

## RESPONSIVE WEB DESIGN TECHNIQUES

Luminita GIURGIU, Ilie GLIGOREA

“Nicolae Bălcescu” Land Forces Academy, Sibiu, Romania  
luminita.giurgiu.a@gmail.com, iliegligorea@gmail.com

**Abstract:** First introduced by Ethan Marcotte in 2010 in his article *Responsive Web Design*, responsive design concept refers to the ability of a website to adapt to different resolutions and devices on which it is displayed (PCs, laptops, tablets, smartphones, etc.) without undergoing degradation. The site should recognize what device it is displayed on and automatically respond to user preferences. This approach comes to the aid of those who were forced to do a lot of versions of the site, so that it can be accessed from the iPad, Blackberry, Kindle, netbook, iPhone and so on[1], and becomes today a necessity. While using the same code and URL, a responsive website can provide an optimum viewing experience - without resizing, panning or scrolling on most of the devices. Web designers approached approach the new challenge with many and various solutions. To keep up with the transformations emerging in web design, this article intends to discuss the issues applied to the Land Forces Academy's official site.

**Keywords:** responsive, web design

## 1. Introduction

Nowadays, responsive web design is no longer a new technique in tests, but a technical trend. Since 2013 it is the most used, because more and more owners of websites want to offer a pleasant experience to users as they use the mobile Internet, and the phones, tablets and mobile devices find their way in daily life. According to the results of the survey 'Internet life everyday', commissioned by Google and conducted in five countries: Romania, Czech Republic, Slovakia, Hungary and Poland, 67% of the Romanians connected to the Internet use smartphones and mobile phones to connect, and 32% use tablets. Other significant figures are as follows: 61% of the people have a better opinion about brands when they provide a pleasant mobile experience and over 20% of Google searches are conducted from a mobile device. Traffic on mobile devices represents over 50% of the

total traffic on the Internet.

Another survey's results (figure no 1) of the web analytics service StatCounter Global Stats shows, for example, that, in Romania, the using of devices with 360x640 resolution increased with 10% in the period may, 2016 – april, 2017.

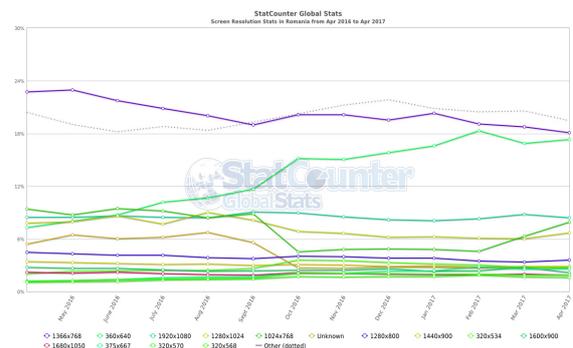


Figure 1 Screen Resolution Stats in Romania, May 2016 to Apr 2017 [2]

So achieving a responsive website is absolutely essential. In the digital age, a last

generation website, compatible with all mobile platforms, has become a necessity. The example below shows the views of the website for tablets, large desktops, smartphones, and small desktops.



Figure 2. Example of a responsive site

different organization of the site in fluid boxes (flexible grids) that slide relative to each other depending on screen resolution, elements with versatile dimensions, flexible images, and the site also having the ability to recognize the dimensions and to choose the optimal display version for a superior navigation experience. It takes three main ingredients to have a responsive site: fluid grids, flexible images/media and media queries[3]. These elements live in the context of a different way of thinking. An example of a responsive website elements are shown in figure no. 3.

## 2. Issues to be considered

First, the responsible design requires a

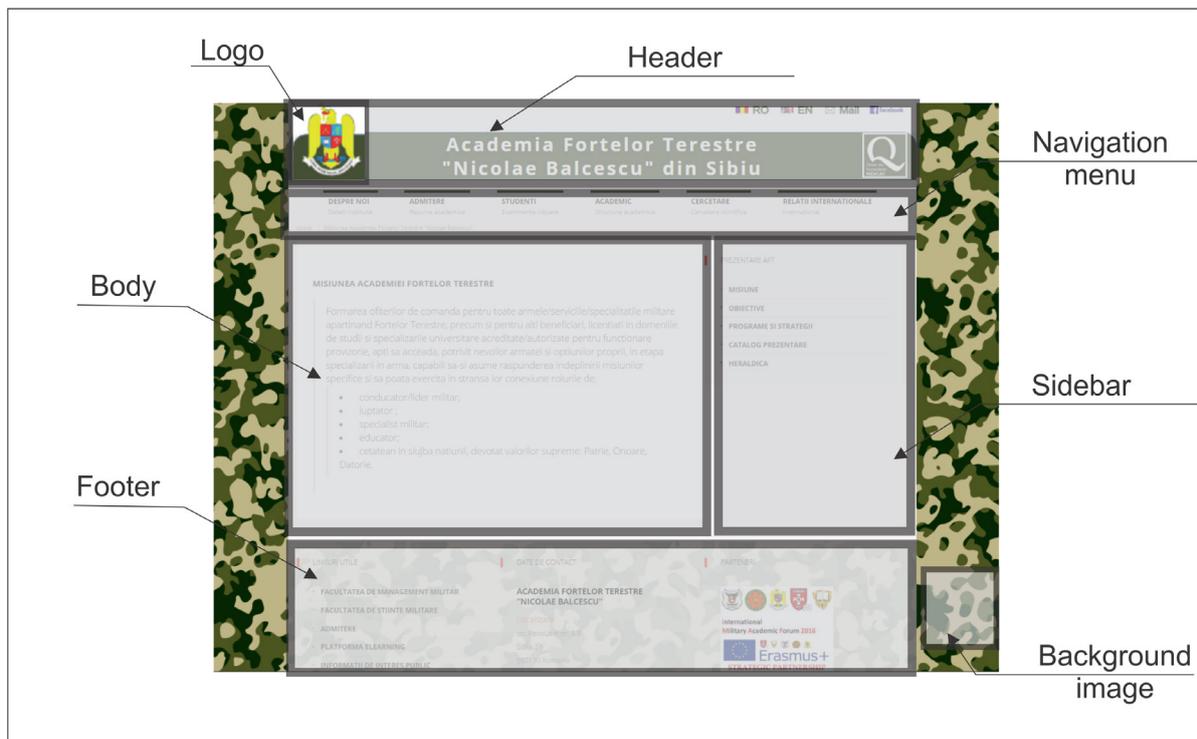


Figure 3. The page elements

### 2.1. Flexible, grid-based layout

First step in responsive web design is the use of so-called fluid grids. Most sites have a fixed width and are centred in the browser, but taking into account the large number of resolutions today, the fluid grids' benefits are hard to ignore. Fluid grids are based on percentage sizes and are very well calculated. It ensures that all elements in the

layout are scaled to each other; no longer measure in pixel dimensions, but in relative units and percentages. This way, when the site is scaled to a very low resolution, all its elements shrink and perfectly align with each other.

The best way to achieve such a site is to make a high resolution design, to take each item measures and to divide the element width by the design width.

## 2.2. Flexible media

The second essential ingredient is represented by thumbnails. They can change the width and height depending on the resolution and the size of the grid. To provide information fast and beautifully images can be saved on the server in sets of three or four dimensions, from which the right image will be loaded later, depending on the size of the screen. There are many techniques that can proportionally resize images; the most used is the `max-width`:

```
img { max-width: 100%; height: auto; }
```

This way, the maximum width of the image will be 100% and if that decreases smaller, so will the image. It is best to use images at maximum size at which they will appear in the site, in order to avoid losing quality. So, fluid images cause the adjustment of the size to the parent block. If the parent block is smaller than the image size then the image is reduced proportionally. For flexible embedded videos, the same technique may apply.

## 2.3. Media Queries (@media)

They are an efficient way to load different CSS styles for different resolutions, to provide the best possible visitor experience. Media Queries are in fact conditions imposed to the CSS style depending on screen resolution. Media query is a CSS technique introduced in CSS3.

The `@media` rule includes a block of CSS properties only if a certain condition is true. Suppose that the browser window is smaller than 320px, the background colour will change to green and the edge of the menu's div will be eliminated:

```
@media only screen and (max-width: 320px) {  
  body { background-color: green; }  
  .menu { margin: 0px; } }
```

Using a series of similar queries, different conditions could be created for higher resolutions: 480px, 600px, 768px, 900px, 1200px. Any number of conditions that are needed could be used. Ideally, the site should look flawless regardless of resolution, but considering its time-consuming action, it is better to think in

advance about the target audience.

## 3. Important steps

Making a new website responsive involves taking more steps that will be described below.

### 3.1. First step - Adding a viewport

A viewport is a meta tag for creating a responsive website, without which the whole process would not be possible. It indicates the browser that the page must be scaled to fit the screen size. There are several ways to apply this viewport, but it is used (the most common) in the `<head>` section of the HTML pages of the site, see Figure 4.

```
</DOCTYPE html>  
<html lang="ro">  
<head>  
  <meta charset="utf-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
</head>  
<title>Academia Fortelor Terestre "Nicolae Balcescu"</title>
```

Figure 4: Adding the viewport

The `width=device-width` sets the width of the page to follow the screen width of the device and `initial-scale=1` sets the initial zoom level when the page is first loaded.

### 3.2. Second step - Setting the breakpoints

Next, scaling the page for mobile devices is done through a number of changes to CSS. It is necessary to set some breakpoints in places where it is needed so that the page elements move suitably to display well on screen.

The dimensions of the page's elements have to be measured in pixels for setting the breakpoints where the content starts to look bad as the screen shrinks. These dimensions can be found using Google Chrome, activating "Inspect element" mode, and then tracing the edge of the appeared section until the page starts not looking well. Breakpoints are set using Media Queries in CSS. So, a breakpoint could be added where certain parts of the design will behave differently on each side of the breakpoint. For example, a media query was used to add a breakpoint at 768px; when the screen gets smaller than 768px, each column should have a width of 100%.

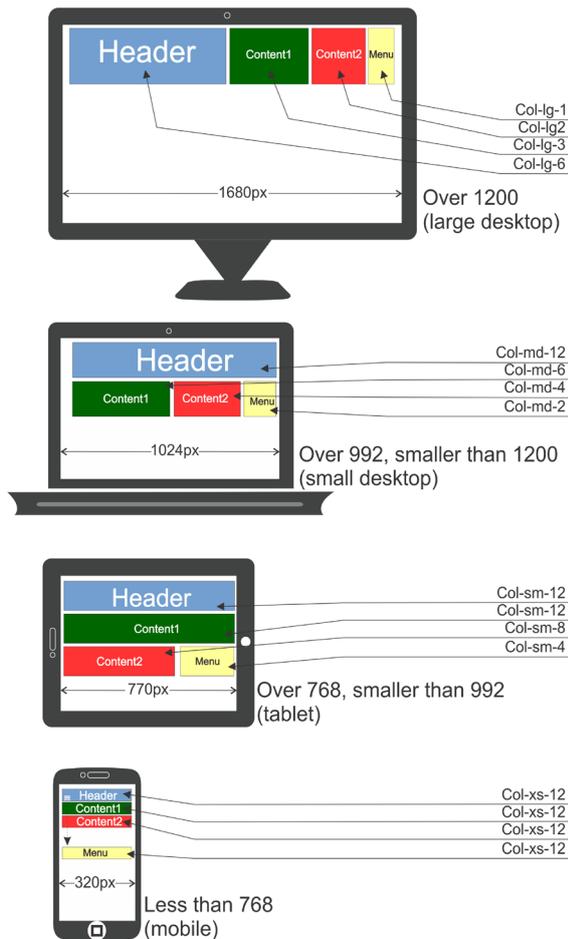


Figure 5. Responsive design

The difference between the classes (see the example in Figure 5) is only the name and it gives the possibility to determine in html how the breakpoints affect the columns:

- For extra small devices Phones (<768px) it would be taken into consideration the class prefix col-xs-\*. Here the place is very limited. The arrangement of the elements must therefore be well thought-out in order to preserve clarity. In most cases, it is clear that the menu from the navigation area will be reduced to an icon. The menu adapts itself to the small space by line breaks and the selection boxes for the target group selection disappear. The options of the selection boxes are moved to a different level and sublevel in the menu.
- For Small devices Tablets ( $\geq 768$ px) it would be taken into consideration the class prefix col-sm-\*. Tablets are generally smaller than monitors, so the page content has to adapt to the new space requirements

- For Medium devices Desktops ( $\geq 992$ px and  $< 1200$ px) it would be taken into consideration class prefix col-md-\*
- For Large devices Desktops ( $\geq 1200$ px) it would be taken into consideration the class prefix col-lg-\*

The symbol “\*” can be replaced by a number. For example, .col-md-4 creates a column four times the size of a .col-md-1 column; .col-xs-12 creates a column twelve times the size of .col-xs-1, etc.

### 3.3. Loading order

For building a responsive site we can take into consideration two ways, depending on the existing content (the approach is different, see figure no 6):

1. First way, if the website already exists, it has no sense to start from zero. It just must be redesigned. In this way we can use **graceful degradation**, where the detailed site for large screen has to gradually stretch down. The smaller the screen, the more elements of the page need to be omitted, condensed or remapped to keep the site clear and easy to use.

2. For a new website, started from zero, the starting concept is mobile first, called **progressive enhancement**. That means that a page is created first for the smallest and finally for large screen, because the better a page is reduced to the absolutely essential, the more clear and useful it is on small end devices. This mobile-focused view of the site is then progressively enhanced to the more sophisticated tablet view and finally to the large desktop version with all the information, images, animations and extras that appear also on the smaller display.

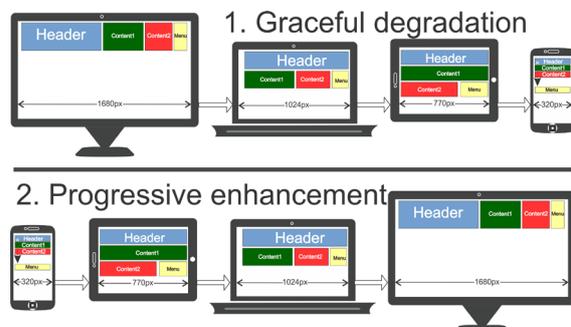


Figure 6. Graceful degradation and Progressive enhancement

### 3.4. Third step – Reducing text size

On the narrow viewport, when the site is displayed on small screens, the text display space is not big enough for formatting settings (heading sites, bold sites, lists, tables, etc.) to display correctly. On the other hand, for a larger viewport, although earlier the text was set to focus and to not display in very long rows, it has to be considered that the visitor is within walking distance and there is a risk that the text font size 12 or 14 cannot be read. Therefore, a final setting to be done in CSS refers to matching the size of the text on the screen that is displayed.

For example, here is how to increase the size of the headers for each section of content while maintaining their size as a maximum 5% of the width:

```
#headline
{padding : 20px 5%}
```

### 3.5. Fourth step – Adjusting elements

When it comes to manually modifying a site to be responsive, it is necessary to make a series of fine settings after following the steps above. This must be done especially in areas where there are visual elements

(photo galleries, video) surrounded by text or tables. So, the adjustment would consist of the following:

- Adapting form elements around their target visual elements;
- Repositioning of videos and images;
- Adapting forms for small screens;
- Extending tables for large screens.

These settings involves making some additions to the CSS code, taking into account the calibration of the visual elements' edge and scaling them to the width of the container. Making these settings depends on the complexity of each site.

## 4 Responsive utility classes

For a site to be responsive on different devices it is recommended to use bootstrap responsive utility classes. With these we can make the content hidden or visible, depending on the device (with the help of media query). Below is a table of the available classes and their effect on a given media query layout (labelled by device)[4].

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
<b>.visible-xs-*</b>	Visible	Hidden	Hidden	Hidden
<b>.visible-sm-*</b>	Hidden	Visible	Hidden	Hidden
<b>.visible-md-*</b>	Hidden	Hidden	Visible	Hidden
<b>.visible-lg-*</b>	Hidden	Hidden	Hidden	Visible
<b>.hidden-xs</b>	Hidden	Visible	Visible	Visible
<b>.hidden-sm</b>	Visible	Hidden	Visible	Visible
<b>.hidden-md</b>	Visible	Visible	Hidden	Visible
<b>.hidden-lg</b>	Visible	Visible	Visible	Hidden

Table 1. Utility classes

Starting with bootstrap v3.2.0, the classes *visible-xs*, *visible-sm*, *visible-md*, *visible-lg* are deprecated, but they still exist.

For example, if we have a class *showLgMd* in the .css file:

```
.showLgMd
{background-color: #396da6;
height: 90px;
margin: 5px;
text-align: left;
font-size: 12px;
text-transform: uppercase;}
```

and we want to display the content only on devices with medium and large devices, the syntax is as following:

```
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 visible-lg visible-md">
<div class=" showLgMd ">Show only devices bigger than 992px – example 1
</div>
</div>
```

Depending on the screen size, the medium or large column will be visible and small and extra small columns will not be visible.

We can obtain the same results by hiding small and extra small columns:

```
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 hidden-sm hidden-xs">  
<div class=" showLgMd ">Show only  
devices bigger than 992px – example 2  
</div>  
</div>
```

In this way we can have a lot of combinations to display the content across all the devices.

## 5. Conclusions

As site design in a user-sensitive manner is becoming a necessity lately, it is important to know that there is no universal right solution for anything, not even for web design. Although it can be an attractive functionality for the user, it can prove to be a disadvantage if it is not used properly. Regardless of the variant, it must be taken into account that the main current trend in

web design is to create the site starting from the screen of mobile devices and not from the desktop screens. If so far we were talking about websites that could later be adapted to mobile devices, in 2017 the balance changed: websites are made directly for mobile devices, considering also how they will be displayed on desktops. The importance of the presentation level is justified by engaging in user interaction. It is what the end user sees after all the app processing: the presentation must be up to the complexity of the application. It has to be simple and intuitive, even if the system is complex because the user does not care about how loaded the system is or how elaborated its structure is; his/her desire is to navigate the application as directly and quickly as possible.

## References

- [1] <https://alistapart.com/article/responsive-web-design>
- [2] <http://gs.statcounter.com/screen-resolution-stats/all/romania>.
- [3] Marcotte, E., *Responsive Web Design, Second Edition*, Publisher: A Book Apart, 2015
- [4] <http://getbootstrap.com/css/>