

Escribe CSS Para El Futuro Con Myth

CSS ha provisto una serie de nuevas características, tales como **gradientes**, **Sombras**, **Border Radio** y **animación**, que pueden ser usados únicamente con CSS. También se han añadido nuevas funciones, pero desafortunadamente, los navegadores todavía no soportan estas características. Si no puede esperar tanto para escribir el nuevo **CSS para el futuro**, entonces deberías conocer [Myth](#).



[Myth](#), a diferencia de otros [pre-procesadores](#) que cuentan con su propia sintaxis, utiliza la misma sintaxis de la especificación estándar. Puede utilizar variables, realizar operaciones matemáticas o de color, y escribir nuevas **propiedades CSS** en su forma oficial. Su objetivo es permitir a los desarrolladores **escribir CSS puro**, y al mismo tiempo ser capaz de utilizar la sintaxis del futuro estándar, como ahora.

Primeros Pasos

Para empezar, tenemos que instalar el binario de **Myth** para poder compilarlo con el **estándar del CSS actual**. No hay aplicación **GUI** como Codekit o [Koala](#) que soporte **Myth** en el momento de la escritura, por lo que esta es la única manera de compilar **Myth** en formato compatible con **CSS**.

En Terminal, escriba el siguiente comando:

```
[bash]npm install-g myth [/bash]
```

A continuación, puede utilizar este comando, para compilar **source.css** en **output.css**.

```
[bash]myth source.css output.css [/bash]
```

O bien, escriba esto para controlar la versión de source.css y compilarlo para output.css.

```
[bash] myth – ver output.css source.css [/bash]
```

Myth no trabaja con nuevas extensiones. Funciona con **.css** como se muestra arriba.

Variables

Vamos a empezar con las Variables. En CSS, se declara una variable, de la siguiente manera:

```
[css]

:root {
var-length: 10px;
var-color: #000;
}
.class {
background-color: var(color);
width: var(length);
}

[/css]
```

Myth compila este código en formato compatible para el navegador:

```
[css]

.class {
background-color: #000;
width: 20px;
}

[/css]
```

Operaciones Matemáticas

Como se menciona, también podemos realizar operaciones matemáticas con la **nueva función de CSS3 `calc()`**. Vamos a ampliar nuestro primer ejemplo de esto:

```
[css]

root {
var-length: 10px;
var-color: #000;
}
.class {
background-color: var(color);
width: calc(var(length) / 2);
}

[/css]
```

Myth compila los anteriores códigos, dejando lo siguiente:

```
[css]

.class {
background-color: #000;
width: 10px;
}

[/css]
```

Ajustes De Color

Myth también soporta algunas operaciones de color o ajustes como en [LESS o Sass](#). Se propone una nueva función estándar para que pueda ser incluido en las especificaciones CSS nombrada como **`color()`**, que incluye un conjunto de funciones de ajuste de color, como `tint()`, `shade()`, y `blend()`, sólo para nombrar unos pocos.

A continuación se muestra un ejemplo: aumentamos el lightness del color de fondo en un 80% y disminuimos el color del borde en un 50%.

```
[css]
```

```
:root {  
var-length: 20px;  
var-black: #000;  
var-white: #fff;  
}  
.class {  
background-color: color(var(black) lightness(+ 80%));  
border: var(border-width) solid color(var(white) lightness(-  
50%));  
width: calc(var(length) / 2);  
}
```

```
[/css]
```

Ese código producirá:

```
[css]
```

```
.class {  
background-color: rgb(204, 204, 204);  
border: 2px solid rgb(128, 128, 128);  
width: 10px;  
}
```

```
[/css]
```

Auto Prefijos

Myth también añadirá automáticamente el prefijo para propiedades CSS del navegador. Simplemente podemos escribir, por ejemplo, Shadow Box de CSS, de esta manera:

```
[css]
```

```
.class {  
box-shadow: 2px 1px 0px var(black);  
}
```

```
[/css]
```

La salida seria:

```
[css]
```

```
.class {  
-webkit-box-shadow: 2px 1px 0px #000;  
box-shadow: 2px 1px 0px #000;  
}
```

```
[/css]
```

Como vemos, **Myth** es una potente herramienta para escribir el **CSS para el futuro**. Ademas de las ventajas que brinda por encima de otros pre procesadores. Te ha gustado esta información? Compártela ;)