

-00CSS + Sass

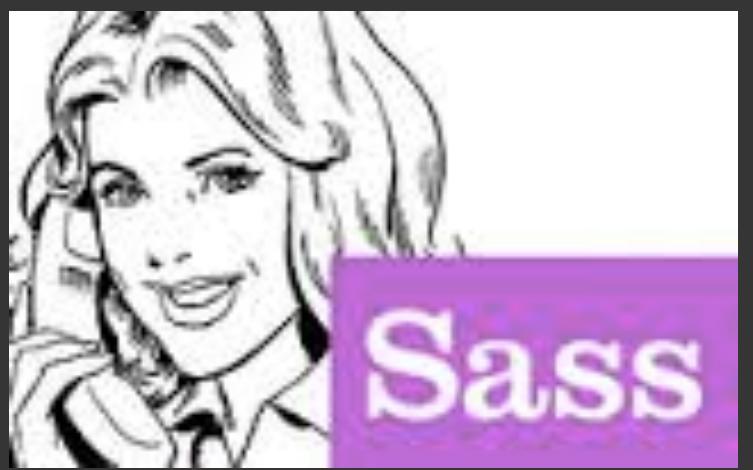
2015-10-18

by @大漠

鞋城国语“最好”的前端

手淘最丑的“伪前端”

页面仔



oocss



Nicole Sullivan

CSS太“脆弱”

为什么会有OOCSS？

“**更少**”的CSS

“**更多**”可重用的CSS

“**更易**”理解的CSS

什么是OOCSS?

结构(**如何组合在一起**)和皮肤(**如何看**)的“分离”

内容(**模块**)和容器(**布局**)的“分离”

结构和皮肤分离

```
#button {  
    width: 200px;  
    height: 50px;  
    padding: 10px;  
    border: solid 1px #ccc;  
    background: linear-gradient(#ccc, #222);  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}  
  
#box {  
    width: 400px;  
    overflow: hidden;  
    border: solid 1px #ccc;  
    background: linear-gradient(#ccc, #222);  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}  
  
#widget {  
    width: 500px;  
    min-height: 200px;  
    overflow: auto;  
    border: solid 1px #ccc;  
    background: linear-gradient(#ccc, #222);  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}
```

```
border: solid 1px #ccc;  
background: linear-gradient(#ccc, #222);  
box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;
```

将皮肤从结构中抽离出来

结构和皮肤分离

```
#button {  
    width: 200px;  
    height: 50px;  
    padding: 10px;  
    border: solid 1px #ccc;  
    background: linear-gradient(#ccc, #222);  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}  
  
#box {  
    width: 400px;  
    overflow: hidden;  
    border: solid 1px #ccc;  
    background: linear-gradient(#ccc, #222);  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}  
  
#widget {  
    width: 500px;  
    min-height: 200px;  
    overflow: auto;  
    border: solid 1px #ccc;  
    background: linear-gradient(#ccc, #222);  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}
```

分离前

```
.button {  
    width: 200px;  
    height: 50px;  
}  
  
.box {  
    width: 400px;  
    overflow: hidden;  
}  
  
.widget {  
    width: 500px;  
    min-height: 200px;  
    overflow: auto;  
}  
  
.skin {  
    border: solid 1px #ccc;  
    background: linear-gradient(#ccc, #222);  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}
```

分离后

容器和内容分离

```
#sidebar h3 {  
    font-family: Arial;  
    font-size: 20px;  
    line-height: 1.5;  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}  
  
#box h3{  
    font-family: Arial;  
    font-size: 20px;  
    line-height: 1.5;  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}  
  
#footer h3 {  
    font-family: Arial;  
    font-size: 20px;  
    line-height: 1.5;  
    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}
```

```
<div id="header">  
    <h3>...</h3>  
</div>  
  
<div id="sidebar">  
    <h3>...</h3>  
</div>  
  
<div id="footer">  
    <h3>...</h3>  
</div>
```

打破容器与内容对称依赖

容器和内容分离

```
.title {  
  font-family: Arial;  
  font-size: 20px;  
  line-height: 1.5;  
  box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;  
}
```

打破容器与内容对象依赖

```
<div id="header">  
  <h3 class="title">...</h3>  
</div>  
  
<div id="sidebar">  
  <h3 class="title">...</h3>  
</div>  
  
<div id="footer">  
  <h3 class="title">...</h3>  
</div>
```

OOCSS优点和缺点

优点

代码更好

性能更好

设计一致

缺点

更大精力编写CSS

更大精力编写HTML

工具来救援

CSS处理器(像Sass/LESS/Stylus)



Sass

Sass是什么？

工具

处理器

Sass功能&特性

Sass功能&特性

定义变量(\$)

混合宏(@mixin)&占位符(%placeholder)&扩展(@extend)

嵌套&继承

运算&函数

Map & List

控制命令

...

Sass Vs. SCSS

SCSS

```
$width: 300px;  
  
.container {  
    width: $width;  
}
```

Sass

```
$width: 300px  
  
.container  
    width: $width
```

Sass 变量



Sass 混合宏

声明混合宏符号

混合宏名称

混合宏参数

```
@mixin box-shadow($shadow...) {  
  @if length($shadow) >= 1 {  
    @include prefixer(box-shadow, $shadow);  
  } @else {  
    $shadow: 0 0 4px rgba(0,0,0,.3);  
    @include prefixer(box-shadow, $shadow);  
  }  
}
```

混合宏代码块

使用@include

引用混合宏

```
.box {  
  @include box-shadow;  
}  
  
.box {  
  @include box-shadow(inset 1px 1px 2px rgba(black,.3));  
}
```

调用默认混合宏

调用带参数的混合宏

Sass 继承

```
.mt5 {  
  margin-top: 5px;  
}
```

Sass

```
.block {  
  @extend .mt5;  
}
```

Sass

```
.mt5, .block {  
  margin-top: 5px;  
}
```

CSS

@extend 来继承
类名

Sass 占位符

使用%声明占位符

```
%mt5 {  
  margin-top: 5px;  
}
```

Sass

@extend 来扩展

```
.block {  
  @extend %mt5;  
}
```

Sass

```
.block {  
  margin-top: 5px;  
}
```

CSS

混合宏 vs 继承 vs 占位符

混合宏@mixin

```
@mixin mt($var){  
  margin-top: $var;  
}
```

Sass

```
.block{  
  @include mt(5px);  
}
```

Sass

```
.block {  
  margin-top: 5px;  
}
```

CSS

继承class

```
.mt5{  
  margin-top:5px;  
}
```

Sass

```
.block{  
  @extend .mt5;  
}
```

Sass

```
.mt5, .block {  
  margin-top: 5px;  
}
```

CSS

占位符%placeholder

```
%mt5{  
  margin-top:5px;  
}
```

Sass

```
.block{  
  @extend %mt5;  
}
```

Sass

```
.block {  
  margin-top: 5px;  
}
```

CSS

单类 vs 多类

多类

```
/*padding and margin all*/
.paxs{padding:5px;}
.pas {padding: 10px;}
.pam{padding:15px;}
.pal{padding:20px;}
.paxl{padding: 25px;}
.paxxl{padding: 30px;}
```

```
.maxs{margin:5px;}
.mas {margin: 10px;}
.mam{margin:15px;}
.mal{margin:20px;}
.maxl{margin: 25px;}
.maxxl{margin: 30px;}
```

CSS

```
<div class="header maxs
paxs"></div>
```

HTML

单类

```
/*padding and margin all*/
%paxs{padding:5px;}
%pas {padding: 10px;}
%pam{padding:15px;}
%pal{padding:20px;}
%paxl{padding: 25px;}
%paxxl{padding: 30px;}
```

```
%maxs{margin:5px;}
%mas {margin: 10px;}
%mam{margin:15px;}
%mal{margin:20px;}
%maxl{margin: 25px;}
%maxxl{margin: 30px;}
```

Sass

```
.header {
  @extend %paxs;
  @extend %maxs;
}
```

Sass

```
.header {
  padding:5px;
  margin:5px;
}
```

CSS

```
<div class="header"></div>
```

HTML

Sass嵌套

```
.header {  
  padding: 5px;  
  margin: 5px;  
  
  nav {  
    background-color: red;  
  }  
  
  &:after {  
    content: "";  
    display: table;  
    clear: both;  
  }  
  
  .home & {  
    background-color: green;  
  }  
}
```

Sass

```
.header {  
  padding: 5px;  
  margin: 5px;  
}  
.header nav {  
  background-color: red;  
}  
.header:after {  
  content: "";  
  display: table;  
  clear: both;  
}  
.home .header {  
  background-color: green;  
}
```

CSS

Sass运算

```
$wide: 200px;  
$space: $wide / 10;  
  
.content {  
    width: $wide * 5;  
    padding: $space;  
}  
  
.block {  
    width: $wide - $space;  
}
```

Sass

```
.content {  
    width: 1000px;  
    padding: 20px;  
}  
  
.block {  
    width: 180px;  
}
```

CSS

Sass函数

```
$link-color: green;  
  
.a {  
  color: $link-color;  
  background-color: lighten($link-color,5%);  
  
  &:hover {  
    color: darken($link-color,5%);  
  }  
}
```

Sass

```
.a {  
  color: green;  
  background-color: #009a00;  
}  
  
.a:hover {  
  color: #006700;  
}
```

CSS

```
@function pxTorem($px,$browser-default-font-size){  
  @return $px / $browser-default-font-size * 1rem;  
}  
  
h2 {  
  font-size: pxTorem(32px,16px);  
}
```

Sass

```
h2 {  
  font-size: 2rem;  
}
```

CSS

字符串函数

三元函数

数字函数

自定义函数

列表函数

颜色函数

Sass的&符

```
a {  
  color: orange;  
  
  &:hover {  
    color: darken(orange,5%);  
  }  
  
  .header & {  
    color: lighten(orange,5%);  
  }  
}  
  
.block {  
  color: blue;  
  
  &__element{  
    color: yellow;  
  }  
  &--modify {  
    color: green;  
  }  
}
```

Sass

```
a {  
  color: orange;  
}  
a:hover {  
  color: #e69500;  
}  
.header a {  
  color: #ffaela;  
}  
  
.block {  
  color: blue;  
}  
.block__element {  
  color: yellow;  
}  
.block--modify {  
  color: green;  
}
```

CSS

Sass的@at-root

```
.block {  
  color: green;  
  
  ul {  
    list-style:none outside none;  
  
    li {  
      margin:0;  
  
      @at-root a {  
        color:green;  
      }  
    }  
  }  
}
```

Sass

```
.block {  
  color: green;  
}  
.block ul {  
  list-style: none outside none;  
}  
.block ul li {  
  margin: 0;  
}  
a {  
  color: green;  
}
```

CSS

Sass的控制命令

@if / @if ... @else

@for

@each

@while

Sass的@if/@if...@else

```
@mixin blockOrHidden($boolean:true) {  
  @if $boolean {  
    @debug "$boolean is #{$boolean}";  
    display: block;  
  }  
  @else {  
    @debug "$boolean is #{$boolean}";  
    display: none;  
  }  
}  
  
.block {  
  @include blockOrHidden;  
}  
  
.hidden{  
  @include blockOrHidden(false);  
}
```

Sass

```
.block {  
  display: block;  
}  
  
.hidden {  
  display: none;  
}
```

CSS

Sass的@for

@for \$var from <start> through <end>

```
$color: green !default;  
@for $i from 1 through 4 {  
  .#$color-$i{  
    color: darken($color,5% * $i);  
  }  
}
```

Sass

```
.green-1 {  
  color: #006700;  
}  
  
.green-2 {  
  color: #004d00;  
}  
  
.green-3 {  
  color: #003400;  
}  
  
.green-4 {  
  color: #001a00;  
}
```

CSS

@for \$var from <start> to <end>

```
$color: green !default;  
@for $i from 1 to 4 {  
  .#$color-$i{  
    color: darken($color,5% * $i);  
  }  
}
```

Sass

```
.green-1 {  
  color: #006700;  
}  
  
.green-2 {  
  color: #004d00;  
}  
  
.green-3 {  
  color: #003400;  
}
```

CSS

Sass的@each

@each \$var in <list>

```
$list: adam john wynn mason kuroi;

@mixin author-images {
  @each $author in $list {
    .photo-#{$author} {
      background: url("/images/avatars/#{$author}.png") no-repeat;
    }
  }
}

.author-bio {
  @include author-images;
}
```

Sass

```
.author-bio .photo-adam {
  background: url("/images/avatars/adam.png") no-repeat;
}
.author-bio .photo-john {
  background: url("/images/avatars/john.png") no-repeat;
}
.author-bio .photo-wynn {
  background: url("/images/avatars/wynn.png") no-repeat;
}
.author-bio .photo-mason {
  background: url("/images/avatars/mason.png") no-repeat;
}
.author-bio .photo-kuroir {
  background: url("/images/avatars/kuroir.png") no-repeat;
}
```

CSS

Sass的List

```
$list:();  
$list:(#b4d455,42,"awesome");  
$list-space: "item-1" "item-2" "item-3";  
$list: (  
    ("item-1.1", "item-1.2", "item-1.3"),  
    ("item-2.1", "item-2.2", "item-2.3"),  
    ("item-3.1", "item-3.2", "item-3.3")  
);
```

定义列表方式

```
@function first($list){  
    @return nth($list,1);  
}  
  
@function last($list){  
    @return nth($list,length($list));  
}  
$colors: green,blue,orange,yellow;
```

```
.header {  
    color: first($colors);  
}  
.footer{  
    color: last($colors);  
}
```

Sass

```
.header {  
    color: green;  
}  
  
.footer {  
    color: yellow;  
}
```

CSS

Sass的Map

```
$map: (  
    key: value,  
    other-key: other-value  
) ;
```

Map语法

```
$configuration: (  
    padding: 1em,  
    margin: 0 1em,  
    color: grey  
) ;  
  
.element {  
    color: map-get($configuration, color);  
    padding: map-get($configuration, padding);  
    margin: map-get($configuration, margin);  
  
    &::before {  
        background-color: map-get($configuration, color);  
    }  
}
```

Sass

```
.element {  
    color: grey;  
    padding: 1em;  
    margin: 0 1em;  
}  
.element::before {  
    background-color: grey;  
}
```

CSS



oocss + Sass

OOCSS & Sass

[Twitter](#)

[Facebook](#)



oocss

OOCSS

Twitter Facebook

```
<a href="#" class="twitter">Twitter</a>
<a href="#" class="facebook">Facebook</a>
```

```
.twitter {
  border:3px solid #000;
  padding:10px 20px;
  color:#fff;
  border-radius:10px;
  background:red;
  text-decoration: none;
}

.facebook {
  border:3px solid #000;
  padding:10px 20px;
  color:#fff;
  border-radius:10px;
  background:blue;
  text-decoration: none;
}
```

```
.twitter,.facebook{
  border:3px solid #000;
  padding:10px 20px;
  color:#fff;
  border-radius:10px;
  text-decoration: none;
}

.twitter {
  background:red;
}

.facebook {
  background:blue;
}
```

OOCSS

Twitter Facebook

```
<a href="#" class="btn btn-twitter">Twitter</a>
<a href="#" class="btn btn-facebook">Facebook</a>
```

```
.btn{
    border:3px solid #000;
    padding:10px 20px;
    color:#fff;
    border-radius:10px;
    text-decoration: none;
}
.btn-twitter {
    background:red;
}
.btn-facebook {
    background:blue;
}
```

Sass能带来什么变化？

OOCSS & Sass

Twitter Facebook

```
<a href="#" class="btn-twitter">Twitter</a>
<a href="#" class="btn-facebook">Facebook</a>
```

```
@mixin btn{
    border: 3px solid #000;
    padding: 10px 20px;
    color: #fff;
    border-radius: 10px;
    text-decoration: none;
}
.btn-twitter {
    background: red;
    @include btn;
}
.btn-facebook {
    background: blue;
    @include btn;
}
```

Sass

```
.btn-twitter {
    background: red;
    border: 3px solid #000;
    padding: 10px 20px;
    color: #fff;
    border-radius: 10px;
    text-decoration: none;
}
.btn-facebook {
    background: blue;
    border: 3px solid #000;
    padding: 10px 20px;
    color: #fff;
    border-radius: 10px;
    text-decoration: none;
}
```

CSS

OOCSS & Sass

Twitter Facebook

```
<a href="#" class="btn-twitter">Twitter</a>
<a href="#" class="btn-facebook">Facebook</a>
```

```
.btn{
    border: 3px solid #000;
    padding: 10px 20px;
    color: #fff;
    border-radius: 10px;
    text-decoration: none;
}
.btn-twitter {
    background: red;
    @extend .btn;
}
.btn-facebook {
    background: blue;
    @extend .btn;
}
```

Sass

```
.btn, .btn-twitter, .btn-facebook
{
    border: 3px solid #000;
    padding: 10px 20px;
    color: #fff;
    border-radius: 10px;
    text-decoration: none;
}
.btn-twitter {
    background: red;
}
.btn-facebook {
    background: blue;
}
```

CSS

OOCSS & Sass

Twitter Facebook

```
<a href="#" class="btn-twitter">Twitter</a>
<a href="#" class="btn-facebook">Facebook</a>
```

```
%btn{
    border: 3px solid #000;
    padding: 10px 20px;
    color: #fff;
    border-radius: 10px;
    text-decoration: none;
}
.btn-twitter {
    background: red;
    @extend %btn;
}
.btn-facebook {
    background: blue;
    @extend %btn;
}
```

Sass

```
.btn-twitter, .btn-facebook {
    border: 3px solid #000;
    padding: 10px 20px;
    color: #fff;
    border-radius: 10px;
    text-decoration: none;
}
.btn-twitter {
    background: red;
}
.btn-facebook {
    background: blue;
}
```

CSS

OOCSS & Sass

Twitter

Facebook

```
<a href="#" class="btn-twitter">Twitter</a>
<a href="#" class="btn-facebook">Facebook</a>
```

```
$vars:(  
  prefix-class: btn,  
  twitter: red,  
  facebook: blue  
)  
%btn{  
  border: 3px solid #000;  
  padding: 10px 20px;  
  color: #fff;  
  border-radius: 10px;  
  text-decoration: none;  
}  
.#{map-get($vars,prefix-class)}-twitter {  
  background: map-get($vars,twitter);  
  @extend %btn;  
}  
.#{map-get($vars,prefix-class)}-facebook {  
  background: map-get($vars,facebook);  
  @extend %btn;  
}
```

Sass

```
.btn-twitter, .btn-facebook {  
  border: 3px solid #000;  
  padding: 10px 20px;  
  color: #fff;  
  border-radius: 10px;  
  text-decoration: none;  
}  
.btn-twitter {  
  background: red;  
}  
.btn-facebook {  
  background: blue;  
}
```

CSS

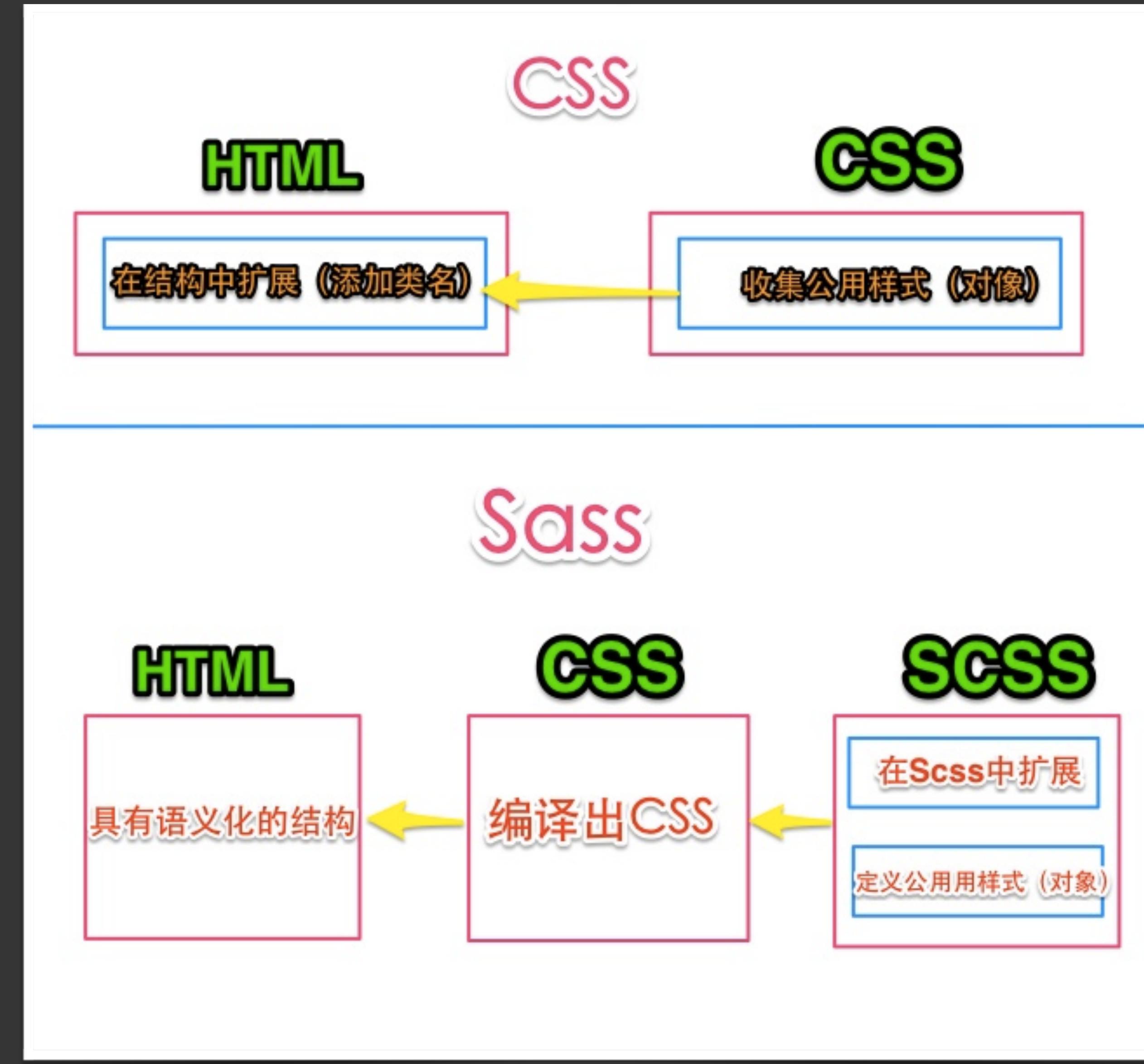
OOCSS & Sass

```
@mixin button-variant($color, $background, $border) {  
  color: $color;  
  background-color: $background;  
  border-color: $border;  
  
  &:hover, &:focus, &.focus, &:active, &.active{  
    color: $color;  
    background-color: darken($background, 10%);  
    border-color: darken($border, 12%);  
  }  
}  
  
@mixin button-size($padding-vertical, $padding-horizontal, $font-size, $line-height, $border-radius) {  
  padding: $padding-vertical $padding-horizontal;  
  font-size: $font-size;  
  line-height: $line-height;  
  border-radius: $border-radius;  
}
```

OOCSS & Sass

```
.btn-default {  
  @include button-variant($btn-default-color, $btn-default-bg, $btn-default-border);  
}  
// Success appears as green  
.btn-success {  
  @include button-variant($btn-success-color, $btn-success-bg, $btn-success-border);  
}  
// Info appears as blue-green  
.btn-info {  
  @include button-variant($btn-info-color, $btn-info-bg, $btn-info-border);  
}  
// Warning appears as orange  
.btn-warning {  
  @include button-variant($btn-warning-color, $btn-warning-bg, $btn-warning-border);  
}  
// Danger and error appear as red  
.btn-danger {  
  @include button-variant($btn-danger-color, $btn-danger-bg, $btn-danger-border);  
}
```

OOCSS + Sass



编写Sass技巧

管理好文件结构

有效使用Sass变量

减少对混合宏依赖

拥抱占位符

用函数做更多的事情

合理嵌套

保持Sass简单

总结

CSS太脆弱了，难以维护

OOCSS改变了，使用多个类

Sass扩展OOCSS的不足

THANK YOU

