Paper encuentro 6

Preprocesadores

Un preprocesador de CSS es una herramienta que nos permite escribir pseudo-código CSS que luego será convertido a CSS real. Este código se conforma de variables, condicionales, bucles o funciones. Podríamos decir que tenemos un lenguaje de programación que genera CSS.
El objetivo de estos preprocesadores es tener un código más sencillo de mantener y editar.

Existen tres preprocesadores más populares que son Sass, Less y Stylus, por si quieren investigarlos. Si bien todos son muy similares, Sass es el más demandado en la actualidad ya que cuenta con una enorme comunidad que se encarga de mantenerlo día a día, no es un dato menor.

Variables

Una de las principales características de los preprocesadores es que nos permiten tener **variables**. Gracias a las variables podemos **almacenar valores y reutilizarlos** en cualquier parte del código. Nos ahorrarán mucho trabajo cuando tengamos que editar un valor que se repite a lo largo de nuestro código.

Funciones, mixins y prefixing

Gracias a las **funciones y los mixins** podemos evitar escribir código duplicado. También los podemos usar para evitar escribir continuamente los prefijos propietarios de cada navegador.

Modularizar el código

Los preprocesadores ofrecen la característica de importar archivos. En lugar de escribir todo en un CSS, o tener varios archivos CSS, tendremos varios archivos preprocesables que serán importados en un solo archivo, que luego será convertido en el CSS resultante.
Esto nos permite modularizar nuestros archivos de una forma lógica y sencilla de mantener, teniendo distintos módulos o componentes para cada elemento de la web. Por ejemplo, podemos tener un archivo en el que se definan los estilos de los botones, otro para los formularios, otro para el sistema de grillas, etc. E igualmente podríamos dividirlo por páginas creando un archivo para el home, otro para el blog, otro para la sección de contacto, y así sucesivamente.
Al hacerlo de esta manera nos va a permitir contar con un código ordenado y prolijo que si queremos cambiar el color de los botones, vamos al archivo de los botones y lo cambiamos sin perder tiempo en buscar el código a modificar.

SASS

Cómo es la sintaxis de Sass

Lo primero que debes conocer de Sass es que existen dos tipos de sintaxis para escribir su código:

- Sintaxis Sass: esta sintaxis es un poco diferente de la sintaxis de CSS estándar. Por ejemplo, te evita colocar puntos y coma al final de los valores de propiedades. Además, las llaves no se usan y en su lugar se realizan indentados.
- Sintaxis SCSS: Es una sintaxis bastante similar a la sintaxis del propio CSS. De hecho, el código CSS es código SCSS válido. Podríamos decir que SCSS es código CSS con agregados poderosos.

En resumen, sabiendo que hay dos maneras que vamos a poder utilizar para escribir nuestro pseudocódigo, vamos a inclinarnos por SCSS por dos razones: la curva de aprendizaje es muchísimo menor, y seguimos manteniendo código CSS en nuestros archivos.

A trabajar

- Primero vamos a instalar Sass mediante node https://sass-lang.com/install
    a. sass --version
- Mostrar sass en terminal
    a. sass <source>/file.scss <dest>/file.css
    b. sass --watch <source>/file.scss <dest>/file.css
- Mostrar extensiones VSCode
    a. Sass
    b. LIve Sass

Import

- @import
- Tener un solo archivo compilable
- Demás archivos como bloque antecedidos con _(guión bajo) e importar en archivo principal, ejemplo: **@import “tables”**;

Variables

Para escapar una variable se usa el comodín #.
Esto es necesario cuando la variable está rodeada por comillas y de no ponerse el escape la variable pasaría como una cadena de caracteres

Mixins

- @mixin
- @include
- $content.

Extends

- @extend and %template

Funciones

- lighten
- darken
- inverter
- https://sass-lang.com/documentation/modules

Controles de flujo

- @if @else
- @each
- @for
- @while
- https://sass-lang.com/documentation/at-rules/control

Web Fonts

Mostrar ejemplo de inclusión de fuentes mediante css y mediante link en html desde Google Fonts

https://fonts.google.com/

Responsiveness / Media Queries

Se describe junto con la explicación y práctica de Sass

Los Media Types (tipos de medios)

Describen la categoría general de un dispositivo. Excepto cuando se utilizan los operadores lógicos not o only, el tipo de medio es opcional y será interpretada como all.

- all: apto para todos los dispositivos.
- print: destinado a material impreso y visualización de documentos en una pantalla en el modo de vista previa de impresión.
- screen: destinado principalmente a las pantallas.
- speech: destinado a sintetizadores de voz.

Operadores lógicos

Se pueden redactar queries utilizando operadores lógicos, incluyendo **not**, **and**, y **only**.

Además se puede combinar múltiples queries en una lista separada por comas múltiples; si cualquiera de las queries en la lista es verdadera, la hoja de estilo asociada es aplicada. Esto es equivalente a una operación lógica "**or**".

```
// Extra small devices (portrait phones, less than 576px)
@media (max-width: 575.98px) { ... }

// Small devices (landscape phones, 576px and up)
```

```
@media (min-width: 576px) and (max-width: 767.98px) { ... }

// Medium devices (tablets, 768px and up)
@media (min-width: 768px) and (max-width: 991.98px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) and (max-width: 1199.98px) { ... }

// Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }

@media all and (max-width: 700px){
  [class*="col-"] {
    width: 100%
}
```

device-width

El valor es de longitud (length) y se aplica a tipos de medios visuales y táctiles. Acepta los prefijos min / max.
Con ‘width’ media feature (función de medios) especificamos el ancho de la superficie de representación del dispositivo de salida.

@media screen and (device-width: 800px) { /*Reglas CSS*/ }

Device-height

El valor es de longitud (length) y se aplica a tipos de medios visuales y táctiles. Acepta los prefijos min / max.
Con ‘height’ media feature (función de medios) especificamos el alto de la superficie de representación del dispositivo de salida.

@media screen and (device-height: 600px) { /*Reglas CSS*/ }

Orientation

Valores: portrait | landscape No acepta los prefijos min / max.

El valor de «orientation» es ‘portrait’ (retrato) cuando el valor de la altura del medio de salida es mayor o igual al valor de la anchura de ese medio.
De lo contrario el valor es ‘landscape’ (paisaje).

@media all and (orientation:portrait) { /*Reglas CSS*/ }
@media all and (orientation:landscape) { /*Reglas CSS*/ }

Utilidades

https://www.sassmeister.com/
https://code.visualstudio.com/docs/languages/css
https://github.com/ritwickdey/vscode-live-sass-compiler/blob/master/docs/settings.md
https://sass-lang.com/documentation/
https://transfonter.org/
https://www.creativefabrica.com/es/webfont-generator/
https://developer.mozilla.org/es/docs/CSS/Media_queries