

► MÉMO • GSAP 3

◁ La base

On ajoute la librairie dans le document après les éléments de la page (h1,div,p,img...) ET avant la fermeture du body (</body>) :

```
<script src="(1)"></script><!-- GSAP -->
<script>
(2) // mon code ou un test
</script>
ou
<script src="mon-fichier.js"></script>
```

(1) `gsap.min.js` (si vous avez téléchargé la librairie)
ou <https://cdnjs.cloudflare.com/ajax/libs/gsap/3.0.4/gsap.min.js> (version en ligne)
(2) `alert(gsap.utils.random(1,6,1));` /* Simule un dé */
ou mettre cette ligne dans `mon-fichier.js`

<https://greensock.com/docs/v3/Installation>

► Animation par interpolation

L'interpolation (*tween*) calcule les étapes intermédiaires entre deux valeurs d'une propriété.

Exemple pour déplacer un ou plusieurs éléments de 100px en x et y en 1s.

```
gsap.to("sélecteur comme en CSS", {
  x : 100,
  y : 100,
  duration : 1
}); // déplacement en diagonal
```

Exemple pour déplacer un ou plusieurs éléments de 100px en x et y en 1s **vers** leur position initiale.

```
gsap.from("sélecteur comme en CSS", {
  x : 100,
  y : 100,
  duration : 1
});
```

Exemple pour déplacer **de** -100px **vers** 100px.

```
gsap.fromTo("sélecteur comme en CSS", {
  x : -100
},{
  x : 100,
  duration : 1
});
```

🧩 il existe aussi `gsap.set()` pour modifier sans animer

☰ Propriétés, valeurs

Presque toutes les propriétés CSS et SVG sont animables, les noms sont à écrire en *camelCase* façon JavaScript et non CSS, voici une liste :

```
/* transformations CSS3 (= performance) */
x : 100, // en px
y : 100, // en px
rotation : 180, // en degré
scale : 0.75,
opacity : 0.5,
transformOrigin : "center center",

/* CSS classique */
left : 15,
top : 100,
backgroundColor : "#ff6600",
...

/* pour le SVG */
fill : "#ff6600",
strokeWidth : 5,
...

/* pour les attributs (plugin inclus)*/*
attr: {cx:100, cy: 50},

/* pour GSAP */
duration : 1,
ease : "power4.in", // ! sans espace (ease ☞)
repeat : -1,
yoyo : true,
stagger : 0.2,
delay : 0.2
...

```

☞ Animation par timeline

La **timeline** permet d'enchaîner, contrôler ou synchroniser plusieurs animations. La timeline est généralement stockée dans une variable avant d'être manipulée.

```
var tl = gsap.timeline({repeat : 3, yoyo : true });
```

```
tl.to("sélecteur comme en CSS", {
  x : 100,
  duration : 1
});
```

```
tl.to("sélecteur comme en CSS", {
  y : 100,
  duration : 2
});
```

```
// enchaîne avec y : 100 (puis lecture 3 x en yoyo de tl)
```

⊗ Ease, valeurs

Le ease joue sur l'accélération, l'amortissement... d'un mouvement en imitant ou exagérant les lois de la physique.

```
/* linéaire */
"none"
"steps(12)"

/* in inOut out */
"power1.in", "power1.inOut", "power1.out"
"power2.in", .....
"power3.in", .....
"power4.in", .....
"circ.in", .....
"expo.in", .....
"sine.in", .....
"bounce.in", .....
"back.in(1)", "back.inOut(1)", "back.out(1)"
"elastic.in(1,0.3)" .....

```

🔗 Function()

Une valeur peut être calculée par une fonction.

```
gsap.to("sélecteur comme en CSS", {
  x : function(){
    return gsap.utils.random(-100, 300)
  },
  duration : 1
});
```

// plus complexe

```
gsap.to(".item", {
  x: function(index, target, targets) {
    return index * 50;
  }
});
```

⚠ Erreurs fréquentes

En cas d'erreur ou d'animation non exécutée :

- une erreur dans le code bloque le script, ouvrir la console et regarder les messages :

gsap undefined : gsap ne se charge pas (voir **La base**)
attention aux {}, {}, []
attention aux , ; =
attention aux '...' et '...'
fautes de frappe (0,0)
beautify ton code !

- pas d'erreur en console ?

mauvais fichier, mauvais sélecteur, rafraîchir la page !