

Estilizando una Plataforma de Revisión de Código con Stylus

Al construir una aplicación web moderna, el estilo no se trata solo de hacer que las cosas se vean bien; se trata de crear una experiencia de usuario intuitiva, receptiva y atractiva. Recientemente exploré la hoja de estilo basada en Stylus de una plataforma de revisión de código impulsada por Vue.js, y su arquitectura CSS es un tesoro de técnicas dignas de desempacar. Vamos a sumergirnos en cómo esta aplicación utiliza Stylus para crear su interfaz de usuario pulida, desde la estructuración del diseño hasta los efectos de desplazamiento, todo mientras se mantiene el código mantenible y escalable.

¿Por qué Stylus? Un breve repaso

Stylus es un preprocesador de CSS que elimina la verbosidad del CSS tradicional (sin llaves ni puntos y comas requeridos) y añade características potentes como variables, mixins y anidamiento. El código proporcionado importa variables de `variables.styl` y una hoja de estilo base de `base.styl`, estableciendo el escenario para estilos consistentes y reutilizables. Por ejemplo, el color primario `#1CB2EF` probablemente esté definido en `variables.styl` y se reutilice en botones y fondos.

Estructurando el diseño: Secciones y contenedores

La página de inicio de la aplicación se divide en secciones distintas: `.slide`, `.feature`, `.reviewer`, `.example` y `.contact`, cada una con su propia estrategia de estilo. Aquí se muestra cómo se estiliza la sección `.slide` (hero):

```
.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
```

Técnicas clave:

- **Hero de pantalla completa:** La `height 800px` y `width 100%` crean un banner audaz de ancho completo. `overflow hidden` asegura que no se derrame ningún contenido.
- **Imagen de fondo:** La clase `.bg` utiliza `background-size cover` para escalar la imagen del héroe proporcionalmente, mientras que `background-position-y 40%` ajusta su alineación vertical para un impacto visual.
- **Anidamiento:** El anidamiento de Stylus mantiene los estilos relacionados agrupados, mejorando la legibilidad en comparación con el CSS plano.

Cuadriculas receptivas con Flexbox y clearfix

La sección `.feature` muestra un diseño de tres columnas:

```
.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
```

Destacados:

- **Flexbox:** `display flex` y `flex-direction row` alinean los elementos de la lista horizontalmente, mientras que `flex-grow 1` asegura que se expandan de manera uniforme para llenar el contenedor.
- **Centrado:** `width 1160px` emparejado con `margin 0 auto` centra el contenido, una técnica clásica para diseños de ancho fijo.

- **Mágica de pseudo-clases:** El selector `&:first-child` elimina el relleno del primer elemento, evitando un espaciado incómodo.

La sección `.example` lleva esto más allá con una cuadrícula de tarjetas de revisión, utilizando el 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:** Este mixin (probablemente definido en `base.styl`) maneja la limpieza de flotadores, asegurando que las filas se apilen correctamente en navegadores más antiguos o diseños personalizados.
- **Cuadrícula basada en flotadores:** `pull-left()` (otro mixin de utilidad) flota los elementos a la izquierda, con `margin-left 48px` añadiendo separadores. Este enfoque complementa Flexbox para una mayor compatibilidad.

Estilo interactivo: Efectos de desplazamiento y transiciones

Las tarjetas de revisión en `.example` brillan con interacciones de desplazamiento suaves:

```
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

```

Desglose:

- **Efectos de desplazamiento:** Al desplazarse, la imagen se escala (`transform scale(1.2,1.2)`) y se ilumina (`-webkit-filter brightness(0.6)`), mientras que los elementos de texto se desplazan hacia arriba con `translate` y ajustan la opacidad.
- **Transiciones:** La `transition all 0.35s ease 0s` asegura animaciones suaves para todas las propiedades, con una duración de 350 ms y una curva de suavizado.
- **Capas:** `position absolute` en `.text` la posiciona sobre la imagen, con `z-index 2` asegurando la visibilidad.

El botón `.author` también reacciona:

```

.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 cambio de color de negro al color de la marca `#1CB2EF` al desplazarse añade un toque encantador.

Pulido visual: Sombras, botones e iconos

Las sombras mejoran la profundidad, como se ve en `.info's box-shadow 0 4px 4px 1px rgba(135,135,135,.1)`.

Los botones, como en `.contact`, están estilizados con cuidado:

```
.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)
```

- **Sombra interna:** La sombra interna sutil (`inset`) emparejada con una sombra de caída externa crea un efecto de botón presionado.
- **Consistencia:** El color del borde `#00A3E6` se integra en la paleta de la marca.

Los iconos, como `.icon_crown`, utilizan imágenes de fondo:

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

El sufijo `@2x` sugiere activos listos para retina, con `background-size contain` asegurando una escalabilidad adecuada.

Mejores prácticas y conclusiones

Esta implementación de Stylus ofrece lecciones para cualquier proyecto de CSS: 1. **Usar preprocesadores:** Los anidamientos y mixins de Stylus (por ejemplo, `clearfix()`) simplifican diseños complejos. 2. **Equilibrar diseños:** Combine Flexbox para navegadores modernos con retrocesos basados en flotadores para robustez. 3. **Mejorar la UX:** Las transiciones suaves y los efectos de desplazamiento hacen que la interfaz de usuario se sienta viva. 4. **Mantenerlo mantenible:** Aproveche las variables y las importaciones para la consistencia en grandes bases de código.

Ya sea que esté estilizando una plataforma de revisión de código o un portafolio personal, estas técnicas pueden elevar su juego de CSS. La próxima vez que escriba una hoja de estilo, considere cómo el anidamiento, las transiciones y un toque de sombra pueden transformar su diseño!