
Typi



Table of contents

- Intro
- Installation
- Configuration
- Using Typi ([base](#) font-map)
- Using Typi (other font-maps)
- Automatically creating classes with Typi
- Sizing in [em](#)
- Em-based media queries
- Vertical Rhythm
- Vertical Rhythm with [em](#)
- Contributing
- Changelog

Intro

Typi does two things for you incredibly well.

1. Typi **helps you write font-size and line-height declarations** at multiple breakpoints without breaking a sweat.
2. Typi **helps you calculate vertical rhythm** without having to do the math yourself.

Here's a quickie example for both points.

For point 1:

```
1 // Sass input
2 h1 { @include typi('h1'); }
```

```
1 /* CSS output */
2 h1 {
3   font-size: 1.5rem;
4   line-height: 1.3;
5 }
6
7 @media all and (min-width: 600px) {
8   h1 {
```

```
9     font-size: 2.369rem;
10    line-height: 1.2;
11  }
12 }
```

For point 2:

```
1 h1 {
2   margin-top: vr(1);
3 }
```

```
1 h1 {
2   margin-top: 1.4rem;
3 }
```

Installation

You can install Typi in four ways:

1. Bower: `bower install typi --save`
2. npm: `npm install typi --save-dev`
3. diamond: `diamond install typi`
4. manual install (<https://github.com/zellwk/typi/archive/master.zip>)

Typi with ruby gems If you want to install Typi with Ruby, check out Pete's repo for installation instructions. (I think you can use v2.3.0. Not sure about v3)

Once you've downloaded Typi, include it in your project with:

```
1 // Change `path-to-typi` with the correct path!
2 @import 'path-to-typi/scss/typi';
```

If you are using diamond, it can be imported with:

```
1 @import '~typi';
```

Configuration

You need to configure two Sass maps:

1. `$breakpoints` – holds breakpoint values
2. `$typi` – holds all your typography config

\$breakpoints map

The `$breakpoints` map is a series of `key: value` pairs that tell Typi what media queries to create for each font-size and line-height property you intend to write. It looks like this:

```
1 $breakpoints: (  
2   small: 600px,  
3   large: 1200px  
4 );
```

Feel free to leave breakpoint values in pixels if you intend to use a breakpoint library that's compatible with Typi (more on that later). Otherwise, I recommend you convert these values into em.

\$typi map

The `$typi` map is a **storage of different font maps** that contain information about breakpoints to create and the font-sizes and line-heights that should be written at that breakpoint.

The bare minimum version looks like this:

```
1 $typi: (  
2   base: (  
3     null: (16px, 1.4),  
4     small: (18px),  
5     large: (20px)  
6   )  
7   // Other font maps here  
8 );
```

The first font-map in `$typi` should always be the `base` font-map. This tells Typi to output the correct font-sizes and line-height in the `html` selector. Here's what it reads:

- **null** key: Create font-size of 16px and line-height of 1.4 without breakpoints
- **small** key: At 600px, change font-size to 18px
- **large** key: At 1200px, change font-size to 20px

You can also create other font-maps, but we'll talk about them later to make things easier to understand. Let's see how to use this `base` font-map first.

Using Typi (base font-map)

Typi uses the `base` font-map to create font-size and line-height values for the `html` selector. You tell Typi to create these values by using the `typi-init` mixin.

```
1 @include typi-init;
```

You should get the following CSS. Notice how pixel values get converted into percentage values when you use `typi-init`.

```
1 html {
2   font-size: 100%; /* this means 16px */
3   line-height: 1.4;
4 }
5
6 @media all and (min-width: 600px) {
7   html {
8     font-size: 112.5%; /* this means 18px */
9   }
10 }
11
12 @media all and (min-width: 1200px) {
13   html {
14     font-size: 125%; /* this means 20px */
15   }
16 }
```

Using Typi (other font-maps)

Typi allows you to create other font-size and line-height and media query combinations by creating another font-map, like this:

```
1 $typi: (
2   // base font-map,
3   h1: (
4     null: (24px, 1.3),
5     small: (2.369em, 1.2),
6   )
7 );
```

In the code above, we created a `h1` font-map with that:

1. Creates a font-size with a value of 24px (written in rem) and a line-height of 1.3 without break-points
2. Changes font-size to a value of 2.369em (written in rem) and line-height to 1.2 at a minimum width of 600px.

You can use this `h1` font-map with the `typi` mixin once you've created it, like this:

```
1 h1 {
2   @include typi('h1');
```

```
3 }
```

The CSS produced by Typi is (notice how font-sizes gets converted into rem):

```
1 h1 {
2   font-size: 1.5rem;
3   line-height: 1.3;
4 }
5
6 @media all and (min-width: 600px) {
7   h1 {
8     font-size: 2.369rem;
9     line-height: 1.2;
10  }
11 }
```

Since Typi works with em values, you can also use Modular Scale easily with Typi like this:

```
1 $typi: (
2   // base font-map,
3   h1: (
4     null: (24px, 1.3),
5     small: (ms(3), 1.2),
6   )
7 );
```

(Note: Make sure to include the modular-scale sass library before doing this. Typi works with Modular Scale Version 2+-. DO NOT install version 3 or Typi will fail).

Automatically creating classes with Typi

Typi can help you create classes automatically if you use the `typi-create-classes` mixin. It extracts the keys present in your `$typi` map and calls `@include typi` on each individual key.

Read this article to see why you might love this feature.

```
1 // Input
2 $typi: (
3   base: (null: (16px, 1.4)),
4   h1: (null: (24px, 1.3))
5 );
6
7 @include typi-create-classes;
```

```
1 /* Output */
2 .base {
3   font-size: 100%;
4   line-height: 1.4;
```

```
5 }
6
7 .h1 {
8   font-size: 1.5rem;
9   line-height: 1.3;
10 }
```

Sizing in em

Although I highly recommend the use of `rem`, there are instances where you want to use `em` over `rem`. If this happens, all you need to do is tell Typi you want to create sizes in `em` by stating `$rem: false`, like this:

```
1 @include typi('h1', $rem: false);
```

And Typi automatically writes sizes in the `em` unit.

```
1 h1 {
2   font-size: 1.5em;
3   line-height: 1.3;
4 }
5
6 @media all and (min-width: 600px) {
7   h1 {
8     font-size: 2.369em;
9     line-height: 1.2;
10  }
11 }
```

Em-based media queries

You should use `em` values for media queries, unless you decide to use a breakpoint library that's compatible with Typi. Typi will automatically convert all pixel values to `em` if you use such a library.

At this moment, Typi supports the use of three breakpoint libraries—Mappy breakpoints, Breakpoint Sass and Sass MQ. You tell Typi about the existence of these libraries with:

```
1 // using Mappy Breakpoint
2 $typi-breakpoint: mappy-bp;
3
4 // using Breakpoint Sass
5 $typi-breakpoint: breakpoint;
6
7 // using Sass MQ
8 $mq-breakpoints: (
9   small: 400px,
```

```
10   med: 600px,
11   large: 800px,
12 );
13
14 $typi-breakpoint: mq;
```

Then, Typi will do it's job and convert pixels to em automatically:

```
1  h1 {
2    font-size: 1.5em;
3    line-height: 1.3;
4  }
5
6  @media all and (min-width: 37.5em) {
7    h1 {
8      font-size: 2.369em;
9      line-height: 1.2;
10   }
11 }
```

Vertical Rhythm

Typi gives you a `vr()` function to count baselines without requiring you to do complicated calculations yourself. It looks like this:

```
1  h1 {
2    margin-top: vr(1); // 1 baseline
3    margin-bottom: vr(2); // 2 baselines
4  }
```

Typi uses the **null** key in your base font-map to calculate the Vertical Rhythm. Typi then uses the `line-height` value (1.4) to calculate the Vertical Rhythm.

```
1  $typi: (
2    base: (
3      null: (16px, 1.4),
4      small: (18px),
5      large: (20px)
6    )
7    // Other font maps here
8  );
```

In this case, `1 baseline = 16px * 1.4` (converted into rem).

```
1  h1 {
2    margin-top: 1.4rem; /* 1 baseline */
3    margin-bottom: 2.8rem; /* 2 baselines */
4  }
```

Vertical Rhythm with em units

Typi gives you the ability to write Vertical Rhythms in em instead of rem whenever you need to. To do so, you include the font-size as a second parameter and Typi will automatically do the rest. This font-size parameter can either be in pixels or em:

```
1 h1 {  
2   // This is equivalent to vr(3) if font-size is 1.5em  
3   margin: vr(3, 1.5em);  
4 }
```

Additional features

More features are coming to Typi. However, I haven't had the time to work them out (or write about them) yet. I'll let you know when I manage to make Typi even better than it is now!

Contributing

Just one for now: Make sure the tests before you submit a pull request.

Steps:

1. Clone the repo
2. Install dependencies with `bower install` & `npm install`
3. Run `gulp` to start tests

Changelog

v3.2

- Added support for Sass MQ

v3.1

- Added ability to write classes automatically with `typi-create-classes`.

v3.0

- Changed `$typi` map. NOTE: BREAKING CHANGE.

v2.2.0

- Added the ability to change Modular Scale within media queries easily

v2.1.1

- Added typi-ms() function
- Bugfix for integration with mappy-bp

v2.0.0

- Changed typi() output to rem by default

v1.1.0

- Added ability to calculate vertical rhythms in em and rem.

:)