

Transitions CSS

Explications sur la transition

Explications

La "transition" est une propriété CSS.

Normalement, quand on utilise un "hover", le style change brutalement. Grâce à la propriété "transition", on peut faire un changement en douceur !

Il suffit de préciser les propriétés CSS de départ et d'arrivée, et le navigateur se chargera de calculer l'entre-deux tout seul.

Donc, si on combine le hover et la transition, on peut créer des animations CSS super facilement !

Exemple pratique

```
#texteCache
{
  color: transparent;
  transition: 2s;
}
```

Par exemple, sur cet exemple, le texte de mon paragraphe est transparent et j'ai une transition de 2 secondes.

Et en utilisant le même ID avec ":hover" derrière, je peux coder le comportement que va avoir le paragraphe quand je passe ma souris dessus.

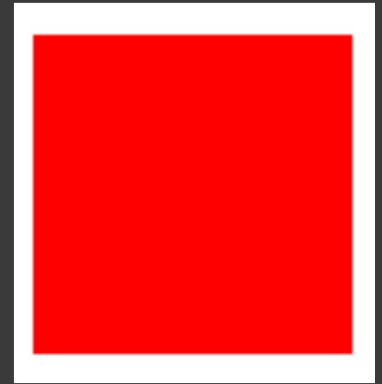
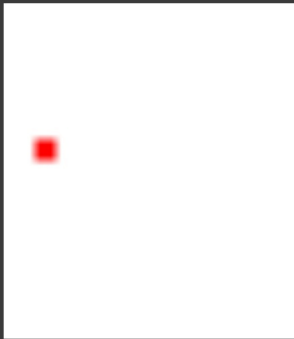
Comme il y a une transition de 2s dans le CSS de mon paragraphe, il créera une animation rendant le texte transparent noir !

```
#texteCache:hover
{
  color: black;
}
```


Transitions CSS

Element caché qui grandit

Le rendu



Le HTML

Pour cette animation, il faut créer un élément HTML (comme une `<div>` ou une `<img>` par exemple), qu'on fera ensuite grandir avec la transition.

```
<div id="toutPetit"></div>
```

Le CSS

```
#toutPetit
{
  width: 2px;
  height: 2px;
  background-color: red;

  transition: 2s;
}
```

Pour le hover, on va tout simplement donner une taille normale à notre élément, et la transition s'occupera de le faire grandir !

Pour les propriétés CSS de base, on va juste devoir donner une toute petite taille à notre élément HTML, ainsi qu'une couleur assez visible. Sans oublier la transition bien entendu ;)

```
#toutPetit:hover
{
  width: 100px;
  height: 100px;
}
```

Animations CSS

Element hors écran qu'on fait apparaître

Le rendu



Cette animation est un peu particulière, car on veut faire apparaître quelque chose qui se trouve en dehors de l'écran !

Pour ça, on va devoir mettre cet élément dans un "conteneur", car on ne peut pas détecter un :hover sur un élément en dehors de l'écran !

Le HTML

```
<article id="conteneur">
  <li>Passe le curseur ici !</li>

  <div id="horsEcranSlide"></div>
</article>
```

Le CSS

```
#horsEcranSlide {
  width: 100px;
  height: 100px;
  background-color: green;

  left: -130px;
  position: absolute;

  transition: 2s;
}
```

Pour les propriétés CSS de base, on doit donner une taille notre élément, ainsi qu'une couleur.

De plus, c'est très important de le placer HORS de l'écran grâce à "left: -130px" et "position: absolute"

Pour faire sortir l'élément de l'écran, on va utiliser la propriété translateX qui permet de faire sortir l'élément de 400px vers la droite

```
#conteneur:hover #horsEcranSlide
{
  transform: translateX(400px);
}
```


Transitions CSS

Texte caché

Le rendu



Le HTML

Ici, l'idée c'est de faire apparaître un texte (ici un paragraphe) lorsqu'on passe la souris dessus. On doit donc mettre directement le texte dans le paragraphe. Il faut également lui donner un ID à manipuler dans le CSS !

```
<p id="texteCache">Tu m'a découvert !</p>
```

Le CSS

```
#texteCache
{
  color: transparent;

  transition: 2s;
}
```

Pour le CSS de base du paragraphe on doit mettre la couleur du texte à "transparent" pour ne pas qu'on le voie de base.

Il faut également mettre la fameuse propriété "transition", avec le temps que tu veux ;)

Pour le CSS qui va se déclencher quand on va passer la souris sur le paragraphe, on change juste la couleur du texte !

```
#texteCache:hover
{
  color: black;
}
```

Transitions CSS

Texte qui devient flou

Le rendu

Je deviens flou quand tu me passes dessus !



Je deviens flou quand tu me passes dessus !

Le HTML

Ici, l'idée c'est de faire rendre flou un texte (ici un paragraphe) lorsqu'on passe la souris dessus. On doit donc mettre directement le texte dans le paragraphe. Il faut également lui donner un ID à manipuler dans le CSS !

```
<p id="texteFlou">Je deviens flou quand tu me passes dessus !</p>
```

Le CSS

```
#texteFlou {  
  color: black;  
  transition: 1s;  
}
```

Pour les propriétés CSS de notre paragraphe, il suffit juste de donner une couleur à son texte, ainsi que la transition.

Pour la transition, on va tout simplement rendre le texte transparent, et lui donner une text-shadow ! Les deux 0 servent à caler la text-shadow pile sur la place du texte, et ensuite, on donne sa largeur et sa couleur !

```
#texteFlou:hover  
{  
  color: transparent;  
  text-shadow: 0 0 3px black;  
}
```

Transitions CSS

Légende animée d'image

Le rendu



Le HTML

On ne peut pas mettre deux "background" dans un seul élément, c'est pour ça qu'il nous en faut deux !

```
<div id="imageConteneur">  
  <div id="overlay">Bonjour</div>  
</div>
```

Un qui contiendra l'image de fond, et un avec une couleur de fond semi-transparente pour l'effet visuel.

Le CSS (partie 1)

```
#imageConteneur  
{  
  height: 150px;  
  width: 150px;  
  
  background-image: url("html.png");  
  background-size: contain;  
  background-repeat: no-repeat;  
}
```

Pour commencer, on doit paramétrer l'élément qui contient l'image (le conteneur).

On lui donne une taille avec "height" et "width".

Puis on paramètre l'affichage de l'image avec les propriétés "background-xxxx".

"size" permet de cadrer l'image, et "repeat" permet de faire en sorte que l'image ne se s'affiche qu'une seule fois.

Animations CSS

Légende animée d'image

Le CSS (partie 2)

```
#overlay {  
  width: inherit;  
  height: 1px;  
  color: transparent;  
  background-color: transparent;  
  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  
  transition: 2s;  
}
```

Pour l'élément qui contient le texte et la couleur de fond, il faut d'abord lui donner une toute petite hauteur (1px). On utilise "inherit" sur la width pour qu'elle prenne la même largeur que l'élément qui contient l'image.

Ensuite, on met le texte et la couleur de fond en transparent.

Puis, on passe cet élément en flex pour pouvoir centrer le texte.

Et enfin, la transition !

Le CSS (partie 3)

```
#imageConteneur:hover #overlay  
{  
  height: inherit;  
  color: red;  
  background-color: rgba(0, 0, 0, 0.8);  
}
```

Pour finir, voici la transition ! On va dire que quand on passe la souris sur l'image, on veut rendre la couleur du texte non transparente, et celle du fond semi-transparente pour donner l'effet visuel de l'exemple. Enfin, pour faire apparaître petit à petit le fond, on doit changer sa taille, pour faire en sorte que le navigateur fasse tout seul l'animation depuis la taille de 1 pixel vers la taille de l'image !