

MODULE 04 – Rabbit

HTML :

- Pas de nouvelles balises

CSS :

- background: linear-gradient(180deg, white 0%, white 50%, #333 50%, #333 100%);
 - background: linear-gradient(180deg, white 50%, #333 50%);
- flex-wrap: nowrap / wrap / wrap-reverse
- transition css :
 - transition-property : Précise les propriétés CSS à transformer all / none / nom-propriété
 - transition-duration : Précise la durée de la transition (en seconde ou milliseconde)
 - transition-timing-function : Précise la fonction de transition à utiliser
 - ease : Rapide sur le début puis décélère jusqu'à la fin.
 - linear : La vitesse est constante sur toute la durée de l'animation.
 - ease-in : Lent sur le début et accélère de plus en plus vers la fin.
 - ease-out : Rapide sur le début et décélère sur la fin.
 - ease-in-out : Le départ et la fin sont lents, accélère au milieu.
 - transition-delay : Indique la durée à attendre avant de débiter la transition (en seconde ou milliseconde)
 - raccourci
 - transition:all 3s ease 0s;
- Transformation css :
 - transform:fonctions()
 - translate() / une ou deux valeurs en px, em, % (x,y)
 - translateX() / une valeur en px, em, %
 - translateY() / une valeur en px, em, %
 - scale() / un coeficient multiplicateur à virgule ou non / une ou deux valeurs en px, em, % (x,y)
 - scaleX() / une valeur en px, em, %
 - scaleY() / une valeur en px, em, %
 - skew() / un angle en degré / une ou deux valeurs (x,y)
 - skewX() / une valeur en px, em, %
 - skewY() / une valeur en px, em, %
 - rotate() un angle en degré / une valeur positive ou négative
- Animation css :
 - animation-name: none (le nom de la nimation)
 - animation-duration: 3s (la durée de l'animation en seconde ou millisecondes)
 - animation-timing-function: ease (le type de fonction à appliquer)
 -
 - animation-delay: 0s (le délai avant le déclenchement de l'animation en seconde ou millisecondes)
 - animation-iteration-count:
 - Un nombre de cycles utilisés pour répéter une animation avant que celle-ci s'arrête. Accepte entier, nombre à virgule
 - Mot clé : infinite)
 - animation-direction:

- **normal** : L'animation est jouée dans le sens normal à chaque cycle. Autrement dit, au début de chaque cycle, l'animation est initialisée dans l'état de début. C'est le réglage par défaut.
 - **reverse** : L'animation est jouée dans le sens inverse à chaque cycle. Au début de chaque cycle, l'animation reprend depuis l'état de fin.
 - **alternate** : La lecture de l'animation se fait de façon alternée et change de sens à chaque cycle en commençant par le sens normal. Lorsque le sens est inversé, les étapes de l'animation sont effectuées de façon inversée et les fonctions de progression (timing functions) sont également inversées (ease-in sera ainsi remplacée par ease-out). Le premier cycle se fait dans le sens normal, le deuxième dans le sens inverse et ainsi de suite.
 - **alternate-reverse** : La lecture de l'animation se fait de façon alternée et change de sens à chaque cycle en commençant par le sens inverse (cf. ci-avant). Le premier cycle se fait dans le sens inverse, le deuxième dans le sens normal et ainsi de suite.
- animation-fill-mode: indique la façon dont une animation CSS doit appliquer les styles à sa cible avant et après son exécution). Valeurs possibles : none / forwards / backwards / both
 - animation-play-state: (détermine si une animation est en cours d'exécution ou si elle est en pause). Valeurs possibles : running / pause;
 -

Exemple d'animation :

```
@keyframes bounce{
    from{
        transform :translateY(0);
    }

    to{
        transform :translateY(-20px);
    }
}

.service i:hover {
    animation: bounce 0.35s ease infinite alternate;
}
```