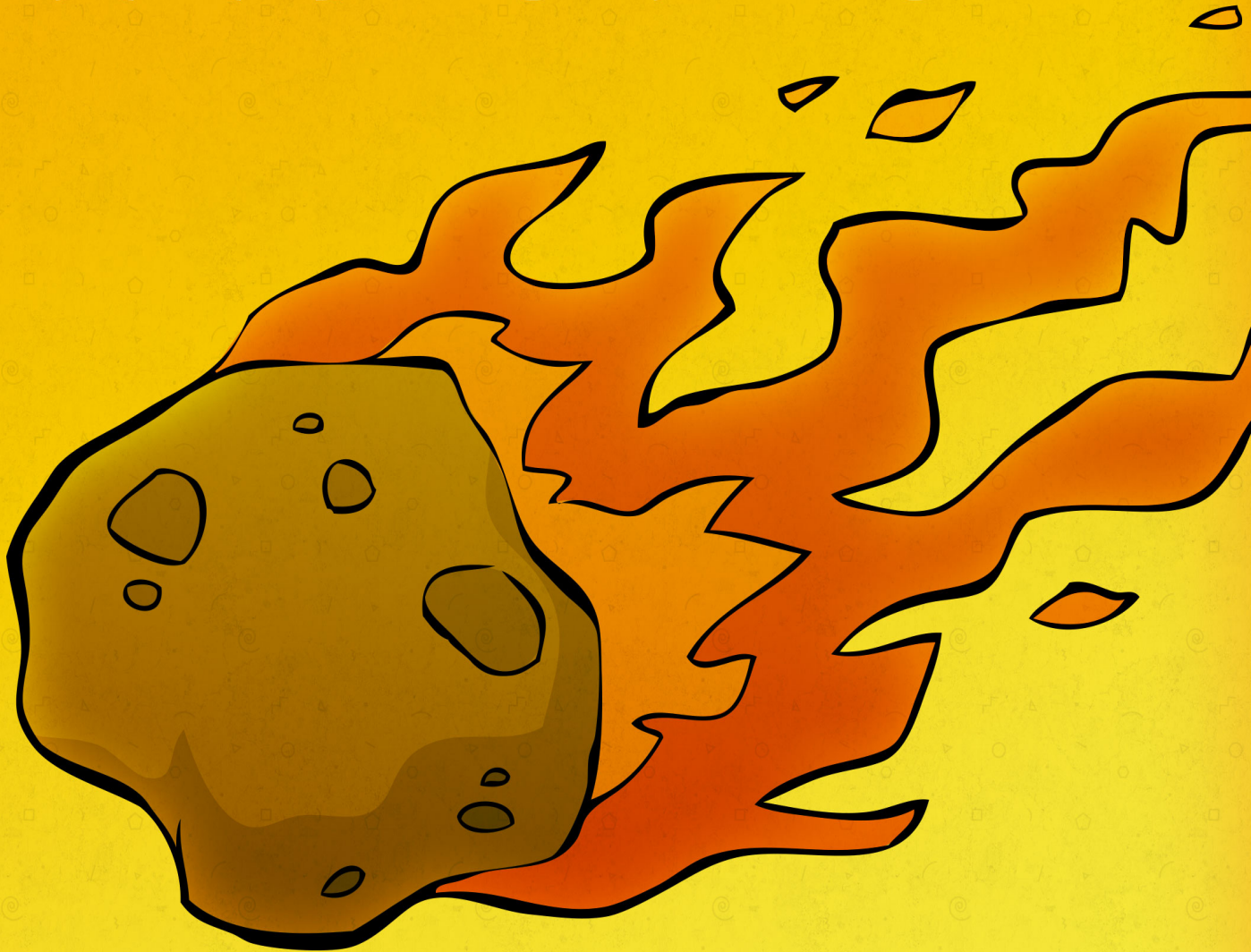


# ***YOUR FIRST METEOR APPLICATION***



# Your First Meteor Application

## A Complete Beginner's Guide to Meteor.js

David Turnbull

This book is for sale at <http://leanpub.com/meteortutorial>

This version was published on 2015-06-09



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2014 - 2015 David Turnbull

# **Tweet This Book!**

Please help David Turnbull by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#meteorjs](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#meteorjs>

# Contents

<b>1. Introduction</b>	<b>1</b>
1.1 Screencasts	2
1.2 Prerequisites	3
1.3 What You'll Need	4
1.4 Summary	5
<b>2. Getting Started</b>	<b>6</b>
2.1 Command Line	7
2.2 Installing Meteor	9
2.3 Summary	11
<b>3. Projects</b>	<b>12</b>
3.1 Create a Project	14
3.2 Local Servers	18
3.3 Default Application	22
3.4 Summary	24

# 1. Introduction

I built my first website at the age of twelve. It was an ugly thing with an animated background, bright yellow text, and a number of suggestions to “Sign My Guestbook”. But even these humble beginnings made me feel like a wizard. Here, I’d turned this blank page into something more with just a few lines of code and now people from around the world could see what I’d done. As *the* quiet kid, I’d never come across something so empowering.

In the years that followed, I spent my days dabbling in PHP and Rails and other technologies to build more feature-rich websites. But while I managed to turn this process into an enjoyable career from a young age, I struggled to recapture the magic of throwing HTML, CSS, and JavaScript into a file and seeing the results appear. Building something for the web had become a lot more powerful but also more complex.

Then I discovered Meteor.

I remember seeing version 0.5 on Hacker News and thinking, “This looks interesting,” but I was busy with other projects and didn’t look too deep into it. *Maybe once it’s matured*, I thought. Then I saw it pop up a few more times and, around version 0.8, spent an afternoon following some tutorials. Quite quickly, I realised this Meteor thing was a heck of a lot of fun.

Because while Meteor is powerful, it has an unparalleled element of simplicity and elegance — the web developer’s equivalent of an Apple product, I suppose. You only need a basic understanding of the classic trio — HTML, CSS, and JavaScript — and, in a matter of days, you can be building feature-rich and real-time web applications.

Repeatedly, Meteor has been described as “magical” and, while that downplays the impressive work of its developers, it’s hard to find another word to describe the feeling of working with it. At every turn, things seem to *just work*, and once-frustrating problems like building real-time interfaces no longer need solving. If you ever saw that infamous “[build a blog with Rails in 15 minutes](#)” video, it’s like that, but on a whole other level.

Is Meteor perfect? No. Nothing is. But unless you have a highly specialised use-case, the benefits of working faster, writing less code, and building something that harnesses the cutting-edge of the web’s technologies probably outweighs any of the negatives you’ll encounter (and with each release, the list of negatives continues to dwindle anyway).

If you’re hesitant to believe the hype though — maybe you’re reading this book just to make sure you’re not missing out — then I’m sure you’ll be convinced by the final page.

We’ll cover a lot of ground over the next few hours, walking through a gentle but broad introduction to what might possibly be the best way to build an application for the web.

Let’s begin.

## 1.1 Screencasts

I've put months of work into this book and I'm sure you're going to love it. You might, however, also like to check out a video training series that I've put together. The series covers the same topics as the book, with the same end in mind, but many would agree that it's easier to learn about programming by watching a video rather than reading a book.

To check out the video training series, visit:

<http://meteortips.com/screencasts>

The course includes:

- Over 2 hours of video footage.
- Detailed instruction for beginning developers.
- Free updates over the coming weeks and months.

It's available for \$39 and comes with a **30-day money-back guarantee** to make sure you have plenty of time to decide whether or not the product is right for you.





## 1.2 Prerequisites

I've read many technical books over the years and have found that authors use the phrase “for beginners” a little too liberally. I remember one book about iOS development, for instance, that had an extremely technical explanation of a particular concept in the second chapter, only for the author to finish the paragraph by saying, “But you don't need to know this yet...” Then I spent the rest of the chapter wondering why they'd bothered to mention this concept in the first place. (It's technically possible to skip parts of a book, of course, but the tricky part of being a beginner is not knowing what you can skip.)

To avoid this throughout the coming pages, let me start by explaining what you *won't* need to successfully make the most of the content inside this book:

1. **You won't need prior experience with Meteor.** I won't dwell on what Meteor is — it's a JavaScript framework for building real-time web applications — or its (relatively short) history, but I will show you how to install it, how to run Meteor applications on your local computer, and all of the other basic details to get you going.
2. **You won't need to have made a web application before.** There are theoretical aspects of coding web applications that we won't talk about, but if you're primarily (or entirely) a front-end developer, that's fine. You'll at least grasp the practical side of things.
3. **You won't need to consult any other sources along the way.** You're free to read other books and tutorials if you want but I won't expect you to read the official documentation or turn your eyes away from this book.

You will, however, need some background knowledge:

1. **You will need a basic understanding of JavaScript.** This means familiarity with variables, loops, conditionals, and functions. You won't need to be a JavaScript ninja, but the more familiar you are with JavaScript, the easier this process will be.
2. **You will need a basic understanding of databases.** This means familiarity with tables, rows, columns, and primary keys. You won't need to be a database whiz — your grasp of databases can be less comprehensive than your grasp of JavaScript — but if you've come into contact with something like MySQL before, that'll be a big plus.

If you need any of these skills, or want a refresher, visit the “[Resources](#)” section of [meteortips.com](http://meteortips.com) to find the relevant training material that I recommend.

## 1.3 What You'll Need

You don't need much "stuff" to develop with Meteor. This might seem like a minor detail, but a notable roadblock when getting started with other frameworks is the need to install software and mess around with configuration before getting the chance to write some code. With Meteor though, there's really only three things you'll need:

First, **you'll need a computer with any of the major operating systems** — Windows, Linux, or Mac. As of writing these words, Meteor is officially supported on Mac and Linux but there are ways to develop on a Windows machine. In either case, official support for Windows is coming in the future (although the precise date of that milestone is unclear).

Second, **you'll need a text editor**. Here, there's no precise requirements. If you want to write code in Notepad, then so be it. I'm quite fond of [Sublime Text 3](#) — a cross-platform editor with plenty of neat plugins and productivity-centric features — but we won't waste time on fancy power-user tricks that'll leave you feeling left out.

Third, **you'll need a relatively modern web browser**. This is what we'll use to preview our web applications on our local machines. Google Chrome is my browser of choice and it's what I recommend. If you're familiar with the development features of Safari or Firefox, then feel free to use them, but all of the examples in this book will make use of Chrome.

You'll also need Meteor itself, of course, and we'll install that in the next chapter.



## 1.4 Summary

Every chapter of this book ends with a summary of everything we've just covered in that chapter. Over the last few pages, for instance, we've learned that:

- Meteor is a fascinating framework that is great for beginning developers who have never built a web application before.
- You won't need a lot of background knowledge to get started with Meteor, but the more you know about JavaScript development and database theory, the better.
- You don't need much stuff to get started with Meteor. There's a very short distance between learning about Meteor and actually writing code.

Along the way, I'll also share a number of exercises to flex your Meteor-centric muscles. You don't have to do these exercises right away — in most cases, I'd suggest doing them *after* you've completed the main project we'll be working on — but I would suggest tackling each and every one of them at some point. You'll be more than capable of solving the problems they present and each one will deepen your understanding of the framework.

## 2. Getting Started

Since we're not sitting in the same room, I can't be sure of what you do or don't know about building web applications. Maybe you've built entire application before. Maybe you just started learning JavaScript a week ago. Maybe you're not even sure why you're reading these words.

Either way, I'll assume two things in the coming chapter:

1. You've never operated a computer with the command line.
2. You haven't installed Meteor.

If these assumptions are wrong, feel free to skip this chapter, but if these assumptions are not wrong, then continue reading to get a handle on the basics that will allow you to start writing code as soon as possible.

## 2.1 Command Line

Once upon a time, computers didn't have graphical interfaces with buttons, windows, and menus. Instead, users controlled their computers by typing out commands and tapping the "Return" key.

These days, *command line interfaces* as they're known — or "CLI" for short — are still used by software developers.

Why?

Well, although graphical interfaces are more welcoming to beginners:

1. They're time-consuming to create.
2. They're ultimately slower to use.

As such, there's no Meteor GUI. To install the Meteor framework on our computers, and then interact with that software, we have to work with the command line. This isn't as scary as it sounds though.

To get started, find the command line application on your computer.

All major operating systems have a command line application, but the name of the application will depend on the system:

- On Mac OS X, the command line application is *Terminal*.
- On Windows, the command line application is *Command Prompt*.
- On Linux, the command line application will depend on the distribution, but if you're using Linux, you probably know what you're doing.

After finding the command line application, open it up so we can continue.



*The command line application on Mac OS X.*

## 2.2 Installing Meteor

At the moment, Meteor is officially supported on:

- Mac: OS X 10.7 and above
- Windows:
  - Windows 7
  - Windows 8.1
  - Windows Server 2008
  - Windows Server 2012
- Linux: x86 and x86\_64 systems

Installing Meteor on Windows is straight-forward enough. Just [download and run the official Meteor installer](#). You only need to follow the prompts and you'll be good to go.

If you're using Mac OS X or Linux though, this is where we'll need to use the command line for the first time.

To begin, copy the following command to the clipboard:

```
curl https://install.meteor.com/ | sh
```

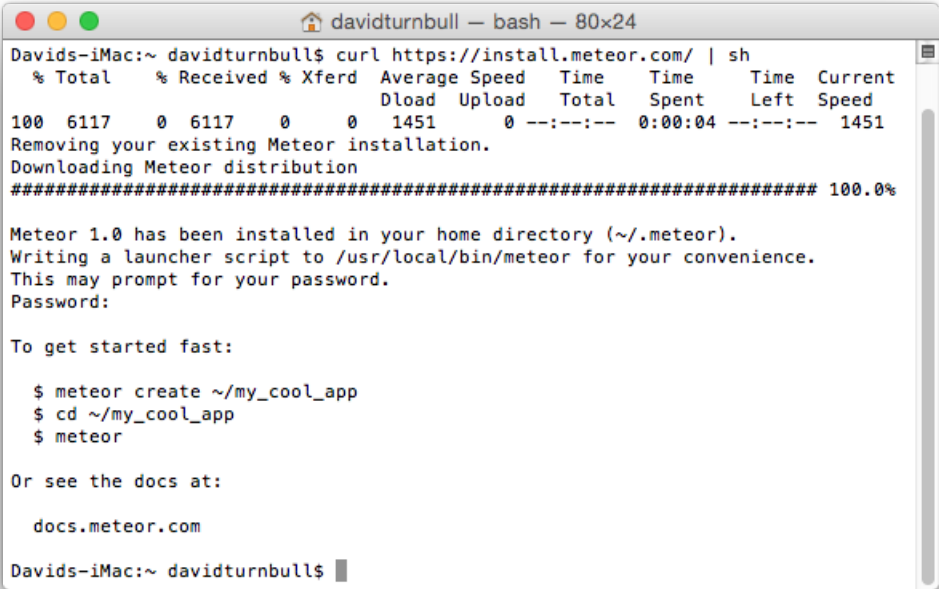
...and paste the command into the command line.

Then tap the "Return" key.

You don't need to understand exactly what this command is doing, but as a quick run-down, this command will:

1. Connect to "install.meteor.com".
2. Download the latest version of Meteor.
3. Install that version of Meteor.

If you're asked to enter your computer's password, fill it out and tap the "Return" key again. This is just to confirm that you have the appropriate privileges for installing Meteor on your computer.

A terminal window titled 'davidturnbull — bash — 80x24' on a Mac. The window shows the output of a curl command to install Meteor. It displays a progress bar for downloading the Meteor distribution, which is at 100.0%. Below the progress bar, it says 'Meteor 1.0 has been installed in your home directory (~/.meteor).', 'Writing a launcher script to /usr/local/bin/meteor for your convenience.', and 'This may prompt for your password.' followed by a 'Password:' prompt. Then it says 'To get started fast:' and shows three commands: '\$ meteor create ~/my\_cool\_app', '\$ cd ~/my\_cool\_app', and '\$ meteor'. Below that, it says 'Or see the docs at:' followed by 'docs.meteor.com'. The prompt 'Davids-iMac:~ davidturnbull\$' is at the bottom.

```
Davids-iMac:~ davidturnbull$ curl https://install.meteor.com/ | sh
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 6117    0 6117    0     0  1451      0 --:--:--  0:00:04 --:--:-- 1451
Removing your existing Meteor installation.
Downloading Meteor distribution
##### 100.0%

Meteor 1.0 has been installed in your home directory (~/.meteor).
Writing a launcher script to /usr/local/bin/meteor for your convenience.
This may prompt for your password.
Password:

To get started fast:

$ meteor create ~/my_cool_app
$ cd ~/my_cool_app
$ meteor

Or see the docs at:

docs.meteor.com

Davids-iMac:~ davidturnbull$
```

*Meteor is now installed.*

## 2.3 Summary

In this chapter, we've learned that:

- Before computers had graphical interfaces, they were controlled through the use of commands. These days, command line interfaces are still heavily used by developers.

To gain a deeper understanding of what we've covered:

- Read “[The Command Line Crash Course](#)”. This isn't necessary reading, but the more you know about working with the command line, the more productive you can be with it.

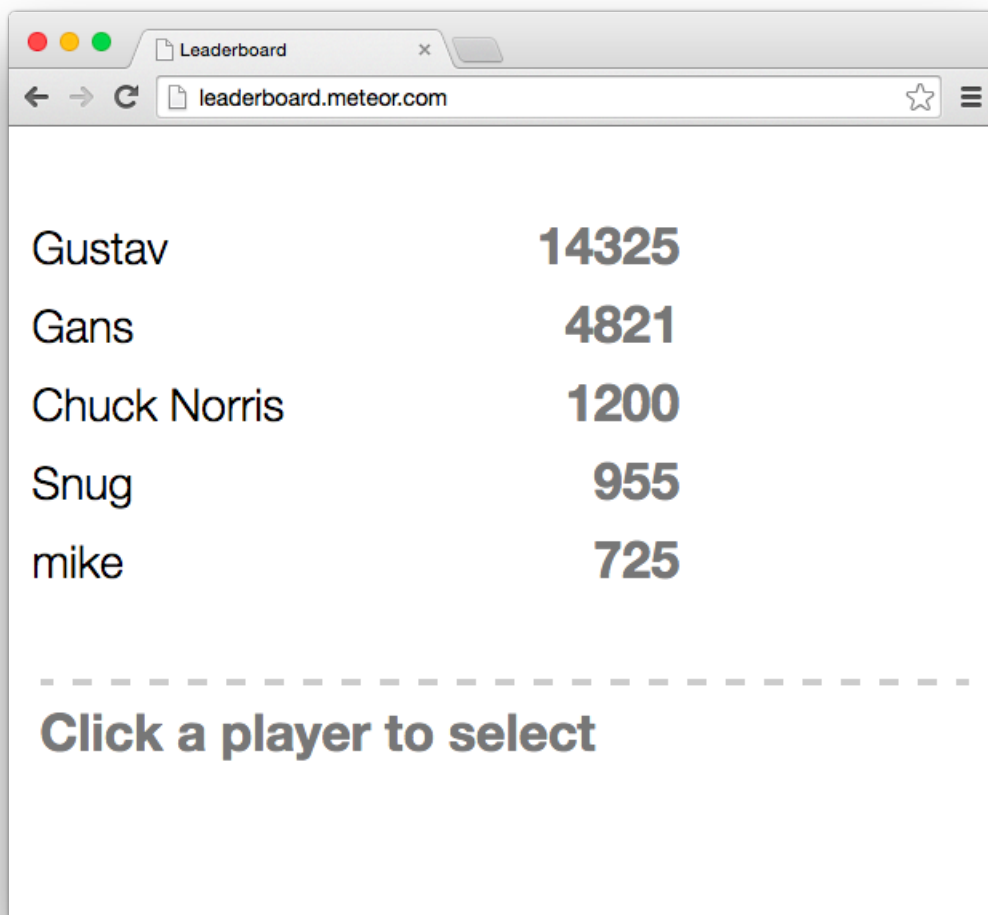


## 3. Projects

A big mistake that beginning web developers make when learning how to make web applications is trying to make progress without having a clear idea of what they're trying to build. But this is like driving to a new destination without a map. You might make a little progress in the right direction but you probably won't get where you need to go. You don't need to have everything figured out from the beginning, but you do at least need a direction.

With this in mind, we're going to build Leaderboard — an example application that was designed by the Meteor Development Group to show off what Meteor could do with very few lines of code.

Here's what it looks like:



*It's not the prettiest thing in the world.*

Leaderboard has since been replaced by more advanced examples on the official website, but it'll be our example project for two main reasons:

First, **the application already exists**. It's something we can play with, and that means we can get a good idea of what we're trying to build before we write a single line of code.

Second, **the application is simple**. This means we don't have to worry about the conceptual aspect of building the software (which is usually the most difficult part). Instead, we can focus on learning Meteor itself.

To get hands-on time with Leaderboard, visit [leaderboard2.meteor.com](http://leaderboard2.meteor.com) and, while clicking around, take note of its core features:

- There's a list of players.
- Each player has a score.
- Players are ranked by their score.
- You can select players by clicking on them.
- You can increment a selected player's score.

We'll create additional features in later chapters, but even this relatively short list will allow us to cover a lot of Meteor's core functionality.

## 3.1 Create a Project

To create our first Meteor application, we'll need to create our first *project*, and a project is the self-contained set of files that form the foundation of an application. You can use the words “project” and “application” interchangeably, but “project” is better-suited when talking about the application as it's being developed.

Every project is different, but will generally contain:

- HTML files, to create the interface.
- CSS files, to assign styles to the interface.
- JavaScript files, to define application logic.
- Folders, to keep everything organized.

A project can contain other types of files, like images and CoffeeScript files, but we'll keep things as simple as possible throughout this book and only work with what we need.

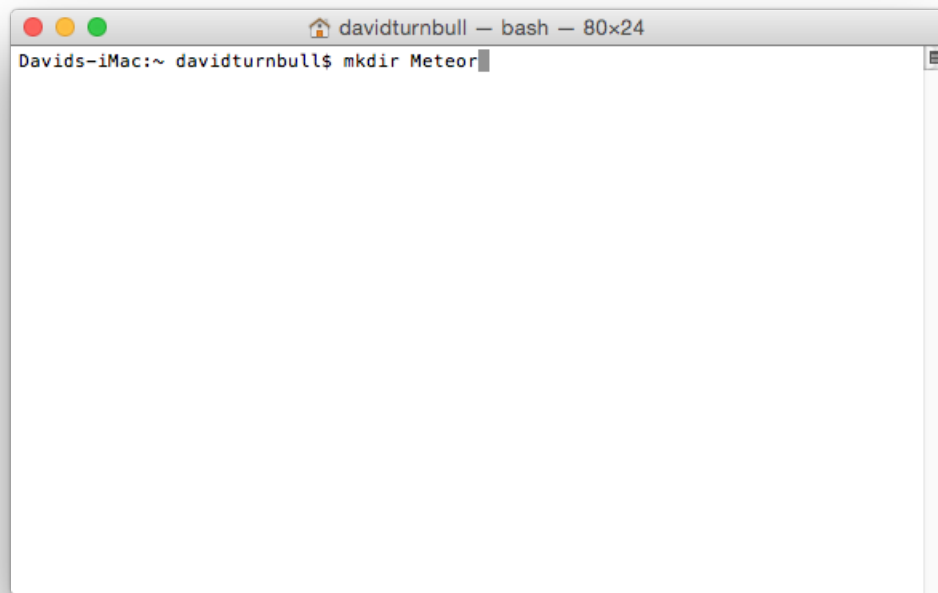
A project can contain other types of files, like images, but we'll keep things as simple as possible throughout this book and only work with what we need.

Before we create a project for the Leaderboard application though, let's create a folder to store our Meteor projects. We don't *have* to do this, but it's a good idea for the sake of keeping things organized.

We could, of course, select the “New Folder” option from the “File” menu, but where's the fun in that? Instead, enter the following into the command line:

```
mkdir Meteor
```

Then tap the “Return” key.



*Creating a folder with the `mkdir` command.*


This `mkdir` command stands for “make directory” and, as you can probably guess from the name, it allows us to make a directory.

In this particular instance, we’re creating a directory named “Meteor”, but you can call the folder whatever you want. The precise location where the folder will appear will depend on your operating system, but at least on Mac OS X, the folder will appear inside the “Home” directory by default. (And if you can’t find the created folder, simply search your computer.)

Once the directory is ready, navigate into it with the following command:

```
cd Meteor
```

This `cd` command stands for “change directory” and it’s the command line equivalent of double-clicking on a directory from within the graphical interface, so after tapping the “Return” key, we’ll be inside the “Meteor” directory.

A screenshot of a macOS terminal window titled "Meteor — bash — 80x24". The window shows a sequence of three commands and their prompts: 1. "Davids-iMac:~ davidturnbull\$ mkdir Meteor" with the prompt "Davids-iMac:~ davidturnbull\$ " on the next line. 2. "Davids-iMac:~ davidturnbull\$ cd Meteor" with the prompt "Davids-iMac:Meteor davidturnbull\$ " on the next line. 3. The prompt "Davids-iMac:Meteor davidturnbull\$ " followed by a cursor. The terminal has a standard macOS window frame with red, yellow, and green control buttons in the top-left corner.

*Navigating into the “Meteor” folder.*

To then create a Meteor project inside this directory, run the following:

```
meteor create leaderboard
```

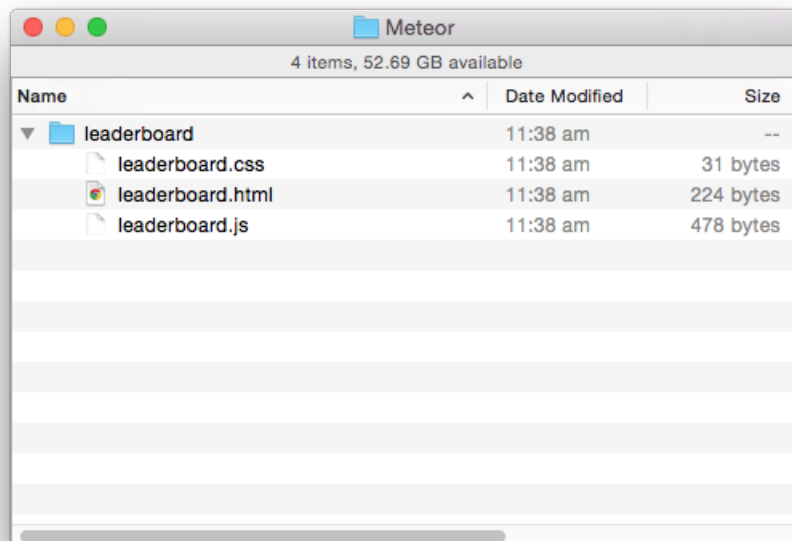
This command has three parts:

- The `meteor` part defines this as a Meteor command.
- The `create` part clarifies that we want to create a Meteor project.
- The `leaderboard` part is the name we’re assigning to the project.

After running this command, a “leaderboard” directory will appear inside the “Meteor” folder, and by default this folder will contain three files:

- `leaderboard.html`
- `leaderboard.css`
- `leaderboard.js`

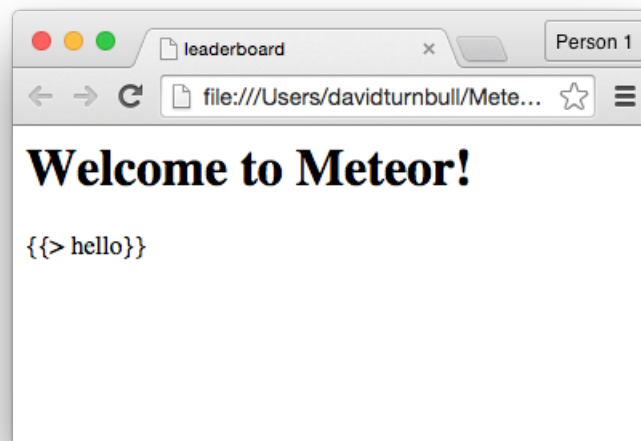
It will also contain a hidden folder — `.meteor` — but if your operating system hides this folder from view, that’s fine. We won’t be touching it.



*Inside our project's folder*

## 3.2 Local Servers

Web applications are not like static websites. We can't just open the `leaderboard.html` file and see the dynamic marvels of a Meteor application. In fact, if we open that file in Chrome, all we'll see is some static text:



*There's nothing dynamic about this.*

To get the web application working as planned, we need to launch what's known as a *local server*. This is a web server that runs on our local machine. It's included with Meteor itself and allows us to:

1. See the processed results of our JavaScript code.
2. Run a database on our local machine.

If you've used an application like MAMP for deployment with PHP and MySQL, this will be familiar, but if all of this sounds new and scary, fear not. In practice, it's quite simple.

Through the command line, navigate into the "leaderboard" directory:

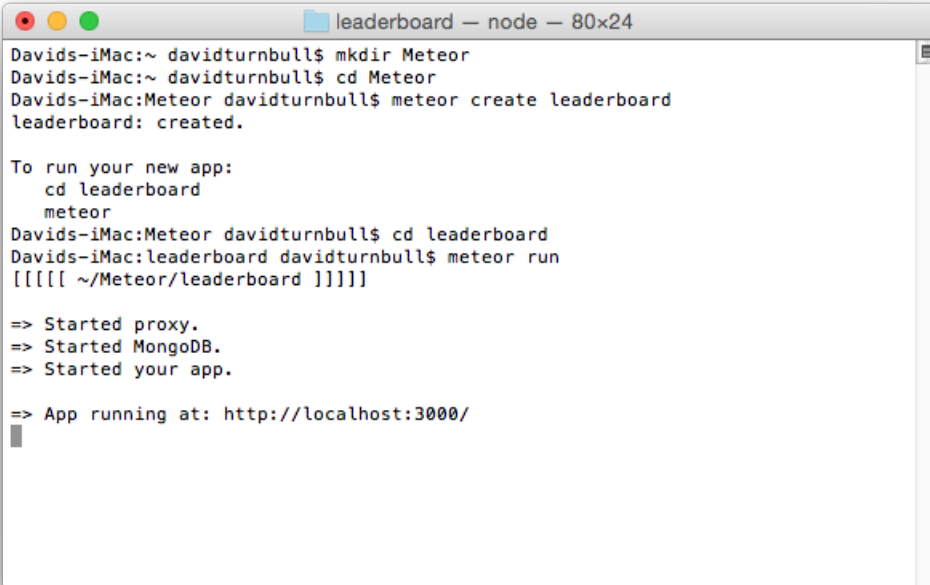
```
cd leaderboard
```

Then enter the following command:

```
meteor run
```

Here, the `meteor` part defines this as a Meteor command and the `run` part clarifies the precise action we want to take. In this context, we're wanting to run the local server.



A screenshot of a macOS terminal window titled "leaderboard — node — 80x24". The terminal shows the following commands and output:

```
Dauids-iMac:~ davidturnbull$ mkdir Meteor
Dauids-iMac:~ davidturnbull$ cd Meteor
Dauids-iMac:Meteor davidturnbull$ meteor create leaderboard
leaderboard: created.

To run your new app:
  cd leaderboard
  meteor
Dauids-iMac:Meteor davidturnbull$ cd leaderboard
Dauids-iMac:leaderboard davidturnbull$ meteor run
[[[[[ ~/Meteor/leaderboard ]]]]]

=> Started proxy.
=> Started MongoDB.
=> Started your app.

=> App running at: http://localhost:3000/
```

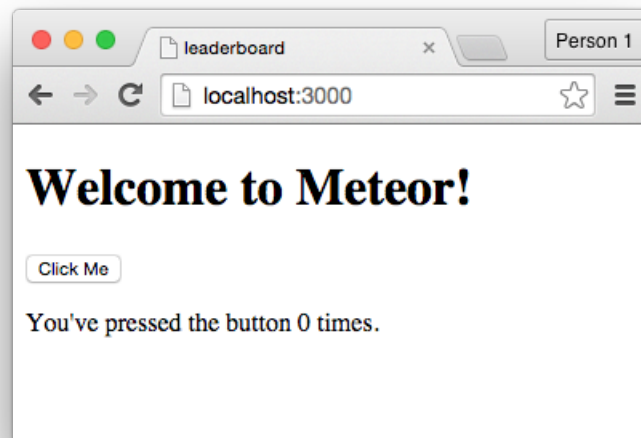
*Starting the local server.*

After tapping the “Return” key, the following should appear:

```
=> Started proxy.
=> Started MongoDB.
=> Started your app.
=> App running at: http://localhost:3000/
```

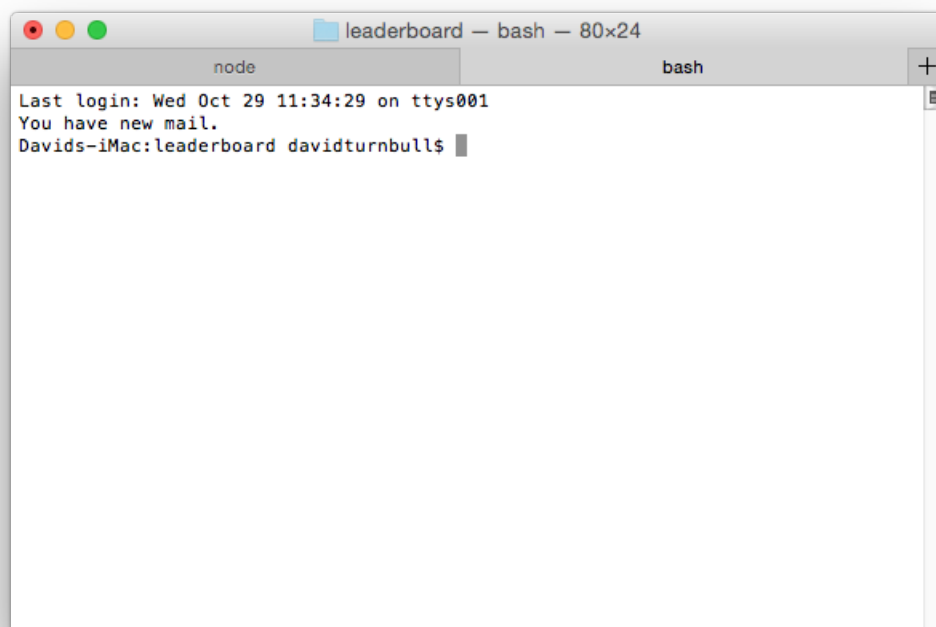
These lines confirm that the local server is starting and the URL on the last line — <http://localhost:3000> — is what we can now use to view our Meteor project in a web browser.

Navigate to this URL from inside Chrome and notice that we’re no longer seeing static text. Instead, we’re seeing a functional web application. The application itself is the result of the code that’s included with every Meteor project by default, and it’s not the most interesting creation in the world, but we’ve nevertheless taken a step in the right direction.



*This is the default Meteor application.*

To continually see the results of our code, we'll need to keep the local server running. This simply means leaving the command line open from this point onward. You will, however, need to open a separate tab or window to write further commands:



*A separate tab for running commands.*

To stop the local server, either quit out of the command line, or with the command line in focus,

press CTRL + C on your keyboard. To then start the local server again, use the same command as before:

```
meteor run
```

Just make sure you're inside a project's folder before running the command.

## 3.3 Default Application

The default application is nothing special, but if we click the “Click Me” button, the number on the screen will increment. This provides a fairly vanilla demonstration of Meteor’s real-time features. The code behind this application, however, isn’t precisely important since we’ll cover a far greater scope throughout the coming chapters.

For the moment, open the project’s files and delete all of the default code. Don’t even look at the code. Just get rid of it. We want to start from a completely blank slate.

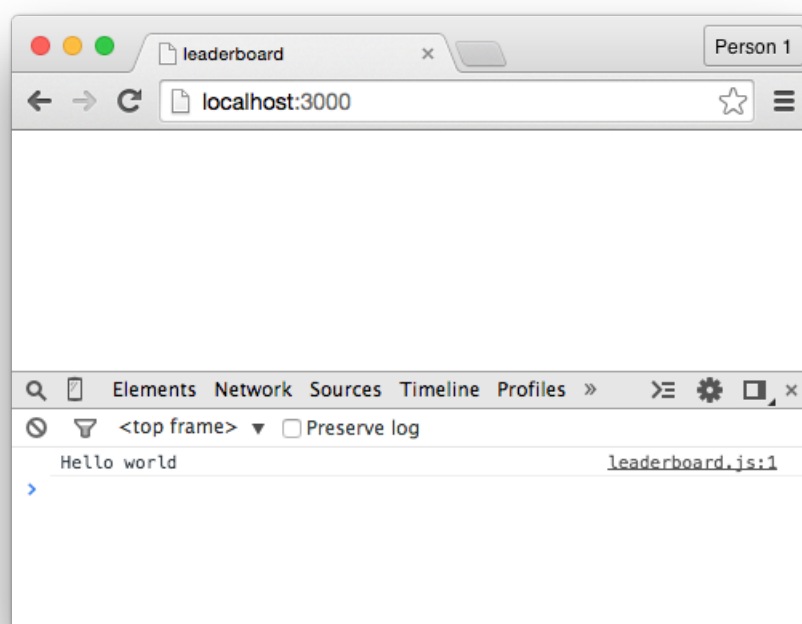
Once that’s done, type the following into the JavaScript file:

```
console.log("Hello world");
```

Then save the file and open the JavaScript Console from inside Chrome:

1. Click on the View menu.
2. Hover over the Developer option.
3. Select the JavaScript Console option.

A pane will open at the bottom of the browser and display the “Hello world” text that we just passed through the `console.log` statement.



*The “Hello World” text appears inside the JavaScript Console.*

If this is familiar, great. If not, then know that `console.log` statements are used to see the output of code without creating an interface to display that output. This means that, before we invest time into creating an interface, we can:

1. Confirm that our code is working as expected.
2. Fix any bugs as soon as they appear.

We can also use the Console to manipulate our application's database, which is what we'll do in the next chapter.

Leave the Console open from this point onward, but feel free to delete the `console.log` statement from inside the JavaScript file.

## 3.4 Summary

In this chapter, we've learned that:

- When learning how to build a web application, it's important to have a clear idea of what you're trying to build.
- The command line can be used to quickly achieve familiar tasks, like creating folders and navigating between them.
- When developing a Meteor application, we refer to it as a “project”, and we can create a project with the `meteor create` command.
- To view our web applications on our local machines, we can use the `meteor run` command to launch a local web server.
- When combined with `console.log` statements, the JavaScript Console is a very handy tool for Meteor development.

To gain a deeper understanding of Meteor:

- If you haven't already, play around with the original Leaderboard application. It doesn't have a lot of features, but that just means there's no reason to not have a strong grasp of its functionality.
- Close the command line application, then open it again and get back to where you were. You should be able to navigate into your project's folder with the `cd` command start the local server with `meteor run`.
- Create a second Meteor project and use this project to experiment whenever you learn something new. Messing around for the sake of it is a fantastic way to drill the details deeper into your brain.

To see the code in its current state, check out [the GitHub commit](#).