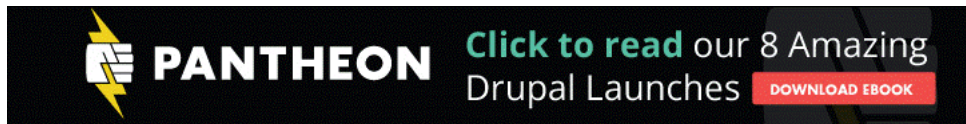


Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour ×

## In SASS, how to avoid space before interpolation when creating a selector?



### What I've already done

I use sprite images and have a SASS rule that goes like this to assign a bunch of properties (a mixin actually) to all these classes:

```
.success-16, .error-16, .warning-16, .info-16,
.user-16, .plus-16, .logout-16, .star-16, .question-16,
.document-16, .save-16, .cancel-16, .email-16, .search-16
{
  @include sprite(16, 'img/Sprites16.png');
  // This @include inserts a bunch of properties common to all classes
}
```

And then I have another `@include` with arguments to insert the `background-position` property calculated for each class. Something like:

```
.success-16
{
  @include spritePosition(16, 2, 1);
}

.error-16
{
  @include spritePosition(16, 3, 1);
}

// ... and so on.
```

This code outputs a CSS like this:

```
.success-16, .error-16, .warning-16, .info-16,
.user-16, .plus-16, .logout-16, .star-16, .question-16,
.document-16, .save-16, .cancel-16, .email-16, .search-16
{
  display: inline-block;
  width: 16px;
  height: 16px;
  background: url(img/Sprites16.png) no-repeat 0 0;
  /* ...and more rules */
}

.success-16 { background-position: -16px 0px; }

.error-16 { background-position: -32px 0px; }
```

### What I want to do

Now, the new requirement is when any of these classes is applied to an `<img>` element I need to overwrite the `margin-right` property, so I thought of saving all my selectors in a string variable:

```
// This variable holds all my selectors
```

```
$All_selectors: '.success-16, .error-16, .warning-16, .info-16,
.user-16, .plus-16, .logout-16, .star-16, .question-16,
.document-16, .save-16, .cancel-16, .email-16, .search-16';
```

and then use interpolation to create the new rule referencing the parent selector:

```
img
{
  &#{$All_selectors}
  {
    margin-right: 0;
  }
}
```

This code produces the following CSS output:

```
img.success-16, img.error-16, img.warning-16, img.info-16, img.user-16,
img.plus-16, img.logout-16, img.star-16, img.question-16, img.document-16,
img.save-16, img.cancel-16, img.email-16, img.search-16
{
  margin-right: 0;
}
```

This turns out to work, but only for the first selector in the variable string. **Notice how the interpolation adds a space between the `img` and the class name for the rest of the selectors.**

I have already tried to remove the spaces in the `$All_Selectors` string variable, but this does nothing, the spaces appear there anyway.

## The question

As you know, the space in between makes the rule to fail, as it is interpreted as the class being a descendant of the `img` element, and not part of it.

Does anyone has any idea how to avoid interpolation to add this extra space that messes up my code?.  
Thanks.

## --- UPDATE ---

### The solution

Thanks to [Russ Ferri](#) answer I came up with a solution using [placeholder selectors](#):

```
img%iconsPH // %iconsPH is just a placeholder that will be replaced with the selectors
{
  margin-right: 0;
}
```

The previous code outputs nothing, until I use it in an `@extend` instruction:

```
#{$ALL_selectors}
{
  @extend %iconsPH;
}
```

This code outputs what I was looking for:

```
img.success-16, img.error-16, img.warning-16, img.info-16, img.user-16,
img.plus-16, img.logout-16, img.star-16, img.question-16, img.document-16,
img.save-16, img.cancel-16, img.email-16, img.search-16
{
  margin-right: 0;
}
```

Everything works now. :)

## More info

Just in case you were wondering what my mixins do, here they are. My sprite image `$file` is a grid of icons, and I use rows and columns in it to calculate each image position. I use the same mixin for icon grids of a different size.

```
// 'sprite' mixin groups all the properties to be applied to a sprite class.
@mixin sprite($size, $file)
```

```
{
  display: inline-block;
  vertical-align: text-bottom;
  width: $size + px;
  height: $size + px;
  overflow: hidden;
  text-indent: $size * 2 + px;
  white-space: nowrap;
  background: url($file) no-repeat 0 0;
  position: relative;
  margin-right: 5px;
}

// 'spritePosition' adds the background-position property with calculated values upon t
@mixin spritePosition($size, $col, $row)
{
  background-position: -($size * ($col - 1)) + px + ' ' + -($size * ($row - 1)) + px;
}
```

I used [Mindscape Web Workbench VS extension](#) to compile all my SCSS files.

[css](#) [css-selectors](#) [sass](#) [interpolation](#) [css-sprites](#)

edited Dec 16 '13 at 17:35

asked Dec 12 '13 at 0:02



[alonso.torres](#)  
412 5 15

[add comment](#)

## 1 Answer

This is a good use case for extending a base [placeholder selector](#).

```
// Base rules for all sprites...
%sprite-16 {
  @include sprite(16, 'img/Sprites16.png');
}

img%sprite-16 {
  margin-right: 0;
}

// Rules for individual sprites...
.success-16 {
  @extend %sprite-16;
  @include spritePosition(16, 2, 1);
}

.error-16 {
  @extend %sprite-16;
  @include spritePosition(16, 3, 1);
}

// ...
```

[example](#)

answered Dec 12 '13 at 1:13



[Russ Ferri](#)  
1,333 9 13

---

I never heard of these placeholder selectors. I'll give them a try and let you know how it goes. – [alonso.torres](#) Dec 12 '13 at 22:25

---

It worked! However, I used the string variable to generate all classes at once, instead of extending each one individually. I'll update my question with your solution. Thank you! – [alonso.torres](#) Dec 13 '13 at 23:57

[add comment](#)

---

Not the answer you're looking for? Browse other questions tagged [css](#) [css-selectors](#)

[sass](#) [interpolation](#) [css-sprites](#) or [ask your own question](#).

