

Styliser une plateforme de révision de code avec Stylus

Lors de la construction d'une application web moderne, le style ne se limite pas à rendre les choses jolies—il s'agit de créer une expérience utilisateur intuitive, réactive et engageante. J'ai récemment exploré la feuille de style basée sur Stylus d'une plateforme de révision de code alimentée par Vue.js, et son architecture CSS est un véritable trésor de techniques à débiller. Plongeons dans la manière dont cette application utilise Stylus pour façonner son interface utilisateur soignée, de la structuration de la mise en page aux effets de survol, tout en gardant le code maintenable et évolutif.

Pourquoi Stylus ? Un rapide aperçu

Stylus est un préprocesseur CSS qui élimine la verbosité du CSS traditionnel (pas de accolades ni de points-virgules nécessaires) et ajoute des fonctionnalités puissantes comme les variables, les mixins et l'imbrication. Le code fourni importe des variables de `variables.styl` et une feuille de style de base de `base.styl`, posant les bases de styles cohérents et réutilisables. Par exemple, la couleur primaire `#1CB2EF` est probablement définie dans `variables.styl` et réutilisée à travers les boutons et les arrière-plans.

Structurer la mise en page : sections et conteneurs

La page d'accueil de l'application est divisée en sections distinctes—.slide, .feature, .reviewer, .example, et .contact—chacune avec sa propre stratégie de style. Voici comment la section .slide (héro) est stylisée :

```
.slide
  height 800px
  position relative
  color #fff
  width 100%
  overflow hidden
  .bg
    background url("../img/home/hero.jpg") no-repeat
    background-size cover
    background-position-y 40%
    position 200% 200%
    width 100%
    height 100%
    padding-top 280px
```

Techniques clés :

- **Héro plein écran** : La `height 800px` et `width 100%` créent une bannière audacieuse en pleine largeur. `overflow hidden` garantit que rien ne déborde.
- **Image de fond** : La classe `.bg` utilise `background-size cover` pour mettre à l'échelle l'image héro proportionnellement, tandis que `background-position-y 40%` affine son alignement vertical pour un impact visuel.
- **Imbrication** : L'imbrication de Stylus garde les styles liés regroupés, améliorant la lisibilité par rapport au CSS plat.

Grilles réactives avec Flexbox et clearfix

La section `.feature` met en avant une mise en page à trois colonnes :

```
.feature
  height 450px
  padding 125px 0
  background white
.list
  width 1160px
  margin 0 auto
  display flex
  flex-direction row
  li
    height 200px
    padding-left 50px
    flex-grow 1
    &:first-child
      padding-left 0
  .short
    width 235px
    height 200px
    margin 0 auto
```

Points forts :

- **Flexbox** : `display flex` et `flex-direction row` alignent les éléments de la liste horizontalement, tandis que `flex-grow 1` garantit qu'ils s'étendent uniformément pour remplir le conteneur.
- **Centrage** : `width 1160px` associé à `margin 0 auto` centre le contenu, une technique classique pour les mises en page à largeur fixe.

- **Magie des pseudo-classes** : Le sélecteur `&:first-child` supprime le padding du premier élément, évitant un espacement inconfortable.

La section `.example` va plus loin avec une grille de cartes de révision, utilisant le mixin `clearfix()` :

```
.example
  .list
    clearfix()
  .row
    clearfix()
    li:first-child
      margin-left 0
  li
    height 354px
    margin-left 48px
    pull-left()
    margin-bottom 48px
```

- **Clearfix** : Ce mixin (probablement défini dans `base.styl`) gère le nettoyage des flottants, garantissant que les lignes s'empilent correctement dans les navigateurs plus anciens ou les mises en page personnalisées.
- **Grille basée sur les flottants** : `pull-left()` (un autre mixin utilitaire) fait flotter les éléments à gauche, avec `margin-left 48px` ajoutant des interlignes. Cette approche complète Flexbox pour une compatibilité plus large.

Stylisation interactive : effets de survol et transitions

Les cartes de révision dans `.example` brillent avec des interactions de survol fluides :

```
li
  .info
    position relative
    height 354px
    width 100%
    color white
    box-shadow 0 4px 4px 1px rgba(135,135,135,.1)
    overflow hidden
    cursor pointer
    &:hover
      img
        transform scale(1.2,1.2)
```

```

    -webkit-filter brightness(0.6)
.title
    -webkit-transform translate(0, -20px)
    opacity 1.0
.tips
    -webkit-transform translate(0, -10px)
    opacity 0.8
img
    height 100%
    -webkit-filter brightness(0.4)
    transition all 0.35s ease 0s

```

Décomposition :

- **Effets de survol** : Au survol, l'image s'agrandit (`transform scale(1.2,1.2)`) et s'éclaircit (`-webkit-filter brightness(0.6)`), tandis que les éléments de texte se déplacent vers le haut avec `translate` et ajustent l'opacité.
- **Transitions** : La `transition all 0.35s ease 0s` garantit des animations fluides pour toutes les propriétés, avec une durée de 350 ms et une courbe de progression.
- **Superposition** : `position absolute` sur `.text` la positionne sur l'image, avec `z-index 2` garantissant la visibilité.

Le bouton `.author` réagit également :

```

.author
    position absolute
    background black
    margin-left 30px
    margin-top 30px
    height 30px
    padding-left 20px
    padding-right 20px
    transition all 0.35s ease 0s
&:hover
    background #1cb2ef

```

Un simple changement de couleur du noir à la couleur de la marque #1CB2EF au survol ajoute une touche agréable.

Finition visuelle : ombres, boutons et icônes

Les ombres renforcent la profondeur, comme dans l'ombre de `.info box-shadow 0 4px 4px 1px rgba(135,135,135,.1)`.

Les boutons, comme dans `.contact`, sont stylisés avec soin :

```
.contact
  .rightbtn
    .more
      width 127px
      height 50px
      color #1CB2EF
      background white
      border-radius 3px
      border 1px solid #00A3E6
      -webkit-box-shadow 0px 1px 0px rgba(255,255,255,0.15) inset, 0px 1px 2px rgba(0,0,0,0.15)
```

- **Ombre interne** : L'ombre interne subtile (`inset`) associée à une ombre externe crée un effet de bouton enfoncé.
- **Cohérence** : La couleur de la bordure `#00A3E6` s'intègre dans la palette de la marque.

Les icônes, comme `.icon_crown`, utilisent des images de fond :

```
.icon_crown
  background url("../img/icon/crown@2x.png") no-repeat
  background-size contain
  width 49px
  height 52px
```

Le suffixe `@2x` suggère des actifs prêts pour les écrans Retina, avec `background-size contain` garantissant un dimensionnement approprié.

Meilleures pratiques et conclusions

Cette implémentation Stylus offre des leçons pour tout projet CSS : 1. **Utiliser des préprocesseurs** : Les imbrications et les mixins de Stylus (par exemple, `clearfix()`) simplifient les mises en page complexes. 2. **Équilibrer les mises en page** : Combinez Flexbox pour les navigateurs modernes avec des solutions de repli basées sur les flottants pour la robustesse. 3. **Améliorer l'UX** : Les transitions fluides et les effets de survol rendent l'interface utilisateur vivante. 4. **Rendre le code maintenable** : Utilisez des variables et des imports pour la cohérence à travers de grandes bases de code.

Que vous stylisiez une plateforme de révision de code ou un portfolio personnel, ces techniques peuvent élever votre jeu CSS. La prochaine fois que vous écrivez une feuille de style, pensez à la manière dont l'imbrication, les transitions et une touche d'ombre peuvent transformer votre design !