



CSS



Propriétés supplémentaires



CSS : Transparence

- propriété css: opacity
- valeur: entre 0 (transparent) et 1 (opaque)
- Peut aussi être définie dans une couleur avec `rgba(red, green, blue, opacity)`

ceci est un paragraphe avec un background-color rouge et la propriété opacity à 1

ceci est le même paragraphe avec un background-color rouge et la propriété opacity à 0.5

ceci est un paragraphe avec la propriété background-color à la valeur `rgba(255, 0, 0, 0.5)`



● CSS : Transparence

1. Créez deux fichiers “imageGlow.html” et “imageGlow.css”, dans lesquels vous devez:
 - 1.1. Ajouter une image
 - 1.2. Rendre l’image partiellement transparente quand la souris passe dessus



CSS : Dégradés Linéaires (Linear gradient)

- Propriété: background-image
- Valeur: linear-gradient(*direction*, *color-stop1*, *color-stop2*, ...)
 - *direction*:
 - to right
 - to left
 - to bottom
 - to top
 - to bottom right
 - to top left
 - -90deg
 - 90deg
 - *color-stopX*:
 - preset:red, blue, yellow...
 - #ffbb00,
 - rgba(255, 3, 45, 0.5)
 - yellow 10%



CSS : Dégradés Radiaux (Radial gradient)

- Propriété: background-image
- Valeur: radial-gradient(*shape*, *start-color*, ..., *last-color*)
 - *shape*:
 - ellipse
 - circle
 - *color-stopX*:
 - preset:red, blue, yellow...
 - #ffbb00,
 - rgba(255, 2, 50, 0.5)



● CSS : Dégradés

1. Créez deux fichiers “exerciceGradient.html” et “exerciceGradient.css”, dans lesquels vous devez:
 - a. Créer un gradient linéaire diagonal qui passe par 3 couleurs de votre choix et qui prend toute la largeur et la moitié de la hauteur de page
 - b. Créer un gradient radial rond qui passe par 3 couleurs de votre choix et qui prend toute la largeur et l’autre moitié de la hauteur de page



● CSS : Paragraphe qui s'estompe

1. Créez deux fichiers “fadingText.html” et “fadingText.css”, dans lesquels vous devez:
 - a. Créer un paragraphe de plusieurs lignes (assez pour occuper la hauteur d'écran) qui occupe 25% de la largeur de page et qui va en s'estompant (opacité progressive)
 - b. Quand la souris passe sur le paragraphe, il doit apparaître complètement

Le résultat à obtenir est visible sur la page suivante.

- Indices:
 - Vous devez utiliser une balise `<p>` dans une balise `<div>`
 - La balise `<div>` doit utiliser un linear gradient avec du blanc plus ou moins opaque
 - La balise `<p>` doit utiliser la propriété css “position” avec la valeur “relative”, ce qui permet d'utiliser la propriété “z-index” sur la balise `<div>` et la balise `<p>`
 - La balise `<p>` doit avoir un z-index négatif et la balise `<div>` un z-index positif



● CSS : Paragraphe qui s'estompe

Rem laboriosam aperiarn qui magni. Ipsa laborum iure molestiae voluptas rerum reiciendis. Consequuntur error in blanditiis nihil a possimus ut. Dolores esse possimus maiores quo rerum accusantium vero. Magnam preferendis quam possimus libero qui ducimus quo. Autem tenetur sit consequatur placeat. Qui exercitationem ratione quisquam voluptate dolorem adipisci. Laboriosam aut non voluptatem ut et eum. Aut eum consequatur quos accusantium voluptatibus tenetur quo. Minima voluptatibus delectus explicabo aliquid. Fugiat fugiat neque aliquid non minima praesentium. Omnis dicta accusamus assumenda aut non rem nihil necessitatibus. Rem laboriosam aperiarn qui magni. Ipsa laborum iure molestiae voluptas rerum reiciendis. Consequuntur error in blanditiis nihil a possimus ut. Dolores esse possimus maiores quo rerum accusantium vero. Magnam preferendis quam possimus libero qui ducimus quo. Autem tenetur sit consequatur placeat. Qui exercitationem ratione quisquam voluptate dolorem adipisci. Laboriosam aut non voluptatem ut et eum. Aut eum consequatur quos accusantium voluptatibus tenetur quo. Minima voluptatibus delectus explicabo aliquid. Fugiat fugiat neque aliquid non minima praesentium. Omnis dicta accusamus assumenda aut non rem nihil necessitatibus. Rem laboriosam aperiarn qui magni. Ipsa laborum iure molestiae voluptas rerum reiciendis. Consequuntur error in blanditiis nihil a possimus ut. Dolores esse possimus maiores quo rerum accusantium vero. Magnam preferendis quam possimus libero qui ducimus quo. Autem tenetur sit consequatur placeat. Qui exercitationem

Rem laboriosam aperiarn qui magni. Ipsa laborum iure molestiae voluptas rerum reiciendis. Consequuntur error in blanditiis nihil a possimus ut. Dolores esse possimus maiores quo rerum accusantium vero. Magnam preferendis quam possimus libero qui ducimus quo. Autem tenetur sit consequatur placeat. Qui exercitationem ratione quisquam voluptate dolorem adipisci. Laboriosam aut non voluptatem ut et eum. Aut eum consequatur quos accusantium voluptatibus tenetur quo. Minima voluptatibus delectus explicabo aliquid. Fugiat fugiat neque aliquid non minima praesentium. Omnis dicta accusamus assumenda aut non rem nihil necessitatibus. Rem laboriosam aperiarn qui magni. Ipsa laborum iure molestiae voluptas rerum reiciendis. Consequuntur error in blanditiis nihil a possimus ut. Dolores esse possimus maiores quo rerum accusantium vero. Magnam preferendis quam possimus libero qui ducimus quo. Autem tenetur sit consequatur placeat. Qui exercitationem ratione quisquam voluptate dolorem adipisci. Laboriosam aut non voluptatem ut et eum. Aut eum consequatur quos accusantium voluptatibus tenetur quo. Minima voluptatibus delectus explicabo aliquid. Fugiat fugiat neque aliquid non minima praesentium. Omnis dicta accusamus assumenda aut non rem nihil necessitatibus



CSS : Ombres de texte

- Ombres de texte:
 - Propriété: `text-shadow`
 - Valeur: *shadow1, shadow2, ...* (ou *none*)
 - *shadowX: distanceHorizontale distanceVerticale flou couleur*
 - Exemple de valeur: `1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue`



● CSS : Ombres de texte

1. Créez deux fichiers “shadow.html” et “shadow.css”, dans lesquels vous devez:
 - 1.1. Créer deux titres <h1> avec du contenu
 - 1.2. Utiliser les propriétés d’ombre pour créer les deux ombres suivantes:



LA BLANQUETTE EST BONNE !!!

LA BLANQUETTE EST BONNE !!!

- Indices:
 - la première ombre est facile et ne demande qu’une ombre
 - la seconde demande plus d’une ombre, mais sans flou
 - Les deux sont noires

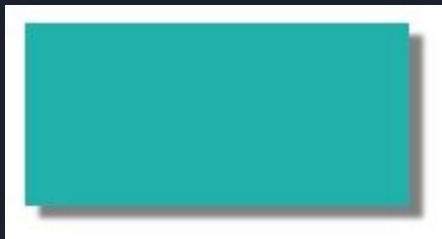


CSS : Ombres de boîte

- Ombres de texte:
 - Propriété: `box-shadow`
 - Valeur: *shadow1, shadow2, ...* (ou *none*)
 - *shadowX: distanceHorizontale distanceVerticale flou couleur*
 - Exemple de valeur: `2px -2px 1px rgba(0, 0, 0, 0.2)`

● CSS : Ombres de boîte

1. Dans vos fichiers “shadow.html” et “shadow.css”:
 - 1.1. Créez l’élément suivant, en utilisant une ombre:



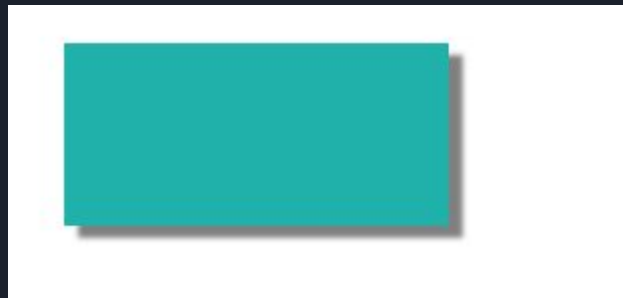


CSS : Transformations

- Les [transformation](#) CSS permettent d'effectuer des...transformations sur un élément
- Propriété: transform
- Valeurs:
 - translation
 - `translate(x, y): translate(10px, 10px)`
 - `translateX(x)`
 - `translateY(y)`
 - rotation
 - `rotate(angle): rotate(10deg)`
 - `rotateX(angle)`
 - `rotateY(angle)`
 - changement d'échelle
 - `scale(x, y): scale(1.5, 1.5)`
 - `scaleX(x)`
 - `scaleY(y)`
 - on peut enchaîner plusieurs transformations en les séparant par un espace

● CSS : Transformations (1)

1. Modifiez l'exercice précédent en effectuant une transformation sur le bloc vert pour donner l'impression qu'on "appuie" dessus:



● CSS : Transformations (2)

1. Dans vos fichiers “shadow.html” et “shadow.css”:
 - 1.1. Copiez le bloc vert pour en créer un second
 - 1.2. Doublez la taille de celui-ci quand la souris passe dessus (l'ombre doit être conservée):



● CSS : Transformations (3)

1. Dans vos fichiers “shadow.html” et “shadow.css”:
 - 1.1. Copiez le bloc vert pour en créer un troisième
 - 1.2. Faites en sorte qu’il se déplace ainsi quand la souris passe dessus:



● CSS : Créez un bouton avec un peu de relief

1. Combinez les techniques précédentes pour créer un bouton avec un peu de “relief”, un vrai feeling de bouton:



- Le bouton doit d'adapter automatiquement à la taille de son contenu. Pour cela, utilisez la propriété “display” avec la valeur “inline-block”

● CSS : Créez un bouton un peu plus “flat”

1. Créez un bouton plat:



- Le bouton doit d'adapter automatiquement à la taille de son contenu. Pour cela, utilisez la propriété “display” avec la valeur “inline-block”

● CSS : Créez un bouton qui “se soulève”

1. Créez un bouton qui se soulève:



- Le bouton doit d'adapter automatiquement à la taille de son contenu. Pour cela, utilisez la propriété “display” avec la valeur “inline-block”

● CSS : Tooltip

1. Créez un tooltip



- Indices:
 - Vous aurez besoin d'une balise `` dans une balise `<div>`
 - La balise `<div>` devra utiliser la propriété "position: relative"
 - La balise `` la propriété "position: absolute"
 - Pour cacher un élément: "visibility: hidden"
 - Pour le montrer: "visibility: visible"
 - [Article sur le positionnement](#)



CSS : Animations

- Qu'est-ce qu'une animation CSS ?
 - Une animation est liée à un élément (balise)
 - C'est un passage par un ensemble d'états entre un état de départ et un état de fin
 - Une animation possède des propriétés
 - Le sélecteur CSS pour définir les étapes de l'animation: @keyframes
- [Liste des éléments](#) qu'il est possible d'animer



CSS : Animations, propriétés

Propriété (élément)	Description	Ex. de valeurs
animation-name	Nom de l'animation (utilisé par le sélecteur @keyframes)	maSuperAnimation
animation-duration	Durée nécessaire pour passer de l'état initial à l'état final	2s 250ms
animation-delay	Délai entre le chargement de l' élément animé et son animation	2s 250ms -2s



CSS : Animations, propriétés

Propriété (élément)	Description	Ex. de valeurs
animation-direction	Direction de l'animation	normal reverse alternate alternate-reverse
animation-iteration-count	Nombre de répétitions de l'animation	5 2.5 infinite
animation-play-state	Permet de mettre en pause une animation	paused running



CSS : Animations, propriétés

Propriété (élément)	Description	Ex. de valeurs
animation-timing-function	Détermine la manière dont l'animation va se dérouler	linear ease ease-in ease-out ease-in-out steps(4 , jump-end) steps(4 , jump-start) steps(4 , jump-both) steps(4 , jump-none)



CSS : Animations, @keyframes

- Ce sélecteur permet de définir les étapes d'une animation
 - Deux étapes clés: 0% (from) et 100% (to)
 - Les étapes intermédiaires, exprimées en %, sont optionnelles



● CSS : Animations, exemple (1)

1. Créez deux fichiers “animation.html” et “animation.css”, et ajoutez les règles suivantes dans “animations.css”:

```
@keyframes fadeIn {  
  from {  
    background-color: white;  
  }  
  to {  
    background-color: blue;  
  }  
}
```

```
div {  
  width: 250px;  
  height: 250px;  
  background-color: white;  
  animation-name: fadeIn;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

2. Ajoutez une balise <div> dans le html pour pouvoir visualiser votre animation



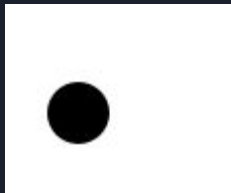
● CSS : Animations, exemple (2)

1. Modifiez l'animation précédente pour ajouter une couleur intermédiaire par laquelle l'animation va passer
2. Profitez-en pour jouer aussi sur les autres paramètres pour comprendre leur fonctionnement



● CSS : Animations, exemple (3)

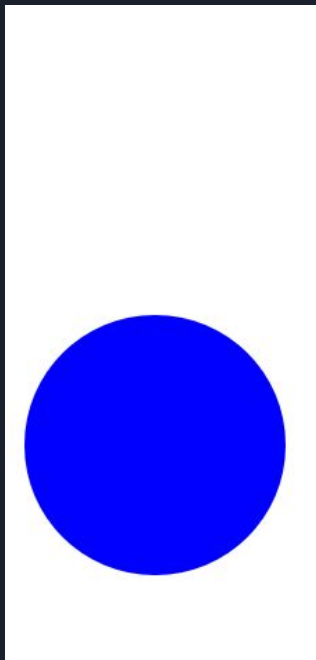
1. Créez cette animation (l'animation n'est pas continue):





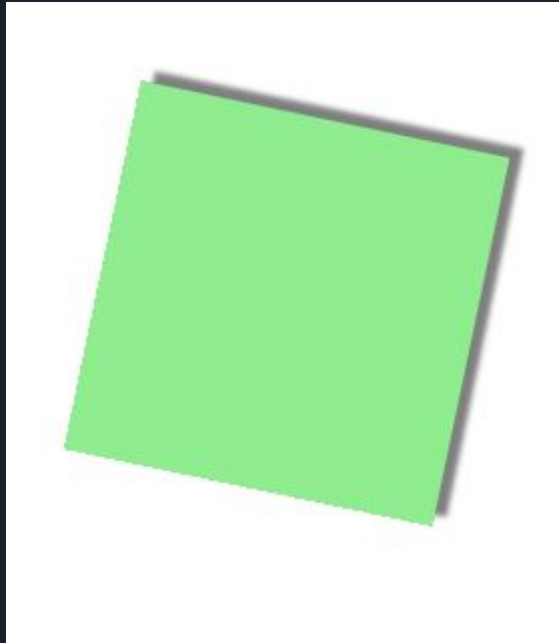
● CSS : Animations, exemple (4)

1. Créez une balle qui “rebondit” (l’animation est continue):



● CSS : Animations, exemple (5)

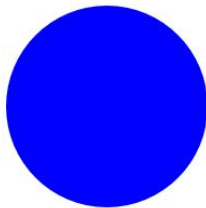
1. Créez un cube qui tourne sur lui-même (l'animation est continue):





● CSS : Animations, exemple (6)

1. Créez une balle qui rebondit plusieurs fois (l'animation est continue):



● CSS : Animations, exemple (7)

1. Reprenez le bouton en relief de la slide 17 pour rendre la transition entre l'état normal et l'état "appuyé" plus fluide grâce à une animation





● CSS, Rendez votre CV animé

- Reprenez votre CV en HTML:
 - ajoutez un peu de relief grâce aux nouvelles propriétés que nous avons vus
 - ajoutez des animations de votre choix
- Ce n'est pas grave si les animations nuisent à la lisibilité de votre cv