3

BOOTSTRAP IN PRACTICE

WILLIAM GHELFI

2. YOUR FIRST PAGE

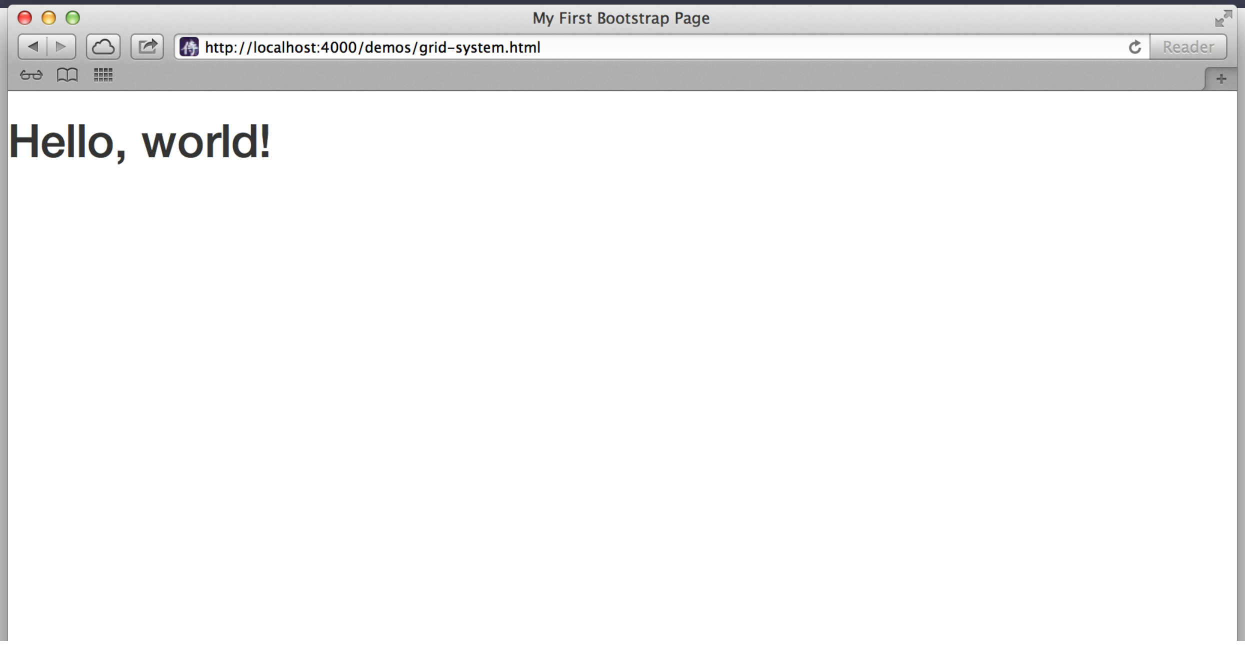
Section 1

BABY STEPS

It's super easy to put together a basic page with Bootstrap.

Just fire up your editor and type in this markup:

```
<!DOCTYPE html>
<html>
 <head>
  <title>Bootstrap in practice: the grid system</title>
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <meta charset="utf-8" />
   <!-- Bootstrap CSS -->
   <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" />
 </head>
 <body>
  <h1>Hello, world!</h1>
 </body>
</html>
```



Hello, world!

What have we done

This is it. There are just two requirements for a basic Bootstrap page:

- HTML5: achieved declaring a `<!DOCTYPE html>`
- Bootstrap CSS: gently hosted by NetDNA

Take a look at the page in a browser, and you'll see a lonely – but stylish – `Hello, world!`.

Heads up!

You'd better serve these pages locally.

If you have Ruby, your best bet is to `gem install serve` and then launch `serve .` from the directory where your HTML file resides.

Then point your browser to **http://localhost:4000/** and you are golden.

If you can't or don't want to serve your pages locally, then please change every protocol relative URL in the sources to its more classic variant, or your browser may not be able to find those resources.

Example:

change `//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css` to `http://netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css`.

Section 2

ADDING SOME BEEF

You are going to see the grid system in action very soon, but first you need to beef up your page with some random content.

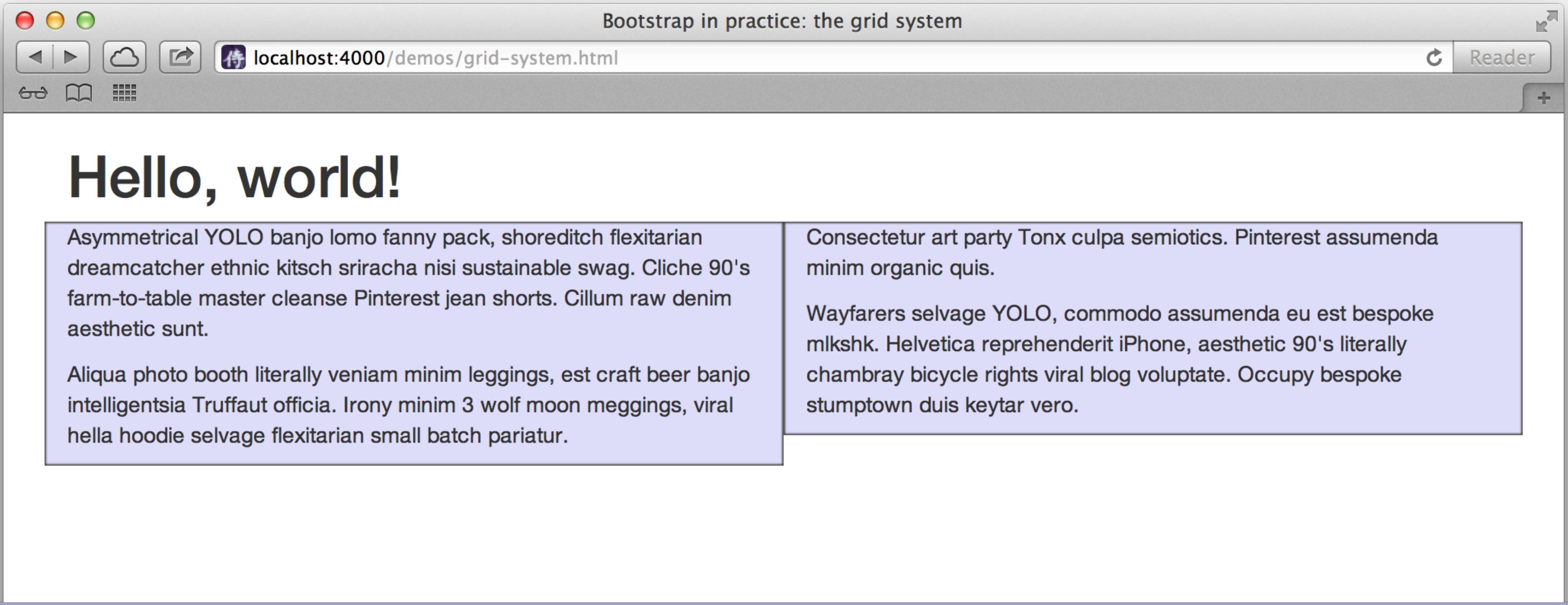
I'll take mine from **hipsteripsum.me**, but the classic **www.lipsum.com** or any editor plugin will be fine as well.

```
<!DOCTYPE html>
<html>
 <head>
  <title>Bootstrap in practice: the grid system</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="utf-8" />
  <!-- Bootstrap CSS -->
  <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" />
 </head>
 <body>
  <h1>Hello, world!</h1>

  <p>Asymmetrical YOLO banjo lomo fanny pack, shoreditch flexitarian dreamcatcher ethnic kitsch sriracha nisi sustainable swag. Cliche 90's farm-to-table master cleanse Pinterest jean shorts. Cillum raw denim aesthetic sunt.</p>

  <p>Aliqua photo booth literally veniam minim leggings, est craft beer banjo intelligentsia Truffaut officia. Irony minim 3 wolf moon meggings, viral hella hoodie selvage flexitarian small batch pariatur.</p>

  <p>Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.</p>

  <p>Wayfarers selvage YOLO, commodo assumenda eu est bespoke mlkshk. Helvetica reprehenderit iPhone, aesthetic 90's literally chambray bicycle rights viral blog voluptate. Occupy bespoke stumptown duis keytar vero.</p>

 </body>
</html>
```

Section 3

HERE COMES THE GRID

Let's start with a very simple layout: two columns, two paragraphs per column.


Bootstrap in practice: the grid system
localhost:4000/demos/grid-system.html
Reader

Hello, world!

Asymmetrical YOLO banjo lomo fanny pack, shoreditch flexitarian dreamcatcher ethnic kitsch sriracha nisi sustainable swag. Cliche 90's farm-to-table master cleanse Pinterest jean shorts. Cillum raw denim aesthetic sunt.

Aliqua photo booth literally veniam minim leggings, est craft beer banjo intelligentsia Truffaut officia. Irony minim 3 wolf moon meggings, viral hella hoodie selvage flexitarian small batch pariatur.

Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.

Wayfarers selvage YOLO, commodo assumenda eu est bespoke mlkshk. Helvetica reprehenderit iPhone, aesthetic 90's literally chambray bicycle rights viral blog voluptate. Occupy bespoke stumptown duis keytar vero.

```html
<!DOCTYPE html>
<html>
 <head>
  <title>Bootstrap in practice: the grid system</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="utf-8" />
  <!-- Bootstrap CSS -->
  <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" />
  <!-- To better visualize the columns -->
  <style>
   .row > div {
    background-color: #dedef8;
    box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;
   }
   </style>
 </head>
 <body>
  <div class="container">
   <h1>Hello, world!</h1>

   <div class="row">

    <div class="col-md-6">
     <p>Asymmetrical YOLO banjo lomo fanny pack, shoreditch flexitarian dreamcatcher ethnic kitsch sriracha nisi sustainable swag. Cliche 90's farm-to-table master cleanse Pinterest jean shorts. Cillum raw denim aesthetic sunt.</p>

     <p>Aliqua photo booth literally veniam minim leggings, est craft beer banjo intelligentsia Truffaut officia. Irony minim 3 wolf moon meggings, viral hella hoodie selvage flexitarian small batch pariatur.</p>
    </div>

    <div class="col-md-6">
     <p>Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.</p>

     <p>Wayfarers selvage YOLO, commodo assumenda eu est bespoke mlkshk. Helvetica reprehenderit iPhone, aesthetic 90's literally chambray bicycle rights viral blog voluptate. Occupy bespoke stumptown duis keytar vero.</p>
    </div>

   </div>
  </div>
 </body>
</html>
```

The details

First of all, we added a `<div class="container">...</div>` container element to ensure proper centering and maximum width for the layout. That's your playground.

Once you have a container in place, you just have to start thinking in terms of rows `<div class="row">...</div>`, and columns inside the rows `<div class="col-md-6">...</div>`.

Keep in mind that **every row in the grid is made up of 12 units**, and you can define the desired size of your columns using those units.

In the previous example you had two columns, each being 6 units wide: `col-md-6`. As you know, `6 + 6 = 12`, and that's exactly why you had two equally large columns (`col-md-6`) in a row taking the whole width (`12`) of the container element.

Being brave

Go on, try some more combinations like:

`<div class="col-md-3">...</div> <div class="col-md-9">...</div>`

or :

`<div class="col-md-7">...</div> <div class="col-md-5">...</div>` and take a look at the results in your browser.

Experiment a bit. Just keep in mind that the sum must always be 12.

If you're feeling brave you could also have more than two columns in the same row, adjusting the column classes accordingly.

Hint: `3 + 3 + 1 + 5 = 12`.

As you may have guessed, if you need to add another row below the first one, you can.

Add as much rows as you want.

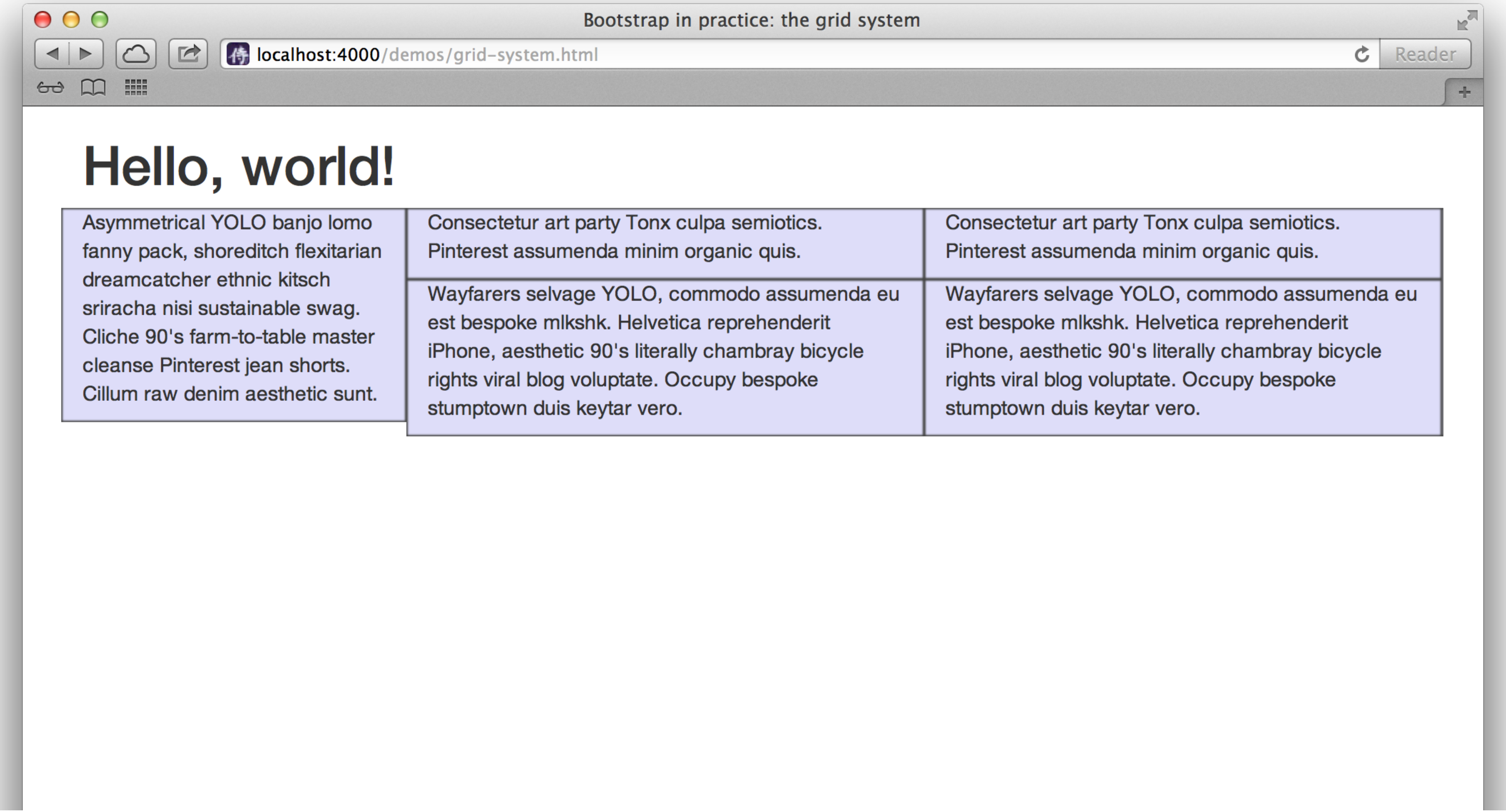
Section 4

NESTING

A great improvement of version 3 over version 2 is that the grid is now fluid by default.

For you, that means you can nest a row into another row and the inner one will always have 12 units for its columns.

Let's see this in action by producing a layout having two columns, with the second one being splitted into four boxes over two rows.


Bootstrap in practice: the grid system
localhost:4000/demos/grid-system.html
Reader

Hello, world!

Asymmetrical YOLO banjo lomo fanny pack, shoreditch flexitarian dreamcatcher ethnic kitsch sriracha nisi sustainable swag. Cliche 90's farm-to-table master cleanse Pinterest jean shorts. Cillum raw denim aesthetic sunt.

Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.

Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.

Wayfarers selvage YOLO, commodo assumenda eu est bespoke mlkshk. Helvetica reprehenderit iPhone, aesthetic 90's literally chambray bicycle rights viral blog voluptate. Occupy bespoke stumptown duis keytar vero.

Wayfarers selvage YOLO, commodo assumenda eu est bespoke mlkshk. Helvetica reprehenderit iPhone, aesthetic 90's literally chambray bicycle rights viral blog voluptate. Occupy bespoke stumptown duis keytar vero.

```
<!DOCTYPE html>
<html>
 <head>
  <title>Bootstrap in practice: the grid system</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="utf-8" />
  <!-- Bootstrap CSS -->
  <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" />
  <!-- To better visualize the columns -->
  <style>
   .row > div {
    background-color: #dedef8;
    box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;
   }
   </style>
 </head>
 <body>
  <div class="container">
   <h1>Hello, world!</h1>

   <div class="row">

    <div class="col-md-3">
     <p>Asymmetrical YOLO banjo lomo fanny pack, shoreditch flexitarian dreamcatcher ethnic kitsch sriracha nisi
sustainable swag. Cliche 90's farm-to-table master cleanse Pinterest jean shorts. Cillum raw denim aesthetic sunt.</p>
    </div>

    <div class="col-md-9">

     <div class="row">
      <div class="col-md-6">
       <p>Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.</p>
      </div>
      <div class="col-md-6">
       <p>Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.</p>
      </div>
     </div>

     <div class="row">
      <div class="col-md-6">
       <p>Wayfarers selvage YOLO, commodo assumenda eu est bespoke mlkshk. Helvetica reprehenderit iPhone, aesthetic
90's literally chambray bicycle rights viral blog voluptate. Occupy bespoke stumptown duis keytar vero.</p>
```

```
       </div>

       <div class="col-md-6">
        <p>Wayfarers selvage YOLO, commodo assumenda eu est bespoke mlkshk. Helvetica reprehenderit iPhone, aesthetic 90's literally chambray bicycle rights viral blog voluptate. Occupy bespoke stumptown duis keytar vero.</p>
       </div>
      </div>

     </div>

    </div>

   </div>
  </body>
 </html>
```

Section 5

ADDITIONAL GRIDS

Bootstrap 3 is Mobile First, meaning that it is engineered for smartphones first, then tablets, and finally desktops.

That means columns actually start at full width, stacked one after the other, and if the browser viewport is large enough they are reorganized and transformed into the nice grid we just read about in the previous sections.

This behavior covers the most common scenario where you want your page to be linearized on a small screen, and well laid out within a grid on a larger screen.

But if you need more control over your columns at every milestone viewport, you have four different sets of column classes.

It's like having four grids in one.

The extra-small devices grid

The first grid is active till the 768px – not included – breakpoint and you can use it with `col-xs-N`, with `N` from `1` to `12`. Defining something like:

```
<div class="row">
 <div class="col-xs-4">...</div>
 <div class="col-xs-8">...</div>
</div>
```

Will produce two columns starting from tiny screen devices, like phones, all the way up to large desktop sizes.

Unless you also intend to use...

The small devices grid

The second grid is active from the 768px breakpoint and you can use it with `col-sm-N`, with `N` from `1` to `12`. So, if you want the same columns as above to be of the same size when displayed inside a tablet-sized screen, you will add:

```
<div class="row">
 <div class="col-xs-4 col-sm-6">...</div>
 <div class="col-xs-8 col-sm-6">...</div>
</div>
```

And now you have a `4 - 8` layout on extra small screens, becoming a `6 - 6` layout for small screens and above.

Unless you need a different approach for...

The medium devices grid

This grid is active from the 992px breakpoint and you can use it with `col-md-N`, with `N` from `1` to `12`.

It's the grid we used at the beginning of this chapter and the heir of the old grid system of Bootstrap version 2, when the CSS class to use was `span-N`.

Anyway. Let's add a third different behavior.

```
<div class="row">
 <div class="col-xs-4 col-sm-6 col-md-3">...</div>
 <div class="col-xs-8 col-sm-6 col-md-9">...</div>
</div>
```

And you have a `4 - 8` layout on extra small screens, becoming a `6 - 6` layout for small screens, again changing to a `3 - 9` one for medium desktops and above.

The large desktop grid

Unless of course you add classes for the fourth grid, which is active from 1200px and above: `col-lg-N`, with `N` from `1` to `12`.

```
<div class="row">
 <div class="col-4 col-sm-6 col-md-3 col-lg-5">
   ...
 </div>
 <div class="col-8 col-sm-6 col-md-9 col-lg-7">
   ...
 </div>
</div>
```

And so you have a `4 - 8` layout on extra small screens, becoming a `6 - 6` layout for small screens, again changing to a `3 - 9` one for medium desktops and above, with a final transformation to a `5 - 7` layout on larger screens.

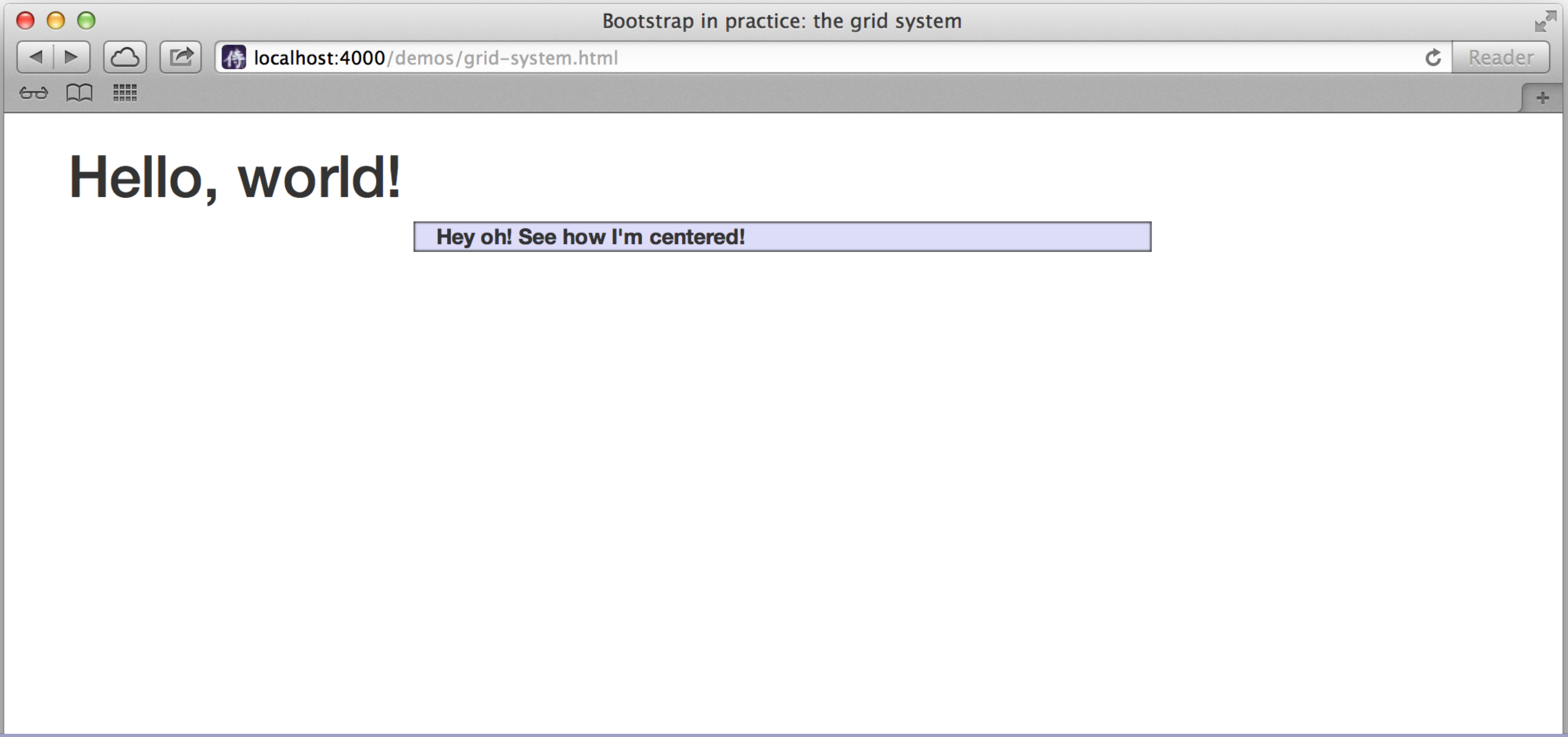
PLACEMENT OPTIONS

Offsetting

You can also move columns to the right using `.col-md-offset-N` where `N`, ranging from `1` to `11`, is the number of units you want the column to be moved.

Offsetting comes in handy when you need to center an element.

Say you have a `<div class="col-md-6">...</div>`. Centering it, is as easy as adding `.col-md-offset-3`:


Bootstrap in practice: the grid system
localhost:4000/demos/grid-system.html
Reader

Hello, world!

Hey oh! See how I'm centered!

```
<!DOCTYPE html>
<html>
 <head>
  <title>Bootstrap in practice: the grid system</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="utf-8" />
  <!-- Bootstrap CSS -->
  <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" />

  <!-- To better visualize the columns -->
  <style>
   .row > div {
    background-color: #dedef8;
    box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;
   }
   </style>
 </head>
 <body>
  <div class="container">

   <h1>Hello, world!</h1>

   <div class="row">
    <div class="col-md-6 col-md-offset-3">
     <strong>Hey oh! See how I'm centered!</strong>
    </div>
   </div>

  </div>
 </body>
</html>
```

That's because you have a total width of 12 units, and a 6 units wide element you want to center. Thus, you split the remaining 6 units in two, and here's your offset for centering the element.

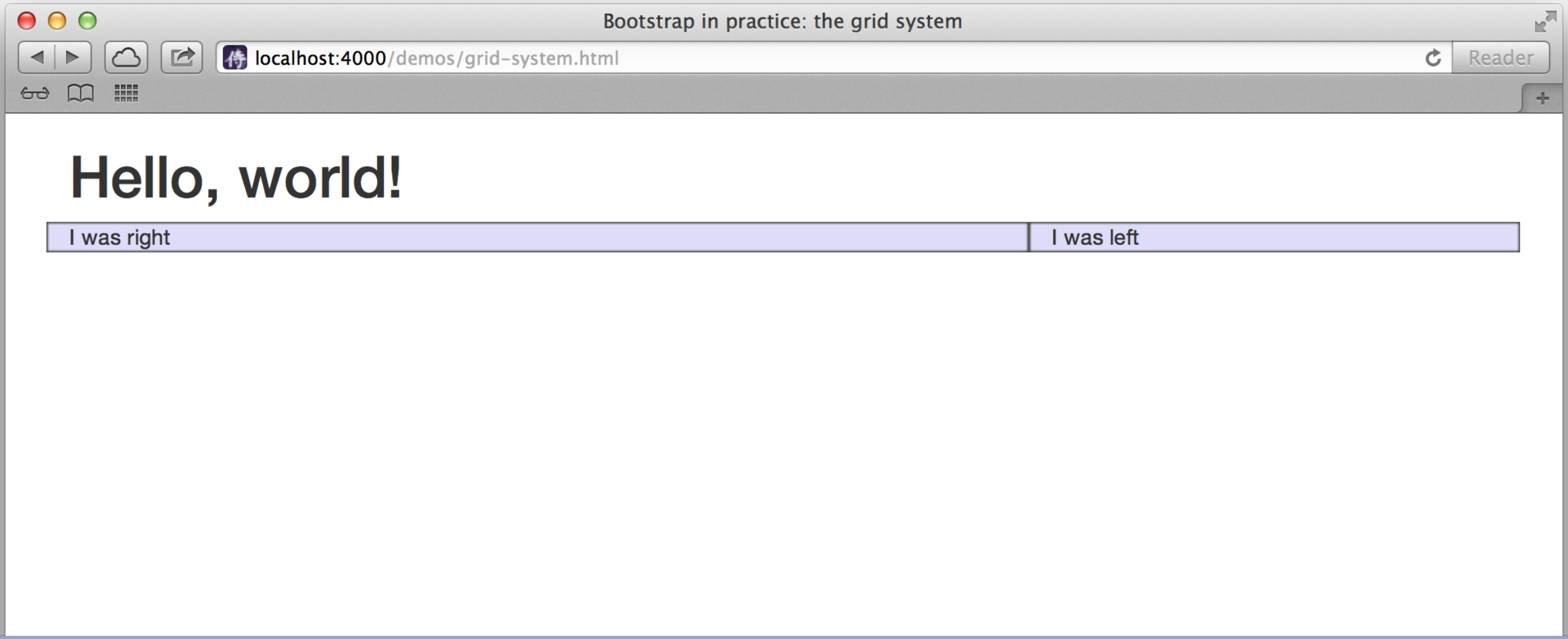
Ordering

Another nice feature of the grid system is that you can easily write the columns in an order, and show them in another one.

So if you have a two-column layout with the left column being the narrowest and acting as a sidebar:

```
<div class="row">
 <div class="col-lg-4">...</div>
 <div class="col-lg-8">...</div>
</div>
```

You can swap the order of the columns using `.col-md-push-N` and `.col-md-pull-N`. Once again, `N` is a number from `1` to `11`:


Bootstrap in practice: the grid system

localhost:4000/demos/grid-system.html

Hello, world!

I was right

I was left

```
<!DOCTYPE html>
<html>
 <head>
  <title>Bootstrap in practice: the grid system</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="utf-8" />
  <!-- Bootstrap CSS -->
  <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" />

  <!-- To better visualize the columns -->
  <style>
   .row > div {
    background-color: #dedef8;
    box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;
   }
   </style>
 </head>
 <body>
  <div class="container">

   <h1>Hello, world!</h1>

   <div class="row">
    <div class="col-md-4 col-md-push-8">I was left</div>
    <div class="col-md-8 col-md-pull-4">I was right</div>
   </div>

  </div>
 </body>
</html>
```

And your sidebar is now to the right.

Limitations

Offsets and ordering are not available for the extra small grid, but work like a charm for `-sm`, `-md`, and `-lg`.

MARK OTTO: creator of Bootstrap

Bootstrap is one of the most beloved superheroes, if you ask a web developer or web designer.

What's the story of its origins?

The background story is decently well articulated here: http://alistapart.com/article/building-twitter-bootstrap

Version 3 is an ambitious complete rewrite of Bootstrap, the most important change being that it's now "Mobile First".

As a professional working everyday inside and for the web, do you think this sudden change is a good move?

I don't think the rewrite or switch to mobile first is sudden at all.

We've been heading this way for awhile now and it makes a lot of sense in terms of file size, performance, and designer/developer mindsets.

Bootstrap, and its main competitor Foundation by ZURB, won almost hands-down the hearts of a multitude of web developers and designers from around the world.

I think the main reason is that after years of slowly honing our tools, we are finally evolving from craftsmanship-only to engineering-whenever-useful. Your thoughts?

Bootstrap in particular blew up I think for three reasons:

1. it has Twitter's name on it
2. it's HTML and CSS – the build blocks of the Web
3. the styleguide approach to the docs.

All those together made it possible for thousands of folks to use and learn from Bootstrap.

The future of the web seems to be taking shape as a multi-device one. Think of wearable computing, smart eyeware, consoles.

Is it just me or the classic smartphone-tablet-desktop triplet is suddenly becoming obsolete?

We'll get there soon enough.

For now, we have our work cut out for us on handheld and desktop devices without having to account for anything else. We're already barely doing that stuff right so throwing another into the mix seems like overkill right now. In a few years, yeah, maybe, but for now I don't see too much changing.