audio>

Attribut	Description
autoplay	Valeur booléenne, <code>true</code> pour démarrer la lecture de l'audio dès que la page est chargée dans le navigateur.
controls	Valeur booléenne, <code>true</code> pour afficher la barre de contrôles par défaut, en dessous du fichier lu.
muted	Valeur booléenne, permettant de couper le son d'un fichier audio. Actuellement très peu pris en charge par les navigateurs.
loop	Valeur booléenne, précisant si le fichier doit être lu en boucle. A utiliser avec précaution, car cela peut devenir une nuisance.
crossorigin	Permettre ou interdire le partage d'un média sur plusieurs domaines, en utilisant la spécification CORS (<u>Cross-Origin Resource Sharing</u>). Valeurs possibles: <code>anonymous</code> (permet l'accès anonyme à la source audio) ou <code>use-credentials</code> (il faut s'authentifier pour lire la source)
mediagroup	Permet de grouper plusieurs médias. Permet la lecture et la synchronisation de 2 sources audio à 2 endroits différents de l'écran

Exemples de contrôlés natifs



Firefox 5



Safari 5.0.5



Chrome 12



Opera 11.50



IE9

Exemples d'utilisation de <audio>

Exemples basiques

- Généralement, la lecture en boucle, la lecture auto et cacher les contrôles sont déconseillés
- Exemple 1: couper le son du fichier lors de son chargement
 - ▣ `<audio src="sayHello.ogg" controls autoplay muted></audio>`
- Exemple 2: empêcher le pré-chargement de la source audio
 - ▣ `<audio src="sayHello.ogg" controls preload="none"></audio>`
- Exemple 3: demander le pré-chargement des métadonnées seules
 - ▣ `<audio src="sayHello.ogg" controls preload="metadata"></audio>`

Exemples d'utilisation de <audio>

Lecture audio avec sources différentes

- Problème: fournir une source audio dans 2 formats à la fois Ogg Vorbis et MP3
- Solution: utiliser l'élément <source> en conjonction avec l'élément <audio>

```
<audio controls autoplay>
  <source src="sayHello.ogg" type="audio/ogg">
  <source src="sayHello.mp3" type="audio/mp3">
  <source src="sayHello.wav" type="audio/wav">
</audio>
```


Exemples d'utilisation de <audio>

Lecture audio avec support arrière

- ❑ Problème: permettre la lecture de la source audio sur des navigateurs anciens (e.g. IE6 à 8) ou ne supportant pas HTML5
- ❑ Solutions:
 - ❑ Soit fournir directement le lien vers la source audio, pour que l'utilisateur la télécharge

```
<audio controls>
  <source src="sayHello.ogg" type="audio/ogg">
  <source src="sayHello.mp3" type="audio/mp3">
  <object type="application/x-shockwave-flash"
    data="player.swf?audioUrl=sayHello.mp3&autoplay=true"
    height="27" width="320">
    <param name="movie"
      value="player.swf?audioUrl=sayHello.mp3&autoplay=true">
  </object>
  <a href="sayHello.mp3">Download the audio file</a>
</audio>
```

Exemples d'utilisation de <audio>

Lecture audio avec support arrière

- Problème: permettre la lecture de la source audio sur des navigateurs anciens (e.g. IE6 à 8) ou ne supportant pas HTML5
- Solutions: http://www.w3schools.com/HTML/html_youtube.asp
 - Soit fournir directement le lien vers la source audio, pour que l'utilisateur la télécharge
 - Soit passer par une encapsulation avec une balise <object>

```
<audio controls>
  <source src="sayHello.ogg" type="audio/ogg">
  <source src="sayHello.mp3" type="audio/mp3">
  <object type="application/x-shockwave-flash"
    data="player.swf?audioUrl=sayHello.mp3&autoplay=true"
    height="27" width="320">
    <param name="movie"
      value="player.swf?audioUrl=sayHello.mp3&autoplay=true">
</object>
</audio>
```



4.3.2 LA BALISE <VIDEO>

Balise `<video>` présentation

- Elle permet d'encapsuler dans une page Web des fichiers ou des flux vidéo
- Utilisation:

```
<video src="media/speaker.mp4" controls></video>
```

- La balise `<video>` comprend des attributs supplémentaires par rapport à la basie `<audio>`, comme `poster`, `width`, et `height`
 - Les autres attributs sont identiques à ceux de la balise `<audio>`

```
<video id="playerVideo" width="480" height="270" controls autoplay>  
  <source src="media/speaker.ogv"  
    type='video/ogg; codec="theora, vorbis"'>  
  <source src="media/speaker.webm"  
    type='video/webm; codec="vp8, vorbis"'>  
</video>
```

Prise en charge de la balise <video>

Video element - LS France 95.48%
Global 89.65%

Method of playing videos on webpages (without requiring a plug-in).

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8		38	29					4.3	
9		43	48	8				4.4	
10		44	49	9		8.4		4.4.4	
11	13	45	50	9.1	36	9.2	8	47	49
	14	46	51	TP	37	9.3			
		47	52		38				
		48	53						

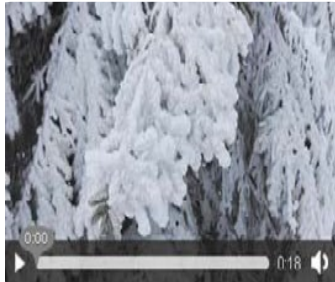
Source: <http://caniuse.com/#feat=video>

Attributs supplémentaires de la balise `<video>`

Attribut	Description
<code>poster</code>	Permet d'afficher une image lorsqu'il n'y a pas de données vidéo disponibles. Si l'attribut <code>poster</code> n'est pas spécifié, le navigateur affichera la 1 ^e image de la vidéo. Doit contenir une URL valide vers l'image.
<code>width</code>	Définit la largeur en pixels de l'élément <code><video></code>
<code>height</code>	Définit la hauteur en pixels de l'élément <code><video></code>

```
<video controls poster="snowy-tree-poster.gif" width="300" height="210">  
  <source src="snowy-tree.mp4" type="video/mp4">  
  <source src="snowy-tree.webm" type="video/webm">  
</video>
```

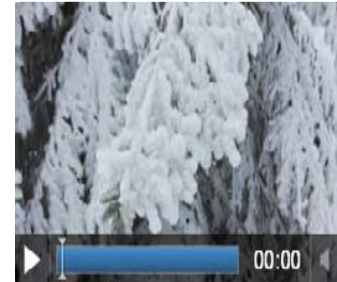
Exemples de contrôlés natifs



Firefox 5

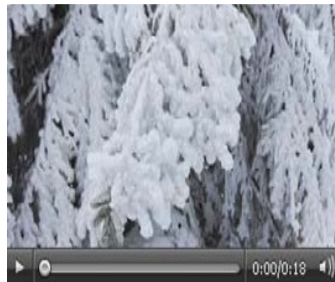


Safari 5.0.5



Chrome 12

```
<video src="snowy-tree.mp4" width="300" height="176" controls></video>
```



Opera 11.50



IE9

Exemples d'utilisation de <video>

Exemples basiques

- Généralement, la lecture en boucle, la lecture auto et cacher les contrôles sont déconseillés
- Exemple 1: couper le son du fichier lors de son chargement
 - ▣ `<video src="match.webm" controls autoplay muted></video>`
- Exemple 2: empêcher le pré-chargement de la source audio
 - ▣ `<video src="match.webm" controls preload="none"></video>`
- Exemple 3: préciser des dimensions pour l'élément <video>
 - ▣ `<video src="snowy-tree.webm" controls width="300" height="210">`

Exemples d'utilisation de <video>

Lecture vidéo avec sources différentes

- ❑ Problème: fournir une source vidéo dans plusieurs formats à la fois (MP4, OGG et WebM)
- ❑ Solution: utiliser l'élément <source> en conjonction avec l'élément <video>

```
<video id="playerVideo" controls autoplay>
  <source src="match.ogv" type="video/ogg">
  <source src="match.mp4" type="video/mp4">
  <source src="match.webm" type="video/webm">
</video>
```

- ❑ L'attribut **type** est déplacé sur l'élément <source> au lieu de <video>
 - ❑ **type** peut aussi contenir les informations relatives au codec utilisé pour encoder le fichier ou le flux vidéo

Exemples d'utilisation de <video>

Lecture audio avec support arrière

- ❑ Problème: permettre la lecture de la source audio sur des navigateurs anciens (e.g. IE6 à 8) ou ne supportant pas HTML5
- ❑ Solutions:
 - ❑ Soit fournir directement le lien vers la source audio, pour que l'utilisateur la télécharge

```
<audio controls>
  <source src="sayHello.ogg" type="audio/ogg">
  <source src="sayHello.mp3" type="audio/mp3">
  <object type="application/x-shockwave-flash"
    data="player.swf?audioUrl=sayHello.mp3&autoplay=true"
    height="27" width="320">
    <param name="movie"
      value="player.swf?audioUrl=sayHello.mp3&autoplay=true">
  </object>
  <a href="sayHello.mp3">Download the audio file</a>
</audio>
```

Exemples d'utilisation de <video>

Lecture vidéo avec support arrière (1)

- Fournir un lien vers le fichier vidéo pour le télécharger

```
<video controls autoplay>
  <source src="snowy-tree.mp4" type="video/mp4">
  <source src="snowy-tree.webm" type="video/webm">
  <figure>
    <a href="snowy-tree.mp4">
      
      <figcaption>Télécharger la vidéo: snowy-tree.mp4</figcaption>
    </a>
  </figure>
</video>
```

- Encapsuler la vidéo dans Flash, en utilisant une balise <object> ou <embed>

```
<video controls autoplay>
  <source src="snowy-tree.mp4" type="video/mp4">
  <object type="application/x-shockwave-flash"
    data="player.swf?videoUrl=snowy-tree.mp4&autoplay=true"
    height="210" width="300">
    <param name="movie"
      value="player.swf?videoUrl=snowy-tree.mp4&autoplay=true">
  </object></video>
```

Exemples d'utilisation de <video>

Lecture vidéo avec support arrière (1)

- C'est mieux d'utiliser une balise <object>, car même si le navigateur ne supporte pas le plugin spécifié par la balise <object>, son contenu sera lu par tous les navigateurs
 - Dans l'exemple précédent, la valeur de la balise <param> sera lue par les navigateurs, même s'ils ne comprennent pas la valeur précisée dans l'attribut type de l'élément <object>
- Il est aussi possible de combiner les 2 solutions vues précédemment

```
<video controls autoplay>
  <source src="snowy-tree.mp4" type="video/mp4">
  <object type="application/x-shockwave-flash"
    data="player.swf?videoUrl=snowy-tree.mp4&autoplay=true"
    height="210" width="300">
    <param name="movie"
      value="player.swf?videoUrl=snowy-tree.mp4&autoplay=true">
  </object>
  <a href="snowy-tree.mp4">Télécharger la vidéo: snowy-tree.mp4</a>
</video>
```



4.3.3 LA BALISE <SOURCE>

Balise `<source>` présentation

- Elle permet de préciser des sources différentes pour un même élément `<audio>` ou `<video>`

- Utilisation:

```
<audio controls>
```

```
  <source src="sayHello.ogg" type="audio/ogg">
```

```
  <source src="sayHello.mp3" type="audio/mp3">
```

```
Désolé, votre navigateur ne supporte pas la balise audio.
```

```
</audio>
```

- Le navigateur lancera la lecture du premier format reconnu, et ignorera les formats non-reconnus
 - Si aucun format n'est reconnu, le navigateur affichera le texte en dessous

Attributs de la balise <source>

Attribut	Description
src	Fournit l'adresse de la ressource média, audio ou vidéo. Cela doit être une URL valide.
type	Spécifie le type de la ressource média, afin d'aider le navigateur à décider s'il peut la lire ou pas. Il est aussi possible de préciser le codec.
media	Spécifie le type de média visé pour la ressource en question (e.g. un certain type de terminal, avec une certaine largeur, etc.). La valeur de cet attribut doit être une requête média valide.

Exemple d'utilisation de `<source>` Avec les requêtes média

- Vidéo pour tous les terminaux ayant une largeur de min 600px
 - `<source src="maVideo.webm" media="all and (min-device-width:300px) ">`
- Ajuster la taille de la vidéo à la taille de l'écran

```
<video controls>
```

```
  <source src="video-small.mp4" type="video/mp4" media="screen and (max-width:600px)">  
  <source src="video-small.webm" type="video/webm" media="screen and (max-width:600px)">  
  <source src="video-medium.webm" type="video/webm" media="screen and (max-width:800px)">  
  <source src="video-medium.mp4" type="video/mp4" media="screen and (max-width:800px)">  
  <source src="video-large.webm" type="video/webm">  
  <source src="video-large.mp4" type="video/mp4">
```

```
</video>
```




4.3.4 LA BALISE `<TRACK>`

Balise <track> présentation

- Elle permet de préciser des éléments textuels synchronisés pour les éléments médias (e.g. sous-titres, descriptions, transcriptions)
- Utilisation:

```
<video src="maVideo.mp4">  
  <track kind="subtitles" src="maVideo-fr.vtt" srclang="fr" label="French">  
  <track kind="subtitles" src="maVideo-de.vtt" srclang="de" label="German">  
</video>
```

Prise en charge de la balise <track>

WebVTT - Web Video Text Tracks - WD

France 91.75%
Global 77.54%

Format for marking up text captions for multimedia resources.

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8		¹ 38	29					4.3	
9		¹ 43	48	8				4.4	
10		¹ 44	49	9		8.4		4.4.4	
11	13	¹ 45	50	9.1	36	9.2	8	47	49
	14	¹ 46	51	TP	37	9.3			
		¹ 47	52		38				
		¹ 48	53						

Source: <http://caniuse.com/#feat=webvtt>

Attributs de la balise <track>

Attribut	Description
kind	<p>Spécifie le type de données fournies par la balise <track>. Si l'attribut est omis, la valeur par défaut est subtitles. Autres valeurs possibles:</p> <ul style="list-style-type: none">▪ subtitles – transcription des dialogues, affichée par-dessus la vidéo▪ captions – transcription complète de l'audio (dialogues, sons, etc.), utile lorsque la piste son n'est pas disponible ou pour les personnes ayant des déficiences auditives▪ descriptions – fournit une description textuelle du composant vidéo de la ressource média, surtout à utiliser lorsque la vidéo n'est pas disponible▪ chapters – spécifie les titres des chapitres, et permet de naviguer à travers le contenu de la ressource média▪ metadata – contient plutôt des données à être utilisées dans des scripts, et non pas à afficher à l'écran
src	<p>URL valide vers les données de la piste texte. Deux formats peuvent être utilisés: WebSRT et le WebVTT</p>

Attributs de la balise <track>

Attribut	Description
srclang	Précise la langue de la piste texte
label	Précise un titre lisible par l'utilisateur pour la piste en question. Doit être unique parmi les différents éléments <track> définis pour une même ressource.
default	Informe le navigateur que l'élément <track> en question doit être utilisé par défaut, si les préférences utilisateur ne spécifient pas autre chose.
ruby	Définit du contenu Ruby (textes courts, affichés à côté du texte principal), souvent utilisé pour les langues asiatiques pour indiquer la prononciation ou pour fournir des annotations. Plus d'informations: https://www.w3.org/TR/css-ruby-1/
[hh:]mm:ss.msmsms	Définit une étiquette temporelle à laquelle le texte devient actif – très similaire à du texte pour karaoke.

Exemples d'utilisation de <track>

- Exemple 1: Fournir 2 fichiers de sous-titres en anglais et en allemand

```
<video controls>
  <source src="video-file.mp4" type="video/mp4">
  <source src="video-file.webm" type="video/webm">
  <track src="en.vtt" kind="subtitles" srclang="en" label="English
subtitles" default>
  <track src="de.vtt" kind="subtitles" srclang="de" label="German
subtitles">
</video>
```

- Exemple 2: Fournir la liste des chapitres

```
<video controls>
  <source src="video-file.mp4" type="video/mp4">
  <track src="en.vtt" kind="subtitles" srclang="en" label="English
subtitles" default>
  <track src="ch-en.vtt" kind="chapters" srclang="en"
label="English chapter listing" default>
</video>
```

Librairies JavaScript utiles pour <track>

LeanBack Player

<http://www.leanbackplayer.com/>

Gratuit, open-source

Supporte les éléments
<video>,<audio>,<track>

Captionator

<http://captionatorjs.com/>

Supporte la balise <track> et l'API
TextTrack

Supporte les sous-titres au formats
*.sub, *.srt, *.sbv, WebVTT

MediaElementJS

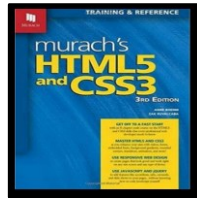
<http://mediaelementjs.com/>

Fournit des players HTML5 pour
l'audio et la vidéo en 100% HTML
et CSS

Encapsule les vidéos au formats
Silverlight et Flash, en simulant
l'API HTML5 MediaElements

RÉFÉRENCES

Livres:



Murach's HTML5 and CSS3

Anne Boehm, Zak Ruvalcaba
Mike Murach & Associates, 2015 - 703 pages
ISBN 978-1-890114-83-7

<https://www.murach.com/shop/murachs-html5-and-css3-3rd-edition-detail>

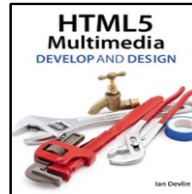
HTML5 Multimedia: Develop and Design

Ian Devlin

Peachpit Press, 2015 – 290 pages

ISBN: 978-0-321-79393-5

<http://html5multimedia.com/>



Liens Internet:

<http://www.dynamicdrive.com/>

<http://www.javascriptsource.com/>

<http://caniuse.com/>

<http://desandro.github.io/3dtransforms/>

<http://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>

<http://www.alsacreations.com/article/lire/1418-css3-transformations-2d.html>

http://www.w3schools.com/HTML/html5_audio.asp

http://www.w3schools.com/HTML/html5_video.asp