

Parcours : Développeur Web

Dynamisez une page web  
avec des animations CSS !

*Romain Chenard - Projet 3*





# Plan

- Problématique / consignes
- Création
- Difficultés
- Conclusion

Adresse du projet :

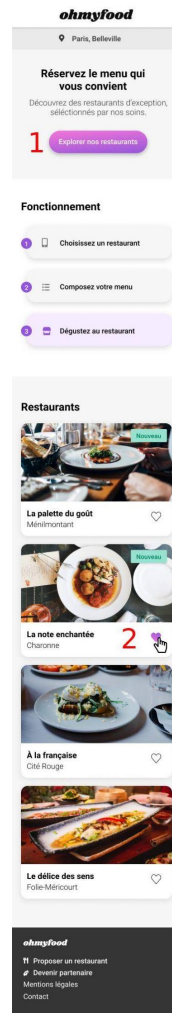
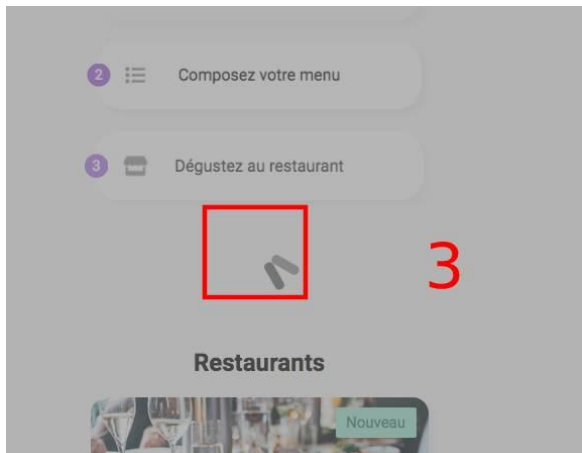
[https://romainc75.github.io/RomainChenard\\_3\\_10072021/](https://romainc75.github.io/RomainChenard_3_10072021/)



# Problématique

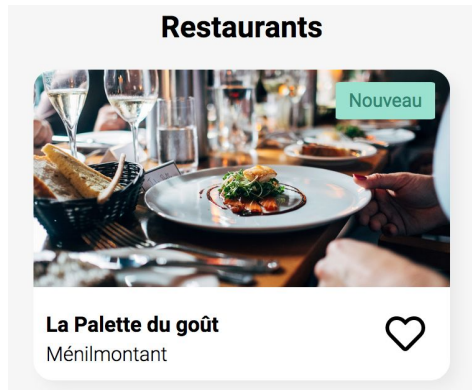
- Compatibilité
- Animations sur :
  - les boutons principaux (1)
  - coeur sur les cartes (2)
  - loading spinner (3)
  - coche vert à la droite des plats (4)
  - décalage à l'apparition des menus
- et autres ....

# Maquettes du site



# Processus de création

- Mise en page : position absolute et/ou relative
- Ancres : <a>
- Animations:
  - boutons “fonctionnement”
  - boutons violets (voir page suivante)
- Sass (2 exemples) :
  - Mixin pour les ombres (biggerShadow)
  - Mixin pour les Boutons violets (buttonV)
- autres





## Sass exemple 1: BiggerShadows

```
@mixin biggerShadow{  
  transform:scale(1.02);  
  box-shadow:8px 8px 10px #c4c4c4;  
}
```

```
&:hover{  
  z-index:1;  
  @include biggerShadow;  
}
```



## Sass exemple 2: mixin pour les boutons violets

```
@mixin buttonV{
  height:50px;
  background:$degradeViolet;
  border-radius:2em;
  color:■white;
  box-shadow:6px 6px 10px ■ #CFCFCF;
  @include flexCenter;
  position:relative;
  ::after{
    content:"";
    width:100%;
    height:100%;
    position:absolute;
    background:linear-gradient(■-170deg,
    ■lighten($secondary-color,4%),lighten($primary-color,3%));
    top:0;
    left:0;
    border-radius:2em;
    opacity:0;
    z-index:-1;
    transition:opacity 1s;
  }
  &:hover{
    z-index:1;
    @include biggerShadow;
  }
  :hover::after{
    opacity:1;
  }
}
```

## Difficultés 1/2 : Svg et Linear-gradient

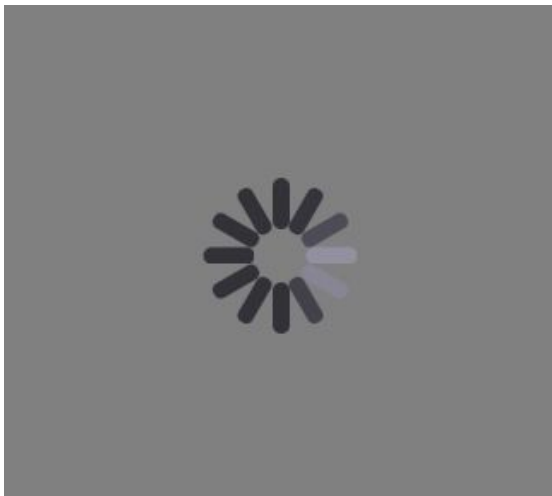


```
@keyframes animationHeart{  
  0% {  
    opacity:0;  
    background-size: 100% 0%;  
  }  
  50%{  
    opacity:0.4;  
  }  
  100% {  
    opacity:1;  
    background-size: 100% 100%;  
  }  
}
```





## Difficultés 2/2 : SpinLoader



```
$total:12;  
$stepDeg:360/$total;  
$stepTime:1/$total;  
  
@for $i from 1 through $total{  
  .b#{$i}{  
    transform:rotate(#{$i - 1} * $stepDeg)deg);  
    animation:cadre 1s #{$i - 1} * $stepTime}s infinite;  
    animation-fill-mode:forwards;  
  }  
}
```



# Conclusion

- Acquisition de nouvelles compétences
- Organisation face à un cahier des charges
- Sass - mixin / fonctions / keyframes
- Avoir un fichier CSS plus lisible
- Résultat plus “organique”
- Attention au rendu “kitsch”
- Prêt pour le projet suivant !!