
Initializr: With Great Templates Comes Great Responsivity!

Posted on [December 20, 2011](#)

4

■ [Tweet](#) 464



Today I'm proud to announce that a "Responsive template" is now available on Initializr! It will help you dealing with all the various devices used to display your site. Mobiles, tablets, netbooks, laptops, desktop monitors, HUGE desktop monitors... Making your site work correctly on all these devices is a real nightmare. Well guess what? This nightmare is... absolutely not over! But as least, the boilerplate code of a responsive layout is now here to make your life easier.

Let's see how it works and what best practices are introduced with this new template.

Little reminder about responsive web design

For those who have been living in a cave for the last few months (understand: for those who have a life and don't spend days reading tweets and blogs) you probably need a short update about what a "responsive" design is.

The concept, introduced by [Ethan Marcotte](#), is pretty simple: the page layout should adapt depending on the screen size so it's always convenient and easily readable for

the user. On mobile for instance, it's more comfortable to get a minimalistic one-column vertical view than getting a 2 or 3 columns view – which is perfect for desktops. What about tablets? Maybe we can also adapt the layout to fit their medium size screen too? Since we can't know the screen size of our users' devices, well it should just work everywhere.



To design a responsive layout, we have to forget about the usual static 960px width layouts and go fluid with percentage widths and media queries.

A fluid layout with percentage widths

The first thing our page needs is a “fluid” layout. A layout is called fluid when the text adapts its container size and when proportions are preserved when the window is resized. To get this working we can't have any hard coded static value for widths, horizontal padding nor horizontal margins. We can basically only use percentages:

```
1  #this-is-a-fluid-container{
2      width:50%;
3      padding: 20px 10%;
4      margin:0 10%;
5  }
```

If you strictly follow this approach, your layout will scale perfectly from a small mobile screen to a huge desktop monitor. It's already pretty cool, but you still have the same layout on every device, which is far from ideal! Don't worry, now comes the fun part with media-queries to solve this problem.

Mobile-first approach media queries

Media queries are a cool way to specify different styles depending on the device used (by aiming particular screen resolutions or orientations for example). The mobile-first approach is the equivalent of progressive enhancement for media-queries. The principle is to start with the lowest resolution devices and then enhance the experience for bigger screens.

The “mobile” view

Here is the very simple mobile display of the responsive template. It doesn't take into account any media query yet:



You can see that an **aside** orange block is located at the bottom of the page:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor.

aside

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices.

footer

Note: Remember that an **aside** element is absolutely not meant to just be on the right or the left of an other element. Its semantic definition is that its content should be related to the main content, so its perfectly fine to have it under the article.

The “intermediate” view

If we display this page on a larger device (like a tablet or a wide smartphone), the vertical menu gets a little bit annoying and ugly because its buttons are stretched too much. We can use a media query to display them horizontally for screens larger than 480px like this:

```
1 | @media only screen and (min-width: 480px) {  
2 |
```

```

3     nav a{
4         float:left;
5         width:27%;
6         margin:0 1.7%;
7         padding:25px 2%;
8         margin-bottom:0;
9     }
10 }

```

Which has the following result:



The “wide” view

We now have a more convenient menu for tablets. Awesome! But what if we open this page on a laptop now? They have bigger screens so the vertical one column layout might not be the best fit for these devices. We would rather prefer a 2 columns layout, so we can move the “aside” block and the menu to the right of the page, and keep the content on the left. Let’s use a second media query to do that for resolutions higher than 768px:

```

1 | @media only screen and (min-width: 768px) {

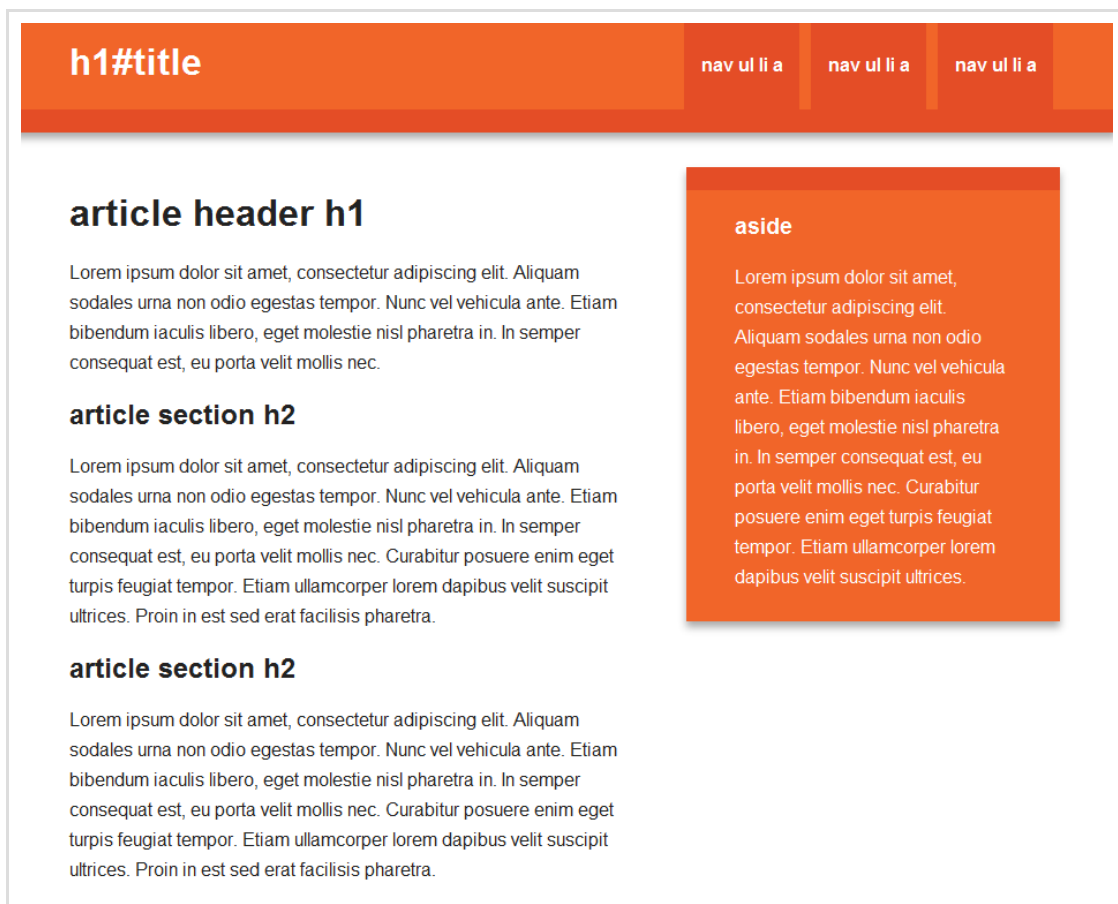
```

```

2
3     #title{
4         float:left;
5     }
6
7     nav{
8         float:right;
9         width:38%;
10    }
11
12    #main article{
13        float:left;
14        width:57%;
15    }
16
17    #main aside{
18        float:right;
19        width:38%;
20    }
21 }

```

This is what we get on a screen of approximately 900 pixels wide:



We now have a nice 2-columns “desktop view”. You see, it wasn’t that hard to start from the mobile version first! I admit it might not be intuitive at first, but you get used to it pretty fast, and it’s definitely cleaner.

Note: You may have noticed some new shadows on this one. Fancy CSS3 effects, like shadows, are CPU consuming so they’re not very suitable for

mobiles and low-CPU devices in general. This layout will probably be displayed on a more powerful device like a laptop or a desktop, so we can suppose they're good enough to handle these effects.

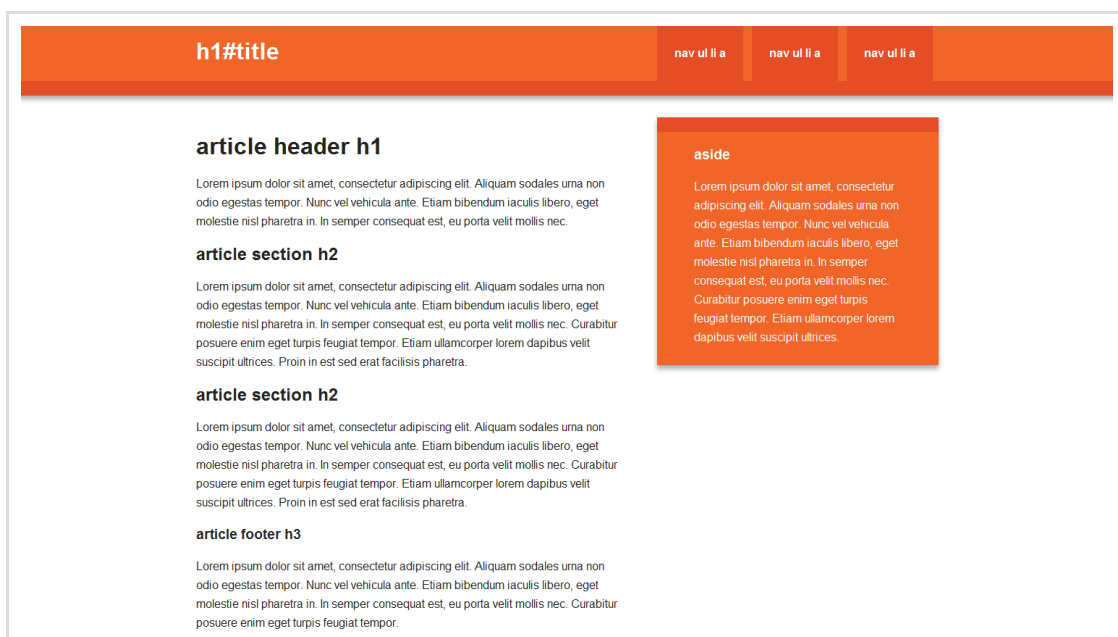
The “maximal width” view

Last but not least, we want a decent display on wide screens. If we leave it as it is now and open the page on a wide screen, a 5 line paragraph would be turned into an awful huge single line, which is terrible for readability. We need a maximal width limit to avoid this on wide screens. This limit can be anything: 960px, 1140px, 2000px, whatever, just choose one that make sense for your site.

In our case I picked 1140px, which suits well 1280 pixels wide screens, the most common resolution. An old school wrapper with fixed width and auto margins does the job perfectly:

```
1 @media only screen and (min-width: 1140px) {
2
3     .wrapper{
4         width:1026px; /* 1140px - 10% for margins */
5         margin:0 auto;
6     }
7 }
```

I've set the wrapper's width to 1026px (1140px – 10%) because on the previous layouts the margins were set to 10%. Therefore if we resize the browser it will move from one layout to another seamlessly. Here is the widest layout we can get:



All right! We're done with media queries!

Christmas presents for nice web developers

Since you've been very nice this year, here are some other presents for you!

1st gift: A Live Demo!

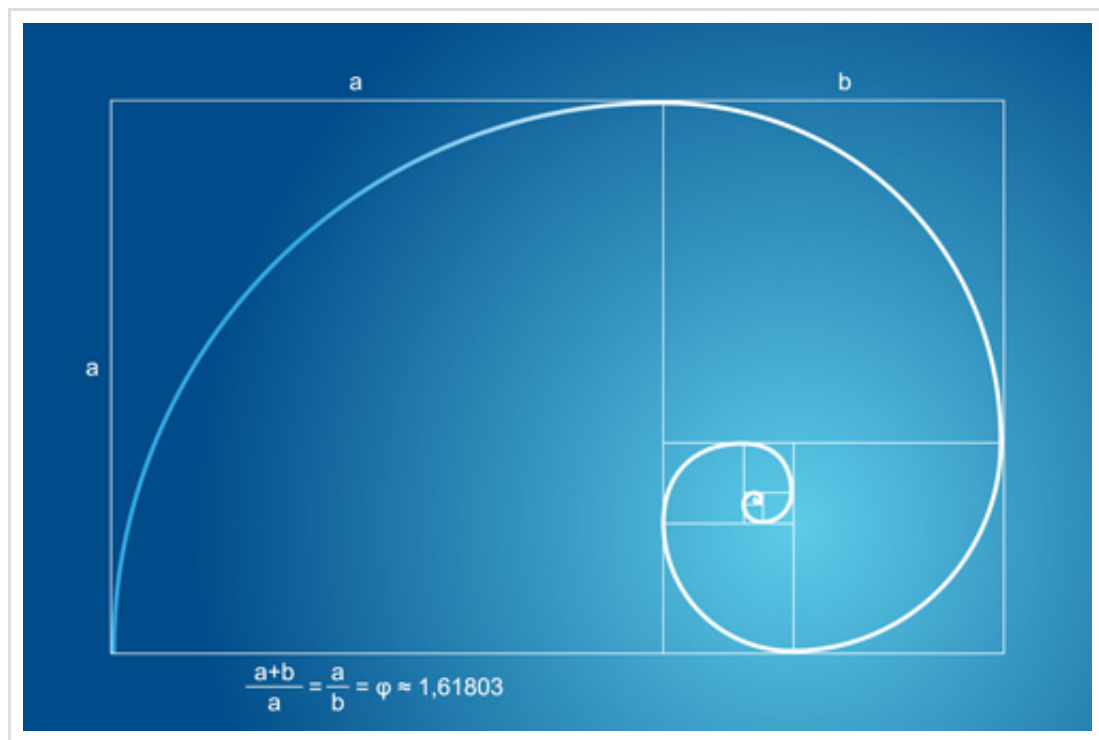
There is now a live demo of the responsive template to see if it works as expected before downloading anything. This is also particularly convenient to see how it runs on your mobile or tablet.

You can find it on initializr.com/try.

2nd gift: The Golden Ratio

This responsive template comes with some awesome golden ratio black magic. You may have heard of it, it's a weird mathematical proportion that our stupid brains seem to like. If you use the golden ratio in your designs, for some reason people will like it more without even understanding why. I've set the proportions of the left and right sections of the desktop view to 62% and 38% (take margins into account) so it always matches the golden ratio as you resize the browser.

Here is also a [golden ratio calculator](#) for you to play with, and an [interesting tutorial](#) about how to apply it to your layout. Feel free to use this black magic or not :)



3rd gift: Readability For The Win

This new template comes with pretty decent styles for a good readability. Remember that readability is absolutely crucial for your users. It will make them actually read what you have to say instead of just looking at pictures, and if this reading is pleasant they will be more eager to subscribe, retweet, buy your stuff or whatever you want them to do. This is a pretty big deal!

Here is some advice to improve readability:

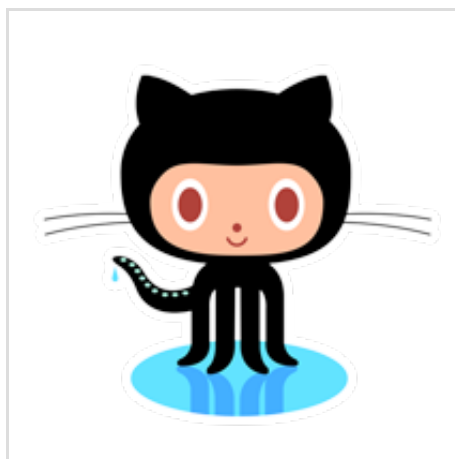
- Use a big font size and an adapted line height (this template uses 16px / 24px).
- A line should have approximately 80 characters (you're all set with the template).
- Avoid paragraphs of more than 5 lines.
- Avoid sections of more than 5 paragraphs in a row.
- Put stuff between paragraphs when you can and when it makes sense so the reader can breath before he starts a new one (charts, pictures, code, decorations...).



4th gift: A Github repository for this responsive template

Some of you might have noticed that Initializr's Github repository is the website's repository, not the template's. So it's pretty useless for you except if you want to know how to generate archives in Java! I've created a new Github repo for this responsive template, so you can fork it and suggest modifications and improvements to it. Way more useful, yay!

Here is the new repository: github.com/verekia/initializr-template



Enjoy!

I hope this new responsive template will make your web developer life easier, so you can focus on your actual business instead of setting up again and again the same boilerplate code. I would love to hear your feedback about this, so feel free to [open](#)

issues on [Github](#) or mention [@initializr](#) or [@verekia](#) on Twitter, I usually reply fast :)

Merry Christmas and Happy New Year!

This entry was posted in [Initializr](#) and tagged [1140px](#), [960px](#), [@media only screen](#), [Android](#), [Best practices](#), [Browser](#), [Christmas](#), [Column](#), [CSS3](#), [demo](#), [Design](#), [Desktop](#), [Fluid](#), [Generator](#), [Github](#), [Golden Ratio](#), [iPad](#), [iPhone](#), [Laptop](#), [Layout](#), [Margin](#), [Media Queries](#), [min-width](#), [Mobile](#), [Mobile-first](#), [Padding](#), [Percentage](#), [Progressive Enhancement](#), [Proportions](#), [Readability](#), [Resize](#), [Resolution](#), [Responsive](#), [Scale](#), [Screen](#), [Size](#), [Smartphone](#), [Tablet](#), [Template](#), [Width](#), [Window](#) by [Jonathan](#). Bookmark the [permalink](#) [<http://verekia.com/initializr/responsive-template>] .

17 THOUGHTS ON “INITIALIZR: WITH GREAT TEMPLATES COMES GREAT RESPONSIVITY!”



Alec

on **December 21, 2011 at 4:10 AM** said:

I think the guidelines you have provided in this article would surely help most of the people. It has been seen that a web template is a convenient website design which is created to use as a basis for fast and high quality website development. Thanks for sharing such useful information.



me

on **December 21, 2011 at 5:40 AM** said:

Thank you!



Rick Roberts

on **December 21, 2011 at 12:03 PM** said:

Excellent work. Thank you! Have you considered em instead of fixed point sizes for type?



Jonathan

on **December 23, 2011 at 4:42 PM** said:

Yes! I will work on this as soon as possible, probably after holidays.
I'll also try to mix nicely the golden ratio with the typographic modular scale.
But feel free to suggest modifications on Github if you know about this!

Happy holidays!



Jose Delgado

on **December 27, 2011 at 7:02 AM** said:

Great stuff. Thanks



Henrijs Šešo

on **January 2, 2012 at 7:51 PM** said:

Well done. Main menu could also turn into default OS dropdown list menu when in narrowest mode. Default OS menus tend to be compact and extremely usable, as they are native elements – fast and optimized. Current solution will turn ugly if 5 or 7 main menu items are present.



Jonathan

on **January 2, 2012 at 7:54 PM** said:

Very relevant Henrijs, thank you!
I just opened the first [issue on Github](#) with your suggestion ;)



Michael D

on **January 2, 2012 at 8:59 PM** said:

Wonderful – thanks, Jonathan!



Austin H

on **January 2, 2012 at 9:31 PM** said:

Fantastic Article! Thanks :)



Joe McCann

on **January 2, 2012 at 9:54 PM** said:

brilliant.



T.J. Barbour

on **January 2, 2012 at 11:35 PM** said:

responsive-template + html5boilerplate == world domination?

<https://github.com/h5bp/html5-boilerplate>

Great intro on media queries



Jonathan

on **January 3, 2012 at 3:09 AM** said:

Initializr is using H5BP already, just download the featured template on initializr.com and you're all set for world domination ;)



Peja

on **January 3, 2012 at 2:25 AM** said:

Great post – thanks for sharing.



Ken Stein

on **January 3, 2012 at 3:00 AM** said:

Awesome! Thanks! Have been using a static left sidebar and now thinking to move it to the right and have it “vanish” for smaller devices. Great starting point saving me that most precious of commodities, TIME!



Alex Kravets

on **January 3, 2012 at 4:01 AM** said:

Great post, extremely useful! Thanks a lot! Happy new year, all!!!



Joss

on **January 3, 2012 at 4:40 AM** said:

Beautiful stuff. Congrats!



Pierre Sim

on **January 3, 2012 at 7:05 AM** said:

Wow, interesting stuff..

It is always nice to hear about new web themes that can cater to mobile users..

Gonna explore this further for our projects.

Thanks for sharing, Verekia! =)