

Animations / Transitions / GSAP

Table des matières

1. CSS.....	2
a. Transition.....	2
b. Animation.....	2
Exemple	4
2. jQuery	4
a. Fonction jQuery CSS	4
b. Fonction jQuery Animate	5
Animation une fois	5
Replay	5
Transformation	5
c. Ajout de Classe CSS	5
Animation une fois	6
Animation aller retour	6
Replay	6
3. Steps	6
4. GSAP	8
a. Installation	8
b. Animation simple	9
c. Sélection d'éléments ...JavaScript ou jQuery.....	10
d. to, from, fromTo	10
to.....	10
from.....	10
fromTo.....	11
Propriétés pouvant être animées	12
Easing	13
Callback.....	14
Delay.....	14
Repeat	14
Stagger.....	14
Contrôler l'animation	16
e. Plusieurs animations simultanées	16
f. Timeline.....	17

1. CSS

a. Transition

Transition entre 2 états suite à un déclenchement.

- **transition-property** : propriété sur laquelle s'applique la transition (exemple : « all » ou « opacity » ou encore « transform »)
- **transition-duration** : durée de la transition (exemple « 2s » ou « 400ms »)
- **transition-delay** : délai avant que la transition démarre
- **transition-timing-function** : linear | ease | ease-in | ease-out | ease-in-out | step-start | step-end | steps(int,start | end) | cubic-bezier(n,n,n,n) | initial | inherit

« Shorthand »

- « transition: property duration timing-function delay | initial | inherit »

Exemple au passage de la souris l'élément double de taille

```
. item{
  width: 250px;
  height: 250px;
  background: #D81B60;

  /* transition*/
  transition: transform 2s;
}

/* etat 2*/
.item:hover{
  transform: scale(2);
}
```

```
<div class="item"></div>
```

b. Animation

Propriétés d'animation :

- **animation-name** : correspond au nom de l'animation @keyframes
- **animation-duration** : durée totale de l'animation
- **animation-delay** : délai avant que l'animation démarre
- **animation-timing-function** : linear | ease | ease-in | ease-out | ease-in-out | step-start | step-end | steps(int,start | end) | cubic-bezier(n,n,n,n) | initial | inherit
- **animation-fill-mode** : none | forwards | backwards | both | initial | inherit
- **animation-iteration-count**: nombre | infinite | initial | inherit
- **animation-direction** : normal | reverse | alternate | alternate-reverse | initial | inherit
- **animation-play-state** : paused | running | initial | inherit

« Shorthand »

- **animation**: « name duration timing-function delay iteration-count direction fill-mode play-state »

Keyframes

Peut être défini avec « from » et « to ». « from » est facultatif si les propriétés de départ sont les mêmes que l'élément ciblé

```
@keyframes my-animation {
  from{

  }

  to{

  }
}
```

Peut également avoir plus

```
@keyframes my-animation {
  0{

  }
  20%{

  }
  40%{

  }
  60%{

  }
  80%{

  }
  100%{

  }
}
```

Les propriétés à 100% doivent être les mêmes qu'à 0 pour animation jouée en boucle (inifnite)

Préfixes pour les différents navigateurs

```
@-o-keyframes my-animation {

}
@keyframes my-animation {

}
@-webkit-keyframes my-animation {

}
```

De même pour les animations

```
-o-nimation :my-animation 2s ;
-webkit-animation :my-animation 2s ;
animation :my-animation 2s ;
```

Exemple

```
.item{
  width: 250px;
  height: 250px;
  background: #D81B60;
  transform: scale(1);

  animation: scale-x2-animation 2s forwards;
}

@keyframes scale-x2-animation {
  from{
    transform: scale(1);
  }

  to{
    transform: scale(2);
  }
}
```

from ici est dispensable

```
<div class="item center"></div>
```

2. jQuery

a. Fonction jQuery CSS

Pour animer il faut définir une transition dans les CSS

```
.item{
  width: 250px;
  height: 250px;
  background: #D81B60;
  transform: scale(1);

  /* transition*/
  transition: transform 2s;
}
```

Ne joue qu'une fois

```
$('.play').on('click', function(){

  $('.item').css({
    'transform': 'scale(2)'
  });

});
```

b. Fonction jQuery Animate

On n'a pas besoin de définir de transition ni d'animation dans les css. On modifie l'élément et on définit directement la durée avec « animate »

Animation une fois

Propriété simple. On passe un élément ayant pour opacité 0 au départ à 1

```
$('.item').animate({'opacity' : 1},2000);
```

Replay

```
$('.play').on('click', function () {
    $({ p: 0 }).animate({ p: 10 }, {
        duration: 2000,
        step: function (now) {
            $(".item").css('opacity', now/10);
        }
    });
});
```

Transformation

```
$('.play').on('click', function () {
    $({ p: 1 }).animate({ p: 2 },
        {
            duration: 2000,
            step: function (now) {
                $(".item").css('transform', 'scale(' + now + ')');
            }
        }
    );
});
```

c. Ajout de Classe CSS

CSS

```
.item{
    width: 250px;
    height: 250px;
    transform: scale(1);
    background: #D81B60;
}
.scale-x2-animation{
    animation: scale-x2 2s forwards;
}
@keyframes scale-x2 {
    from{
        transform: scale(1);
    }
    to{
        transform: scale(2);
    }
}
```

HTML

```
<button class="play">play!</button>
<div class="item center"></div>
```

Animation une fois

```
$(".item").addClass('scale-x2-animation');
```

Animation aller retour

```
$('.play').on('click', function () {
    $(".item").toggleClass('scale-x2-animation');
});
```

Replay

```
$('.play').on('click', function () {
    var el = $('.item');

    el.before( el.clone(true) ).remove();
    $(".item").addClass('scale-x2-animation');
});
```

3. Steps

Permet de spécifier le nombre d'étapes pour jouer l'animation complète. Plus il y en a plus l'animation est fluide.

Exemple avec une barre de chargement, au clic sur le bouton play la barre se charge en 5 étapes

CSS

```
.item{
    background:#c0392b;
    height:10px;
    width:0;

    transition: width 2s steps(5);
}
.show{
    width:100%;
}
```

HTML

```
<div class="item"></div>
<button class="play">Play!</button>
```

jQuery

```
$('.play').on('click', function(){
    $(".item").addClass('show');
});
```

Si on veut pouvoir replay utiliser une animation

CSS

```
.item{
    background:#c0392b;
    height:10px;
    width:0;
}
.progress-animation{
    animation: progress 2s steps(5) forwards;
}

@keyframes progress{
    to{
        width: 100%;
    }
}
```

jQuery

```
$('.play').on('click', function(){
    var el = $(' .item' );
    el.before( el.clone(true) ).remove();
    $(".item").addClass('progress-animation');
});
```

Il est possible de personnaliser le nombre de steps avec jQuery

```
.item{
    background:#c0392b;
    height:10px;
    width:0;
}
@keyframes progress{
    to{
        width: 100%;
    }
}
```

jQuery

```
$('.play').on('click', function(){

    var steps = Math.floor((Math.random() * 10) + 5);

    var el = $(' .item' );
    el.before( el.clone(true) ).remove();
    $(".item").css({'animation': 'progress 2s steps(' + steps + ') forwards' })

});
```

4. GSAP

[Site](#), [learning](#), [Codepen](#)

a. Installation

Télécharger soit le ZIP, soit récupérer le lien CDN

Get GSAP
Version: 1.18.4 updated 2016-04-26

[Download zip](#) [github](#)

CDN Link
Copy/paste this code into your HTML:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js">
```

Robust Lightweight Customize ▾ Version: 1.18.4 ▾

By using GreenSock code, you agree to the [terms of use](#).

Il est possible de récupérer 3 « packages » : robuste (contient tout), « lightweight » (contenant le « minimum ») ou sélectionner les éléments à inclure.

Get GSAP
Version: 1.18.4 updated 2016-04-26

[Download zip](#) [github](#)

CDN Link
Copy/paste this code into your HTML:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenLite.min.js">
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/plugins/CSSPlugin">
```

Robust Lightweight Customize ▾ Version: 1.18.4 ▾

Core	Plugins	
<input checked="" type="checkbox"/> TweenLite (9kb)	<input checked="" type="checkbox"/> css (15kb)	<input type="checkbox"/> scrollTo (1kb)
<input type="checkbox"/> TweenMax (34kb)	<input type="checkbox"/> bezier (4kb)	<input type="checkbox"/> colorProps (1kb)
<input type="checkbox"/> TimelineLite (4kb)	<input type="checkbox"/> directionalRotation (1kb)	<input type="checkbox"/> cssRule (1kb)
<input type="checkbox"/> TimelineMax (6kb)	<input type="checkbox"/> attr (1kb)	<input type="checkbox"/> easel (3kb)
	<input type="checkbox"/> text (1kb)	<input type="checkbox"/> roundProps (1kb)
Extras		
<input type="checkbox"/> EasePack (2kb)		
<input type="checkbox"/> jquery.gsap.js (2kb)		
<input type="checkbox"/> Draggable (12kb)		

By using GreenSock code, you agree to the [terms of use](#).

b. Animation simple

Avec le package « Robuste »

```
<body>
<div class="item" style="width:0; height:0;background:red;"></div>
<!-- ROBUSTE -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script>
  function animation_1() {
    var item = document.querySelector('.item');
    TweenMax.to(item, 0.5, {width: "100px", height: "100px"});
  }
  window.onload = function () {
    animation_1();
  };
</script>
</body>
```

Avec le package « Light » on a pas accès à « TweenMax », on utilisera alors « TweenLite »

```
<body>
<div class="item" style="width:0; height:0;background:red;"></div>
<!-- LIGHT -->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/plugins/CSSPlugin.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/easing/EasePack.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenLite.min.js"></script>
<script>
  function animation_1() {
    var item = document.querySelector('.item');
    TweenLite.to(item, 0.5, {width: "100px", height: "100px"});
  }
  window.onload = function () {
    animation_1();
  };
</script>
</body>
```

c. Sélection d'éléments ...JavaScript ou jQuery

On peut sélectionner :

- Un élément avec **JavaScript**

```
var item = document.querySelector('.item');
TweenMax.to(item, 0.5, {width: "100px", height: "100px"});
```

- Par l'id

```
TweenMax.to('#myitem', 0.5, {width: "100px", height: "100px"});
```

- Plusieurs éléments

```
var items = document.querySelectorAll('.item');
TweenMax.to(items, 0.5, {width: "100px", height: "100px"});
```

- Par la classe

```
TweenMax.to('.item', 0.5, {width: "100px", height: "100px"});
```

- Tableau d'éléments

```
TweenMax.to(['.item-1', '.item-2'], 0.5, {width: "100px", height: "100px"});
```

GSAP supporte **jQuery** également. Bien entendu il faut **référencer** jQuery auparavant qui n'est pas inclus.

```
var item = $('#item');
TweenMax.to(item, 0.5, {width: "100px", height: "100px"});
```

d. to, from, fromTo

- **target** : le ou les éléments ciblés
- **duration** : durée de l'animation en secondes
- **vars** : propriétés à modifier

to

```
TweenLite.to(target, duration, {vars});
TweenMax.to(target, duration, {vars});
```

from

... permet de définir le départ l'animation

```
TweenLite.from(target, duration, {vars});
TweenMax.from(target, duration, {vars});
```

Exemple

La boîte commencera avec une taille de 100px par 100px et finira à une taille de 0

```
<body>
<div class="item" style="width:0; height:0;background:red;"></div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js">
</script>
<script>

    function animation_1() {
        TweenMax.from('.item', 2, {width: "100px", height: "100px"});
    }

    window.onload = function () {
        animation_1();
    };

</script>
```

fromTo

```
TweenLite.fromTo(target, duration, {from},{to}) ;
TweenMax.fromTo(target, duration,{from}, {to}) ;
```

Exemple

```
<body>
<div class="item" style="background:red;"></div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script>

    function animation_1() {
        TweenMax.fromTo('.item', 2, {width: 100, height: 100},{width:0,height:0});
    }

    window.onload = function () {
        animation_1();
    };

</script>
</body>
```

Propriétés au départ de
l'animation

Propriétés en fin
d'animation

Propriétés pouvant être animées

Note : par défaut si aucune unité « px » est appliqué en général si aucune unité n'est indiquée. Exemple ici cela équivaut à 100px.

```
{width: 100, height: 100},
```

On retrouve toutes les propriétés CSS. Les propriétés avec un tiret «-» (exemple « font-size » ou background-color ») le perdent et gagnent une majuscule (exemple « fontSize », « backgroundColor »)

Pour les **transformations** on retrouve :

- rotation (+ rotationX et rotationY pour la 3D)
- scale
- pour translate on indiquera plutôt x ou y

Exemple rotation de 180 degrés et scale

```
<body>
<div class="item" style="width:100px; height:100px;background:red;"></div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script>
  function animation_1() {
    TweenMax.to('.item', 2, {rotation:180, scale:0.5});
  }

  window.onload = function () {
    animation_1();
  };
</script>
```

Exemple déplacement de 500px vers la droite et 200px vers le bas. Autrement utiliser - pour un déplacement vers le haut par exemple « y :-100 ».

```
<body>
<div class="item" style="width:100px; height:100px;background:red;"></div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script>
  function animation_1() {
    TweenMax.to('.item', 2, {x:500, y:200});
  }

  window.onload = function () {
    animation_1();
  };
</script>
```

Il est possible également d'utiliser les notations avec « -= » « += », exemple « x :+=100 »

```
TweenMax.to('.item', 2, {x:"+=500px"});
```

Ou en %

```
TweenMax.to('.item', 2, {x:"+=100%"});
```

On peut également indiquer left, top, bottom, right ... pour un élément en position « relative » ou « absolute ».

```
TweenMax.to('.item', 2, {left:"+=100%"});
```

Opacity et autoAlpha

autoAlpha améliore les performances d'affichage.

```
TweenMax.to('.item', 2, {autoAlpha:0});
```

Easing

[Utilitaire](#)

La fonction de easing est indiquée avec les propriétés à animer.

GreenSock Ease Visualizer

Power0
Power1
Power2
Power3
Power4
Back
Elastic
Bounce
Rough
SlowMo
Stepped
Circ
Expo
Sine

Type: easeOut

RUN

```
// click and modify the underlined values
TweenLite.to(graph, 2.5, { ease: Power2.easeOut, y: 0 });
```

Exemple

```
TweenMax.to('.item', 2, {x:500, ease:Bounce.easeOut });
```

Callback

- onUpdate
- onComplete : Fonction appelée une fois que l'animation est terminée.
- onUpdateParams, onCompleteParams : paramètres à passer

```
<body>
<div class="progress-bar" style="height:20px;background:red;"></div>
<span class="progress-label"></span>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script>
    var progressLabel = document.querySelector('.progress-label');

    function update() {

    }

    function complete(params) {
        progressLabel.innerHTML = params;
    }

    function animation_1() {
        TweenMax.fromTo('.progress-bar', 4, {x:"-100%"},onUpdate:update,
onComplete:complete, onCompleteParams:['Terminé!'] },{x:"0%"});
    }

    window.onload = function () {
        animation_1();
    };
</script>
</body>
```

Delay

Délai avant que l'animation commence.

```
TweenMax.to('.item', 2, {width: "100px", height: "100px", delay:1});
```

Repeat

Exemple animation répétée une fois après un délai d'une seconde

```
TweenMax.to('.item', 2, {scale:2, repeat:1, repeatDelay:1});
```

Stagger

CSS

Pour l'exemple on met en forme les éléments

```
.list > li {
    position: relative;
    display: inline-block;
    background: #c0392b;
    width: 120px;
    height: 120px;
    margin: 10px;
```

```

transform-origin: center center;
transform: scale(0);
}

```

HTML

```

<ul class="list">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>

```

JS

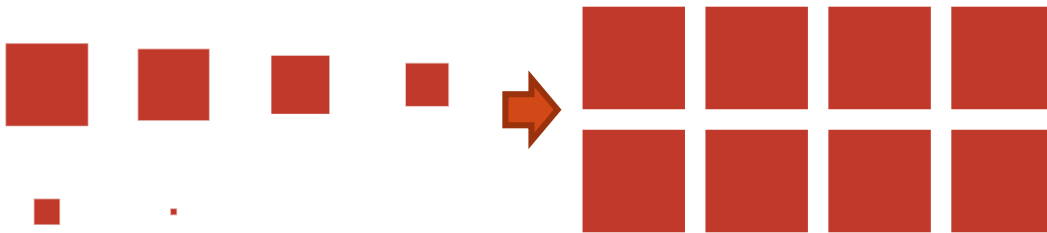
```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script>
function animation_1() {
  var items = document.querySelectorAll('.list > li')
  TweenMax.staggerTo(items, 2, {scale:1},0.2);
}

window.onload = function () {
  animation_1();
};
</script>

```

On indique le délai en dernier paramètre de staggerTo



Contrôler l'animation

Méthodes : play, pause, reverse, etc.

```

<button class="play">Play</button>
<button class="pause">Pause</button>
<button class="reverse">Reverse</button>

<div class="item" style="width:0; height:0;background:red;"></div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script>
    window.onload = function () {
        var item = document.querySelector('.item');
        var timeline = TweenMax.to(item, 0.5, {width: "100px", height: "100px"});
        timeline.pause();

        document.querySelector('.play').addEventListener('click', function () {
            timeline.play();
        });
        document.querySelector('.pause').addEventListener('click', function () {
            timeline.pause();
        });
        document.querySelector('.reverse').addEventListener('click', function () {
            timeline.reverse();
        });
    };
</script>

```



e. Plusieurs animations simultanées

```

<body>
<div class="red-box" style="width:100px;
height:100px;background:red;"></div>
<div class="orange-box" style="width:50px;
height:50px;background:orange;"></div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js">
</script>
<script>

    function animation_1() {
        TweenMax.to('.orange-box', 2, {scale:2});
        TweenMax.to('.red-box', 2, {backgroundColor:'blue'});
    }

    window.onload = function () {
        animation_1();
    };

</script>

```


f. Timeline

Comme pour TweenLite et TweenMax on a **TimelineLite** et **TimelineMax**, accessible selon le package utilisé (light ou robuste).

Chaque étape de l'animation s'enchaîne, cela peut être pour le même élément ou un autre élément.

```
<body>
<div class="item-1" style="width:0; height:0;background:red;"></div>
<div class="item-2" style="width:0; height:0;background:red;"></div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js">
</script>
<script>

    function animation_1() {
        var item1 = document.querySelector('.item-1'),
            item2 = document.querySelector('.item-2');
        var timeline = new TimelineLite();

        timeline.to(item1, 0.5, {width: "100px", height: "100px"})
                .to(item1, 1, {backgroundColor: 'blue'})
                .to(item1, 1, {rotation: 360})
                .to(item2, 1, {width: "100px", height: "100px"})
    }

    window.onload = function () {
        animation_1();
    };

</script>
</body>
```

g. SplitText

Animation de texte

```
<body>

<p class="title">Mon texte animé</p>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.4/TweenMax.min.js"></script>
<script src="js/SplitText.min.js"></script>
<script>

    function animation_1() {
        var timeline = new TimelineLite,
            splitText = new SplitText(".title", {type: "words,chars"}),
            chars = splitText.chars;

        TweenLite.set(".title", {perspective: 400});
        timeline.staggerFrom(chars, 0.8, {opacity: 0, scale: 0, ease: Back.easeOut},
0.01, "+=0");
    }

    window.onload = function () {
        animation_1();
    };

</script>
</body>
```