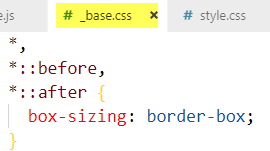
1. gulpfile.js



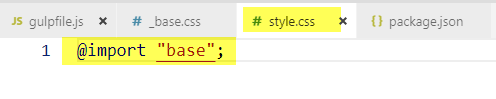
006 Organizing your CSS into partials

由于我们gulp的设置，postcss通过postcss-partial-import这个plugin, 会自动读入所有的 带有’\_’ prefix的partial css file，

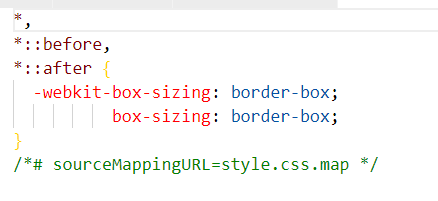
我们先创建一个\_base.css



然后在style.css当中导入这个partial, 由于postcss的作用，你可以使用像sass的@import相似的语法：



此时这个partial就会被包含在最终的style.css当中

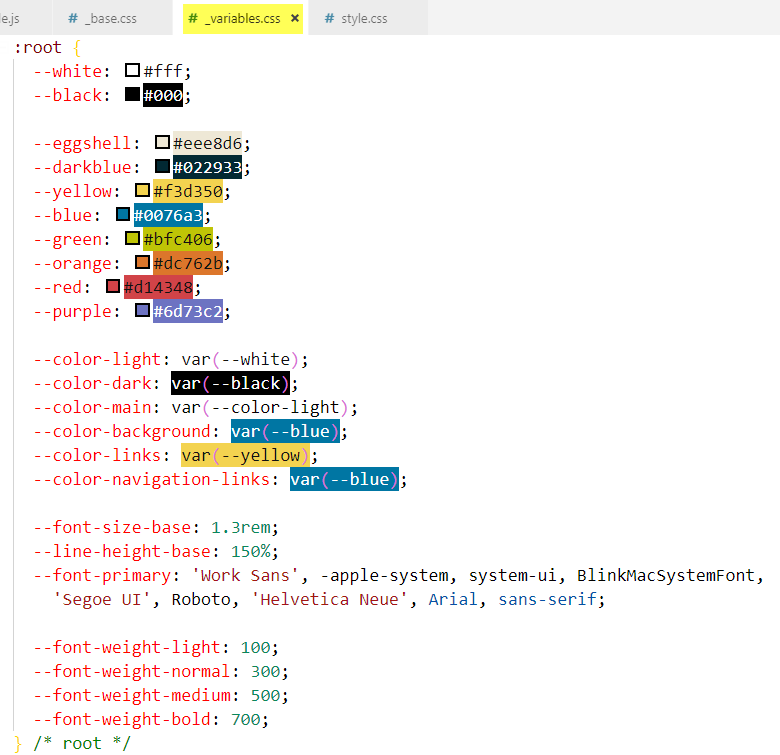


可以看到，由于postcss的autoprefixer的作用， box-sizing自动加入了vendor prefix for webkit

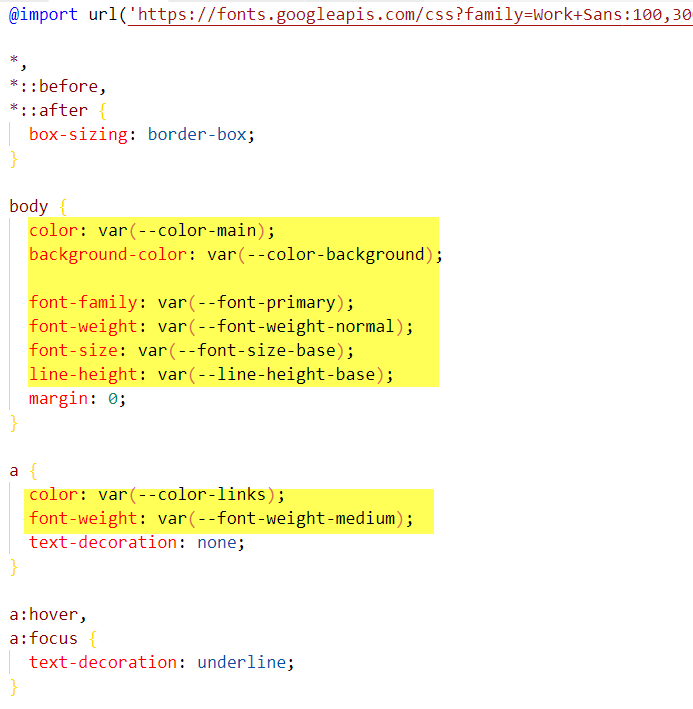
007 Creating variables

这里的语言是标准的css variables, 即变量名为--<name>, 然后必须包裹在一个selector里面，比如:root{} 里面，调用这个css variables的方法是var(--<name>)

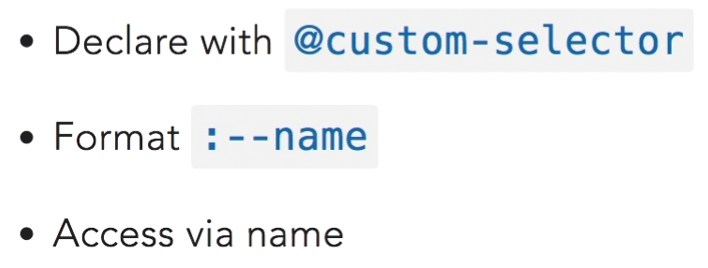
创建一个\_variables.css文件，并且import到style.css当中去



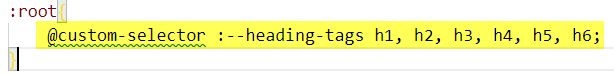
由于你已经在style.css当中import了，所以其他的\_css partial都可以访问这些variables



008 Using custom selectors

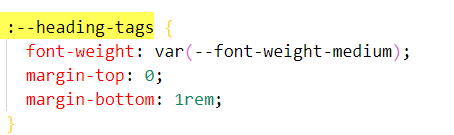


我们创建一个\_custom-selectors.css



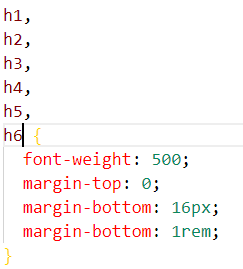
注意：

1. 它跟css variables一样，要指定起作用的scope, 我们这里选择 :root
2. 定义的时候，前面加上@custom-selector, 然后后面可以空格，然后是使用:--<name> 来给这个custom selector起名字，然后空格，后面跟被选择的selectors
3. 调用的是，使用:--<name> {} 的格式



这个custom selector指代的就是h1-h6 这几个tags

Build出来的css



009 Advanced pseudo-classes

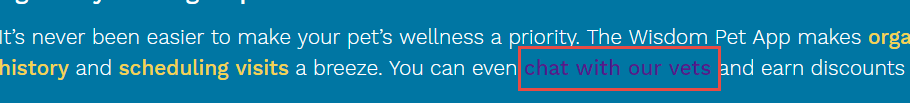
下面是几个例子，个人感觉不是很有用

Nextcss允许你使用以下几个新的pseudo-class

1. ：any-link

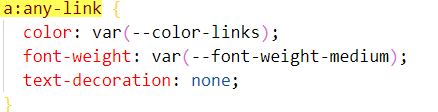
The :any-link pseudo-class represents an element that acts as the source anchor of a hyperlink. It matches an element if the element would match :link (:link means unvisited links) or :visited

正常情况下： visites link有自己的默认样式

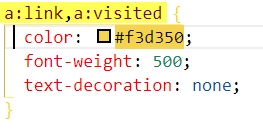


我们经常使用a {} 来覆盖它

:any-link 同时指代:link 和:visited两种状态

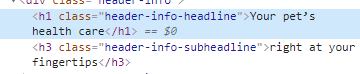


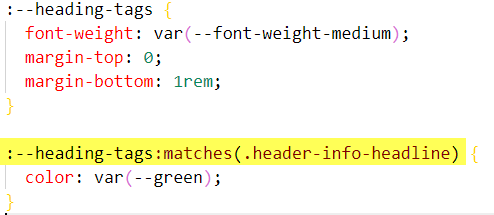
经过postcss 处理



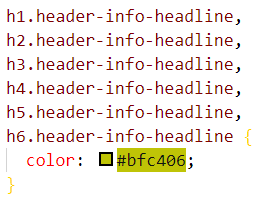
1. :matches() pseudo-class

比方说我们想对既有custom selector, --headings-tags，又有’.header-info-headline’ class的elements使用绿色





经postcss处理



1. :not() pseudo-class

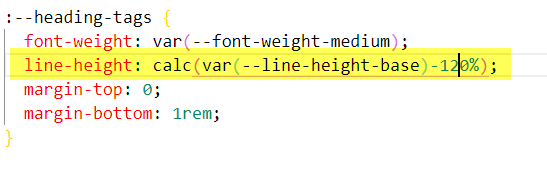
PostCSS plugin to transform :not() W3C CSS leve 4 pseudo class to :not() CSS level 3 selectors

跟现有的:not()很像，只是支持多个条件

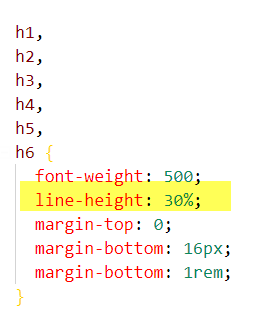


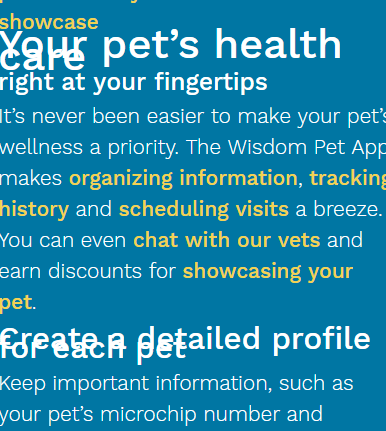
010 Variables with reduced calculations

calc() 现在大多数浏览器都支持，只是配合css variables使用的情况下可以使用postcss

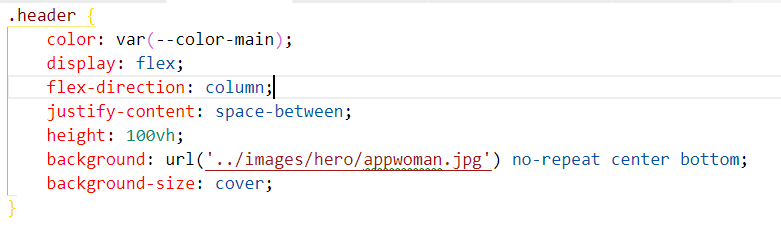


经过处理





011 Working with flexbox layouts

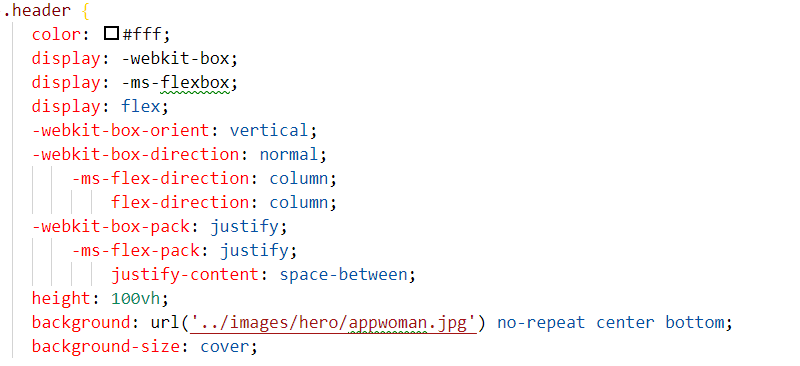


因为有height: 100vh, 这个css 会占据整个viewport, 之后的内容要scroll才能看见，

然后将.header里面的两个child elements vertically 排列，display: flex; flex-direction: column;

分别push向屏幕的上下方 justify-content: space-between;

经过postcss处理

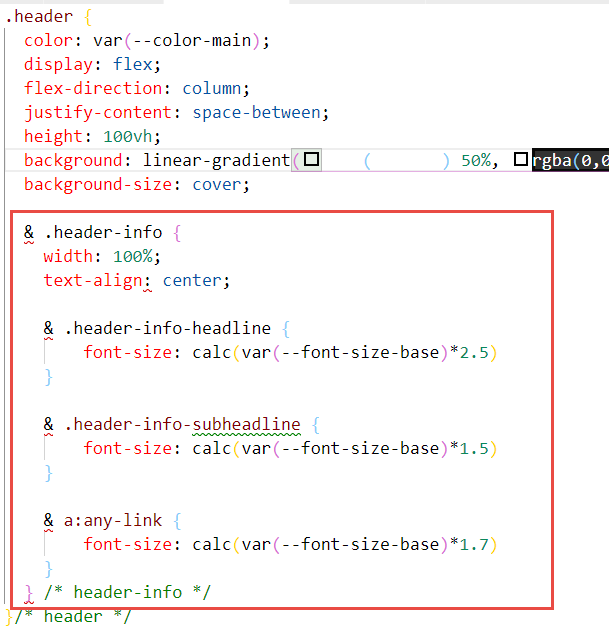


效果：



012 Nesting CSS rules

跟sass的nested rules非常类似，只是你必须使用&



013 Creating property groups

custom properties set & @apply

Allows you to store a set of properties in a named custom property, then reference them in other style rules.

:root {

--danger-theme: {

color: white;

background-color: red;

}

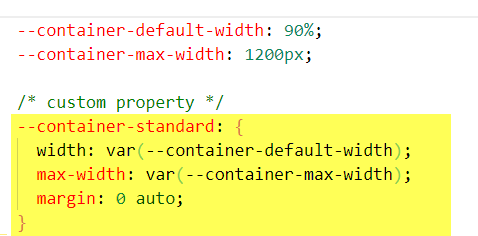
}

.danger {

@apply --danger-theme;

}

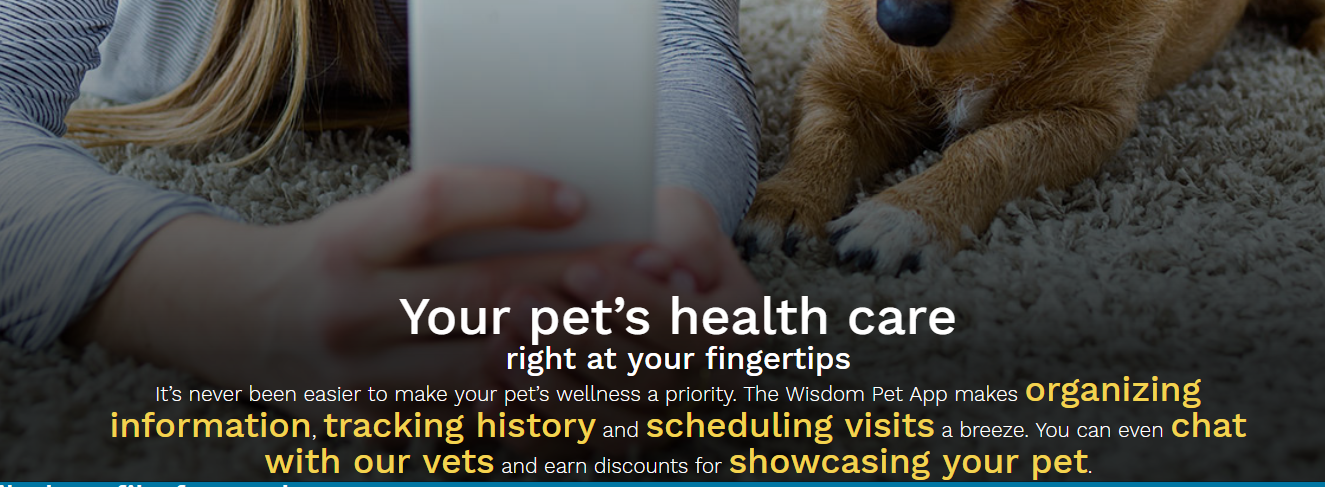
我们也定义一个property group, 叫做 –container-standard, 里面包含两个property



然后使用@apply keyword,调用这个 custom property group



效果是限制了宽度



014 Custom media queries

## media queries ranges

Allows to replace min-/max- with <= & >= (syntax easier to read).

@**media** (width >= 500px) and (width <= 1200px) {

/\* your styles \*/

}

/\* or coupled with custom media queries \*/

@**custom-media** --only-medium-screen (width >= 500px) and (width <= 1200px);

@**media** (--only-medium-screen) {

/\* your styles \*/

}

并不是一个多特别的技术，只是更适合阅读

016 Using color functions

## color() function

A color function to modify colors (transpiled to: rgba()).

a {

color: color(red alpha(-10%));

}

a:hover {

color: color(red blackness(80%));

}

### List of color-adjuster

* [red( | green( | blue( | alpha( | a(] ['+' | '-']? [<number> | <percentage>] )
* [red( | green( | blue( | alpha( | a(] '\*' <percentage> )
* @todo
* @todo
* @todo
* [hue( | h(] ['+' | '-' | '\*']? <angle> )
* [saturation( | s(] ['+' | '-' | '\*']? <percentage> )
* [lightness( | l(] ['+' | '-' | '\*']? <percentage> )
* [whiteness( | w(] ['+' | '-' | '\*']? <percentage> )
* [blackness( | b(] ['+' | '-' | '\*']? <percentage> )
* tint( <percentage> )
* shade( <percentage> )
* blend( <color> <percentage> [rgb | hsl | hwb]? )
* @todo
* contrast( <percentage>? )

Notes:

* some adjusts have shortcuts,
* can be used on every value on any property,
* some values can use add/subtract/scale modifiers or a direct value.

### Examples

whatever {

color: color(red a(10%));

background-color: color(red lightness(50%)); /\* == color(red l(50%)); \*/

border-color: color(hsla(125, 50%, 50%, .4) saturation(+ 10%) w(- 20%));

}

比如

&:hover{

color: color(var(--orange) saturation(+20%))

}