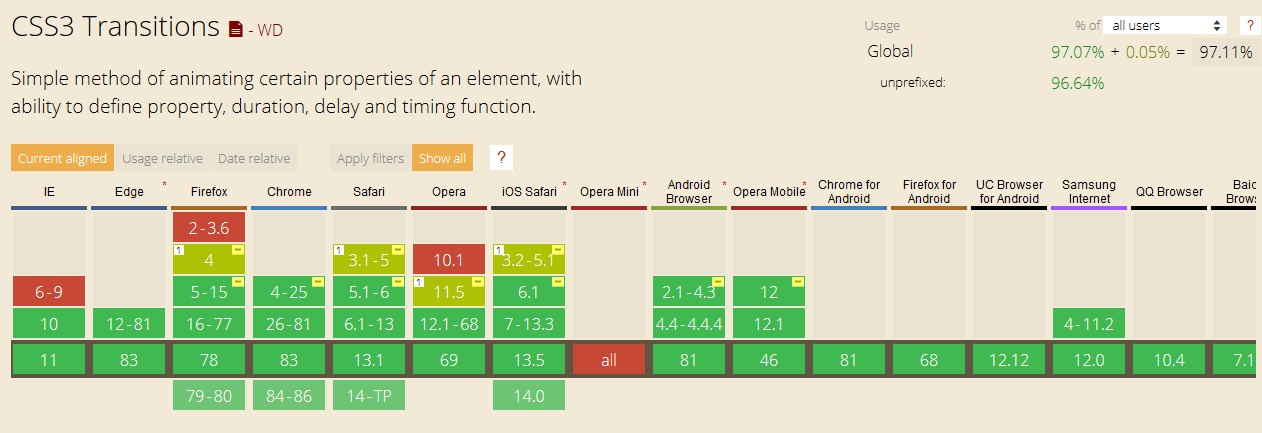
**CSS3 - transition**

* Support par les navigateurs
* Principe des transitions
* Propriétés CSS3 associées
* Ordre et sens des transitions
* Propriété raccourcie
* Transition pilotée par JavaScript
* class="gauche"
* class="droite"

# Support par les navigateurs



Les propriétés CSS3 liées aux transitions sont supportées depuis de nombreuses versions <https://caniuse.com/#search=css3%20transition>

# Principe des transitions

* Une **transition** permet de **passer** d’un **état de départ** à un **état d’arrivée** en un certain temps
* Par exemple, le passage de la souris ( **:hover** )
* On définit les propriétés CSS d’un élément et au survol, on peut **changer certaines propriétés**
* Au lieu que le **changement** soit abrupt, il sera **progressif** en jouant sur plusieurs paramètres

**Exemple**



# Propriétés des transitions

* Il y a principalement quatre propriétés liées aux transitions :
  + **transition-duration**
  + **transition-delay**
  + **transition-property**
  + **transition-timing-function**
* Et enfin une propriété raccourcie qui regroupe les précédentes : **transition**

(dans l’ordre :transition-property transition-duration transitiontiming-function transition-delay séparés par un espace)

# transition-duration

* Cette propriété permet de définir la **durée** de la transition d’un état à l’autre
* On donne une valeur exprimée en **secondes** (ex : 2s) ou en **millisecondes** (ex : 500ms) pour définir la durée.

La valeur **par défaut** étant de **0s** ● **Par exemple :**

On ajoute dans #element {...} transition-duration: 2s;

# transition-delay

* Cette propriété permet de définir le **délai** à attendre **avant que la transition** d’un état à l’autre ne **démarre** réellement
* On donne une valeur exprimée en **secondes** (ex : 2s) ou en **millisecondes** (ex : 500ms) pour définir le délai.

La valeur **par défaut** étant de **0s** ● **Par exemple :**

On ajoute dans #element {...} transition-delay: 500ms;

# transition-property

* Pendant une transition, on peut **modifier zéro, une ou plusieurs propriétés** en même temps
* Cette propriété permet de définir **quelle(s)** **propriété(s) change(nt)** pendant latransition
* On donne la liste des propriétés (séparées par une virgule) qui changent pendant la transition. La valeur **par défaut** étant **all** ● **Par exemple :**

On ajoute dans #element {...} transition-property: border-radius, background-color;

# Modifier plusieurs propriétés

* Supposons que dans #element:hover {…}, plusieurs propriétés ont été modifiées
* Celles qui sont renseignées dans property-transition changeront progressivement pendant la transition, les autres le feront instantanément (exemple 1)
* On peut paramétrer différemment les propriétés impliquées dans la transition (exemple 2)

# Exemple 1

* 2 propriétés sont modifiées pendant le passage d’un état à l’autre :
  + border-radius
  + background-color
* Seulement une va changer progressivement pendant la transition :
  + border-radius
  + durée : 2 secondes et délai : 500 millisecondes

# Exemple 2

● dans #element:hover, trois propriétés vont être modifiées

* **border-radius** démarre avec un délai de 0s et dure 2s
* **background-color** démarre après 2s et dure 1s
* **border-color** change instantanément

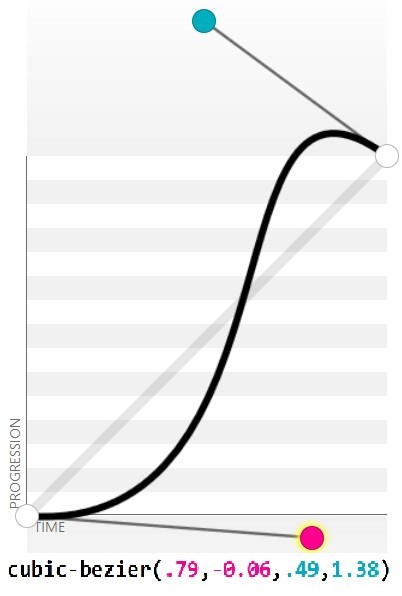
# transition-timing-function

* Cette propriété permet de définir une **courbe de vitesse de progression** pendant latransition

(sous forme d’une courbe de Bézier) construction mathématique pour définir la forme d’une courbe. On définit l’orientation du point de départ et du point d’arrivée.

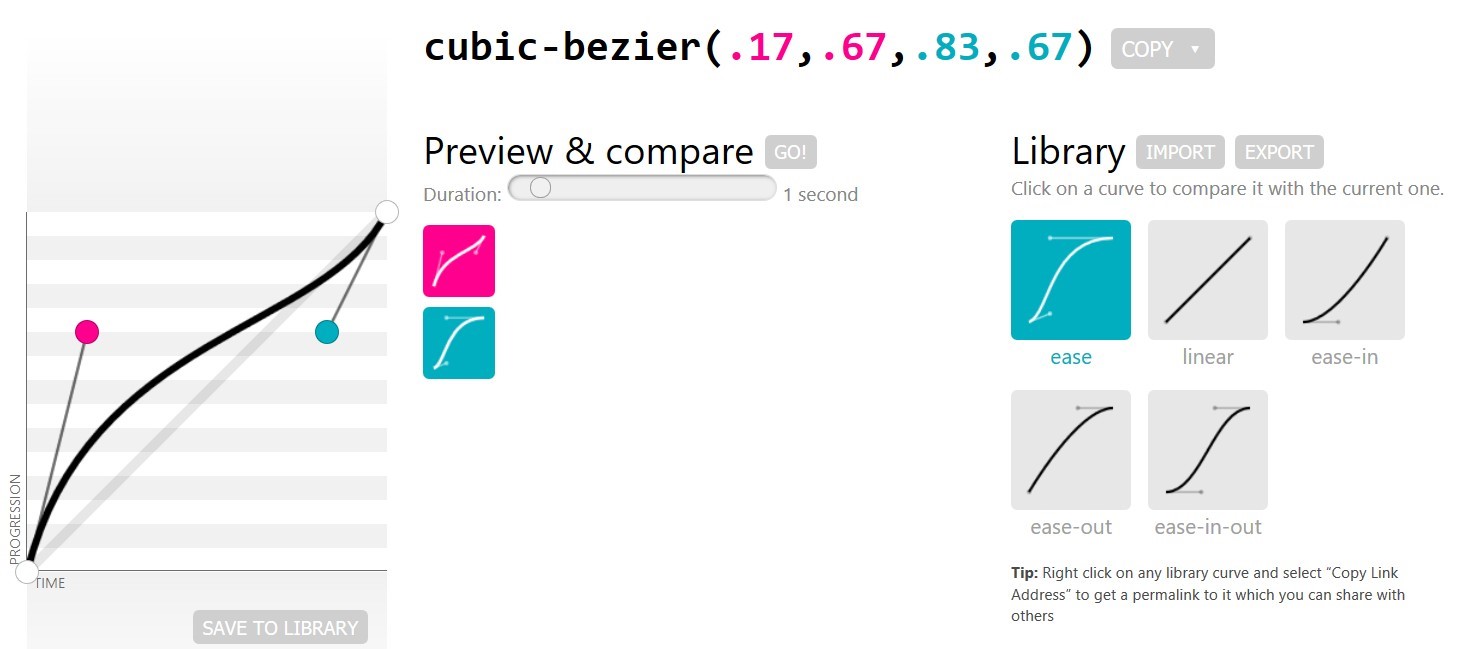
* Les valeurs possibles sont : ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier(n,n,n,n) La valeur **par défaut** étant **ease**
* Cela donne la possibilité de régler la vitesse, l’accélération et le ralentissement de la transition

# transition-timing-function



# Courbe de Bézier

Pour définir les paramètres de la courbe de Bézier [https://cubic-bezier.com](https://cubic-bezier.com/)



# Sens des transitions

* Les transitions ne **doivent pas être** forcément **symétriques** (de l’état A vers B ou de B vers A)
* Si on définit dans un sens, la propriété **transition-property:none;** alors le changement est instantané car il n’y a pas de transition

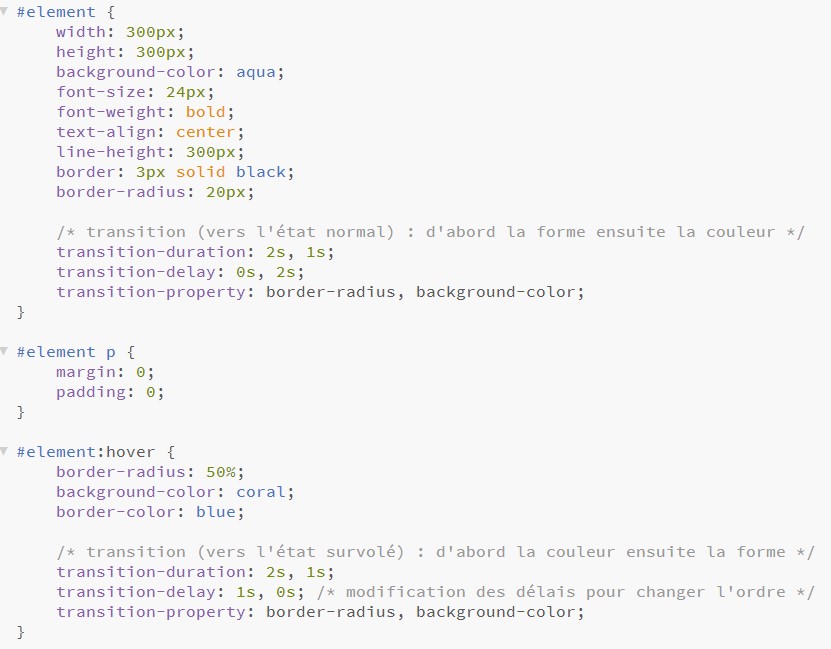
(exemple 3)

* On peut aussi définir des **transitions différentes** en modifiant les paramètres **dans un sens ou dans l’autre** (exemple 4)

# Exemple 3



# Exemple 4



# transition

* Il y a une *propriété raccourcie* **transition** qui regroupe les 4 propriétés séparées, par exemple:

transition-duration: *2s*, 1s;

transition-property: *border-radius*, background-color; transition-timing-function: *ease*, linear; transition-delay: *0s*, 2s;

* Cela devient **transition: *border-radius 2s ease 0s*, backgroundcolor 1s linear 2s ;**

Dans l’ordre: [ property duration timing-function delay ]

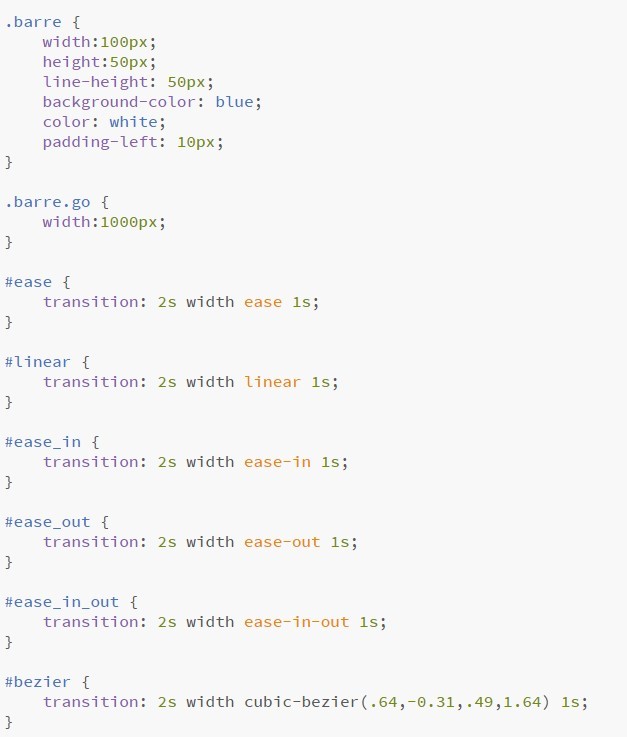
# Transitions et JavaScript

* Pour provoquer une transition, il faut :
  + un état initial
  + un état final
  + un déclencheur pour passer de l’un à l’autre
* Si on ne veut pas réagir au survol avec la souris ( :hover ), on utilise JavaScript pour déclencher sur un autre événement (Exemple 5)

# Exemple 5 - HTML



# Exemple 5 - CSS



# Exemple 5 - JS

