Become a Technical Marketer

A Guide to Working Faster, Accelerating Growth, Automating Marketing Tasks, and More

By Justin Mares and Nathaniel Eliason

Foreword

When I started at Facebook as a product manager I wanted certain features to be built. I'd approach one of the smart engineers on the team and ask them to make something. Then the waiting process began.

Ultimately, I decided to just learn how to code myself to get the features I wanted sooner. This taught me the fundamentals of programming and has paid huge dividends in my marketing roles later in life.

Making products can be easy but getting the word is generally the hardest part. Many think of it like a black magic that doesn't exist. Just make a great thing and hopefully people will hear about it.

The reality is, now with technical skills you can look at marketing the same way an engineer does, as a science. I did this at Mint when I started to really focus on the quantitative skills of marketing and how to engineer the growth that you want for your business. No more voodoo marketing where you are hoping that the activities you do produce results.

When I stumbled across Justin and Nat's course I was jealous. Jealous that it wasn't around while I was in college or available when I was just learning marketing. They are very thorough with the relevant technical skills you need to kick ass with marketing today. It's not just read this code and something happens, they show you the exact diagram to do it. I definitely recommend everyone to not only read / enjoy the material they create but more importantly take action.

- Noah Kagan, Chief Sumo at Appsumo

Introduction: Why Technical Marketing?

“Code is part of our Marketing DNA, and even if we don’t need to be able to build sites from scratch, as marketers today, we need to understand how it works to make informed decisions.” - *Evelyn Wolf, HubSpot*

Marketing and technology become increasingly entwined each year. As more marketing and advertising dollars move online and away from the old physical standbys, the necessity for understanding how these digital marketing avenues work at their core becomes more important.

To be involved in marketing at any modern company, technical skills are a major asset.

They’re the difference between asking a developer to fix part of your landing page, and fixing it yourself.

They’re the difference between manually emailing dozens of reporters and having your computer do it all automatically.

They’re the difference between a generic onboarding welcome sequence, and one that creates a viral referral loop.

They’re the difference between being able to query your company’s database to learn more about your users, and having to ask someone to look up the info for you.

And they’re the difference between being in very high demand as a marketer with technical competence, vs. the more generic non-technical marketer.

But if you wanted to get started becoming a “technical” marketer, what would you do? Courses like Codecademy and Code School are focused on turning you into a developer. They're useful, but they're overkill for someone who just wants marketing-focused technical skills. Not to mention that since they are so in-depth they take a long time to finish and could scare off newcomers.

That's why we created the "Programming for Marketers" email course, and subsequently this ebook. The goal is to show you that being a "technical marketer" is primarily about being able to hack things together and come up with your own solutions to problems drawing from some basic technical knowledge.

Programming is actually the **last resort** of a good technical marketer. Instead, a good technical marketer looks for tools first, and then looks for tactics using services we'll cover shortly, then looks for existing code snippets like the ones we'll give you, and only after they've exhausted those three options do they start writing code.

This mentality leads to the motto for our course, and the motto for technical marketers everywhere:

"Don't write code when a hack will suffice, and don't use a hack when a system will suffice."

These eight lessons will show you how to approach common marketing problems or needs through this lens, and along the way we'll show you some great tools, teach you some cool hacks, and give you some familiarity with HTML, CSS, Python, and SQL.

This ebook will always be evolving, so don't hesitate to send us feedback on it. You can reach Justin at justin@programmingformarketers.com, and you can reach Nat at nat@programmingformarketers.com.

A Special Bonus

Before we get started, we want to share something with you.

Some of our early readers commented that going through these lessons on a Kindle or desktop Kindle reader was a little frustrating. So we want to make it easier on you.

We created a PDF version of this book with larger full color pictures that we'll send you for free since you bought the book. In addition, we'll send you:

New **bonus lessons** every couple weeks that aren't in the ebook.

A PDF covering 15+ marketing hacks you can do **right now** using 0 programming that most people will assume you had to write code for.

A PDF explaining how to automate your scripts that you'll be creating throughout the course so they run daily, hourly, whatever you please, and another PDF on how to add email addresses to your list by text message.

To get it, just go to: http://www.programmingformarketers.com/ebook and then enter in the coupon code "ebook" to get everything for free.

Setup

If you want to, you can skip straight to Lesson 1 and come back here as you need it. Throughout the lessons though, you'll need to use things such as Python, Text Editors, GitHub, and other digital tools that will be confusing if you haven't used them before. This section will explain how to get up and running with each of them.

Getting Familiar with Stack Overflow

Throughout the process of learning to be a more technical marketer, Stack Overflow will be your best friend. Stack Overflow is a message board where anyone can post questions specific to programming, and then the best answers are voted up so they rise to the top.

Any problem that you'll conceivably have with Python, SQL, HTML/CSS, or other programming languages has most likely already been answered there. So if you do run into any issues, go there first and see if someone has posted a solution to the problem you're having. If you still can't figure it out, then go ahead and email Nat (nat@programmingformarketers.com) and we'll try to help you out.

Getting Set Up With a Text Editor

In many of the lessons you're going to need to slightly edit some code in order to make it work for you. To do this, you want to have a text editor that's friendly with programming. Apps like "Notepad" will work, but they're suboptimal. We're going to make two recommendations for the course.

For Editting HTML/CSS

In one of the lessons you'll be working on a referral landing page to help motivate people to share your site, ideally for a bonus. When you edit HTML and CSS (the main programming languages of the web) you can use "Brackets" on Mac, or "Coffee Cup" on Windows. The benefit to these editors is that since HTML and CSS are very visual languages, you can instantly see how your code changes are affecting the appearance of the webpage.

For Editing Everything Else

When you're not working with HTML/CSS, you can use a text editor called "Sublime Text". We like Sublime because if you continue learning programming there are a lot of plugins that will make your life easier, and it just looks pretty. Sublime works on both Mac and Windows.

Getting Set Up With Python

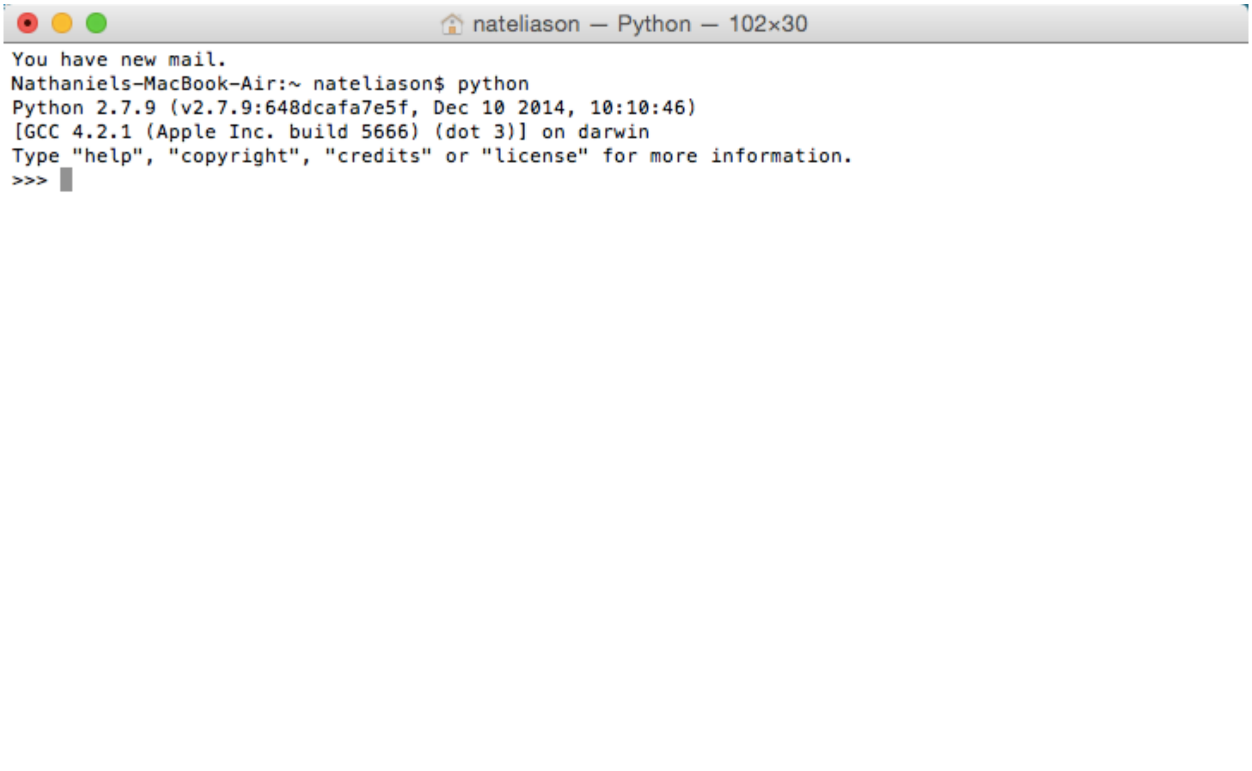
The main programming language we're going to use the course is Python. There are two reasons for this: it's syntactically the most clear and easy to understand; and since it's so popular there are a ton of pre-existing code snippets and APIs for it. The latter means that it's very easy to, say, connect to Twitter through Python, among other services.

Installing Python

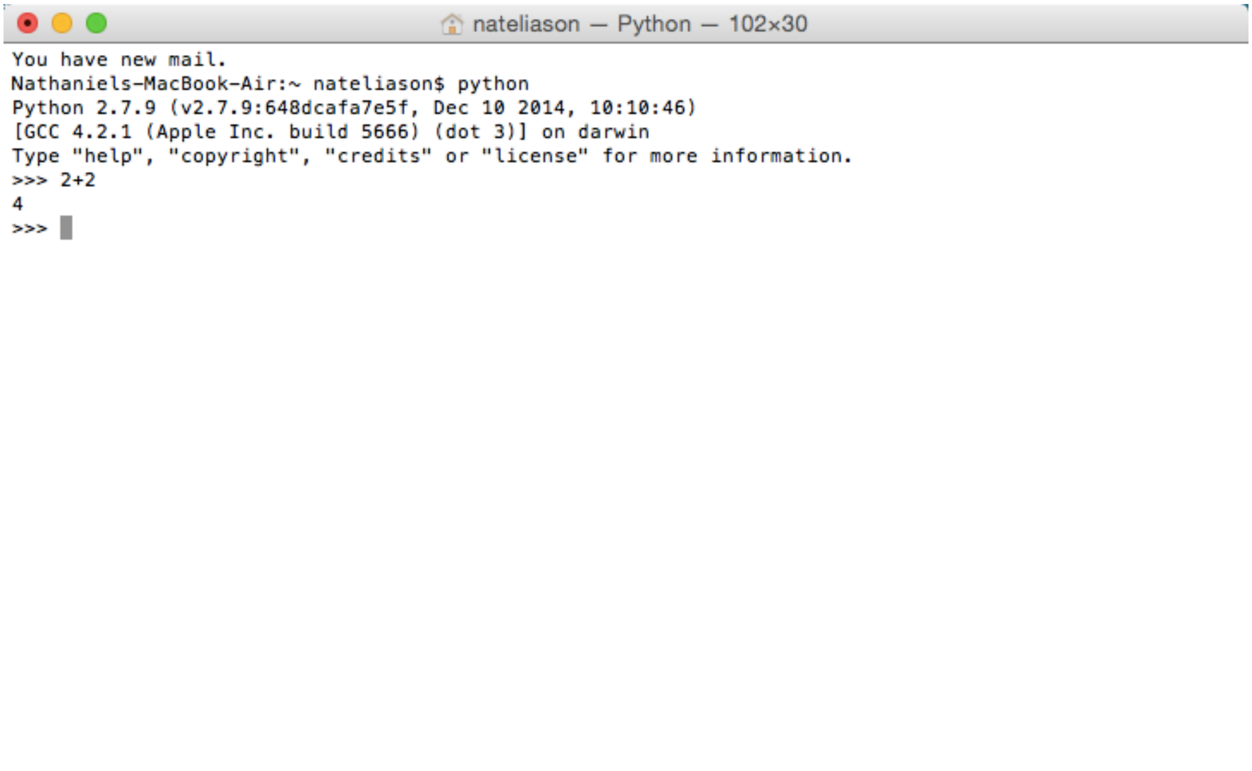
If you're a Mac user, good news! Python is already installed. If you're a Windows user, go to this link and follow the directions: https://www.python.org/downloads/windows/

Testing Python

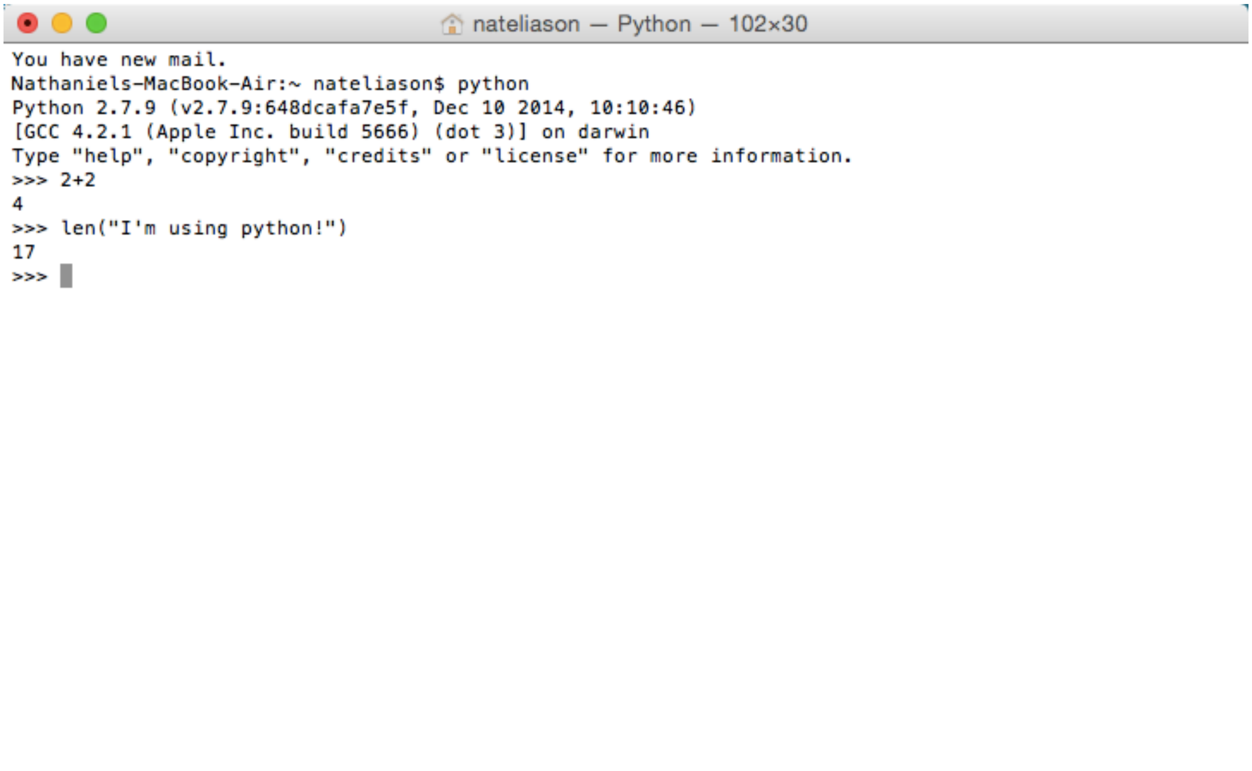
If everything installed properly, then you'll have Python available from anywhere on your computer. To test this, open up "Terminal" on Mac or "Powershell" on Windows and just type in "Python." If it installed correctly you should see something like this:



What that did is start the "Python" process so you can enter code. For example, try typing in "**2+2**" and then hit enter.



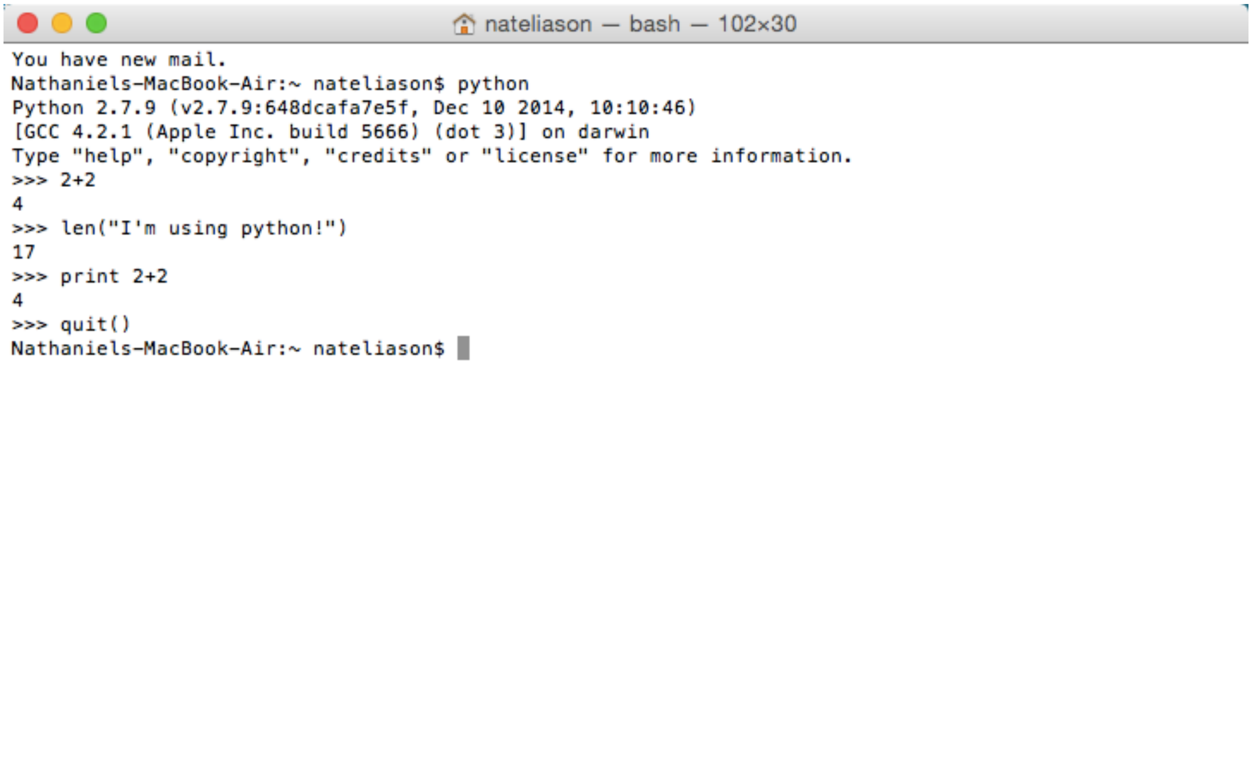
You can also do more complicated things. Try typing in "**len("I'm using python!")**" and then hit enter.



What that did is calculated the **len**gth of "I'm using python!" and then printed the result. In this case 17.

The most important thing you can do is run scripts saved as Python files. First, create a folder on your desktop called "Test." Next, open up your text editor from before, and just write "print 2+2." Now save that file as "test.py" within the folder.

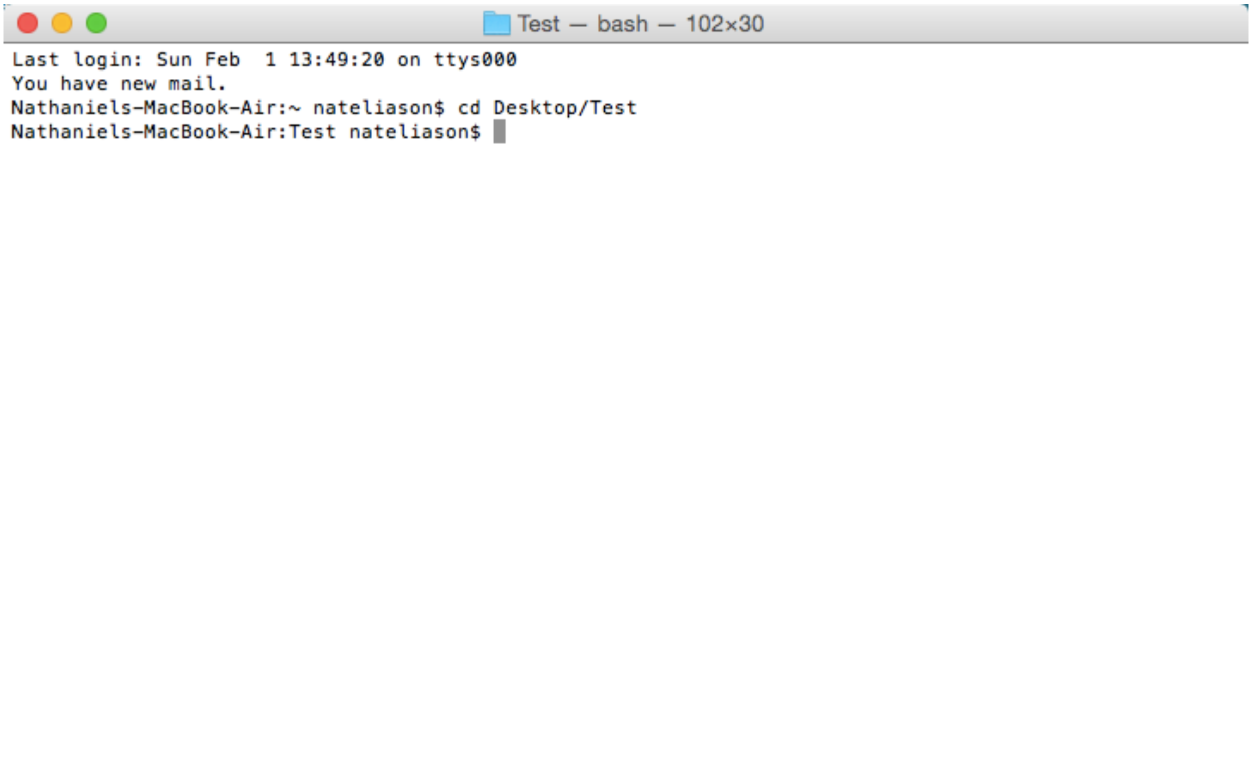
To run it, go back to your terminal. If you still have python running, type "quit()" so your terminal looks like this again:



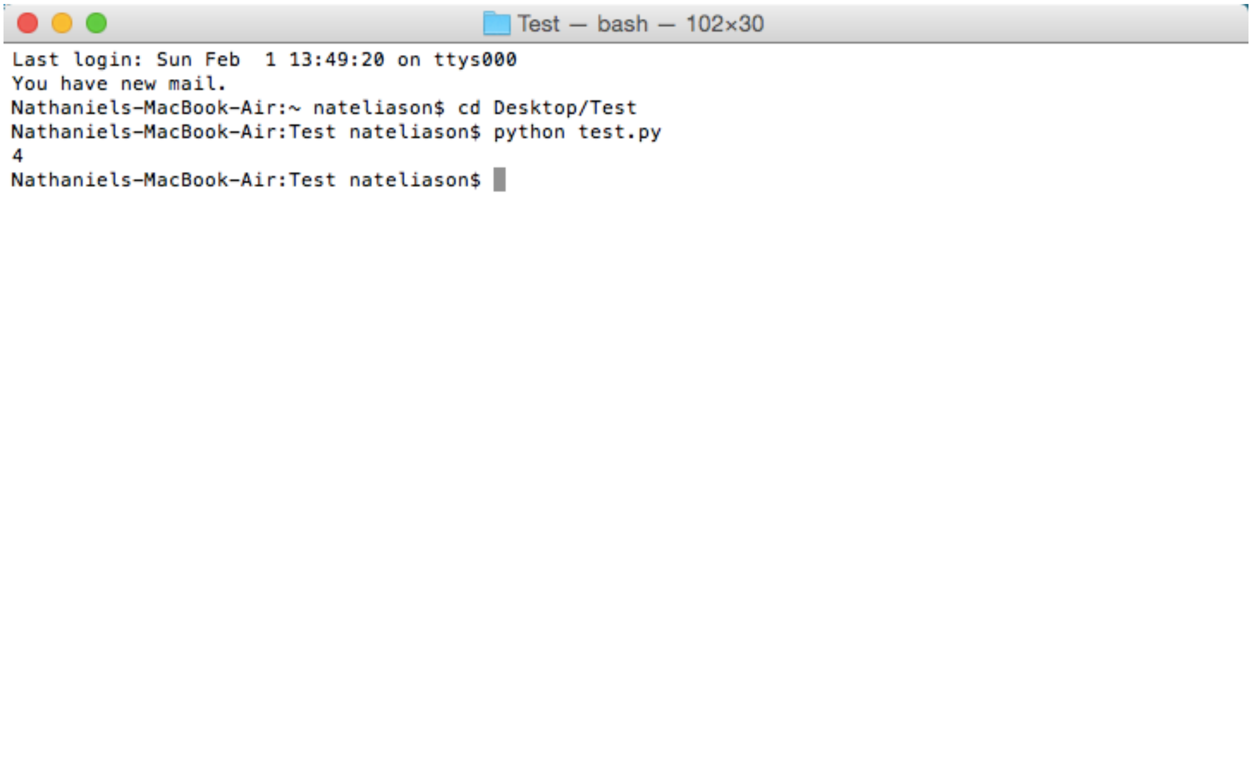
Now you need to navigate to where you put that file. In the terminal, this is done with the command "CD." So to get to my desktop, I'm going to type in "CD Desktop/" and hit enter. (Note: If you start typing "Des…" you can hit "Tab" and it will finish the folder name for you)

```
nateliason — bash — 102×30
You have new mail.
Nathaniels-MacBook-Air:~ nateliason$ python
Python 2.7.9 (v2.7.9:648dcafa7e5f, Dec 10 2014, 10:10:46)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+2
4
>>> len("I'm using python!")
17
>>> print 2+2
4
>>> quit()
Nathaniels-MacBook-Air:~ nateliason$ cd Desktop/
```

Now you're in your desktop. To get to your "Test" folder you can type "cd Test" or you could have just added it after "Desktop" in the last step.



Now to run your test program, just type in “python test.py”



And as you can see, it ran the file, and the result of "2+2" is 4, so it "print"(ed) 4 as we told it to in the file.

This process of navigating to files and running them will be involved in a few of the lessons, so be sure to refer back here as you need to.

Lesson 1: How to Create Your Own Landing Page from Scratch

A landing page is the perfect way to test a new idea without having to spend too much money or too much time putting the experiment together. Instead of creating a ton of content, or even a full-fledged product, you can put up a landing page with an option to pre-order or sign up for updates and then gauge people's interest in the final product by how they respond to the landing page.

When we first created Programming for Marketers, we created initial sketches of what the lessons would look like, but we didn't write the entire lessons out. We wanted to make sure people were interested first.

To do that, we put up our landing page at www.programmingformarketers.com and then saw how many people signed up in the first few days. Since the response was extremely positive (we got three times as many signups as we wanted for our validation threshold) we decided to move ahead with creating the lessons.

You too might have an idea you want to test out, but you're not sure where to start. Or maybe you want to practice your technical marketing skills on something besides your day job. Whatever your primary reason is, knowing how to quickly set up a landing page is a useful skill that will serve you well.

This lesson will take you from zero to landing page in less than 30 minutes, and most of those 30 minutes will be you designing it on your own after the lesson. Actually getting it built is the easy part.

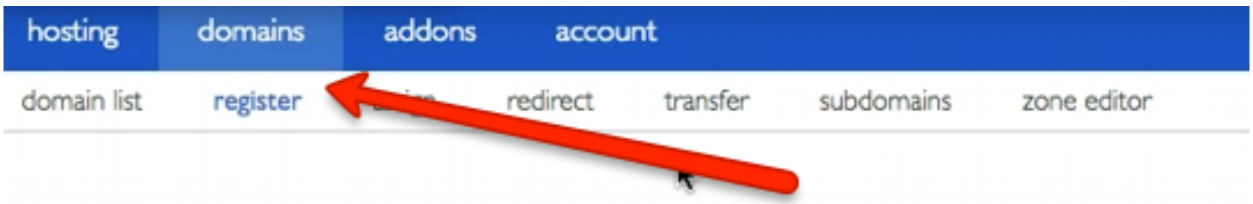
Step One: Get Hosting for Your Landing Page

Before you can build a landing page, you need a place to put it. We're going to be using WordPress as the base for this landing page, which means we want to use BlueHost for our hosting.

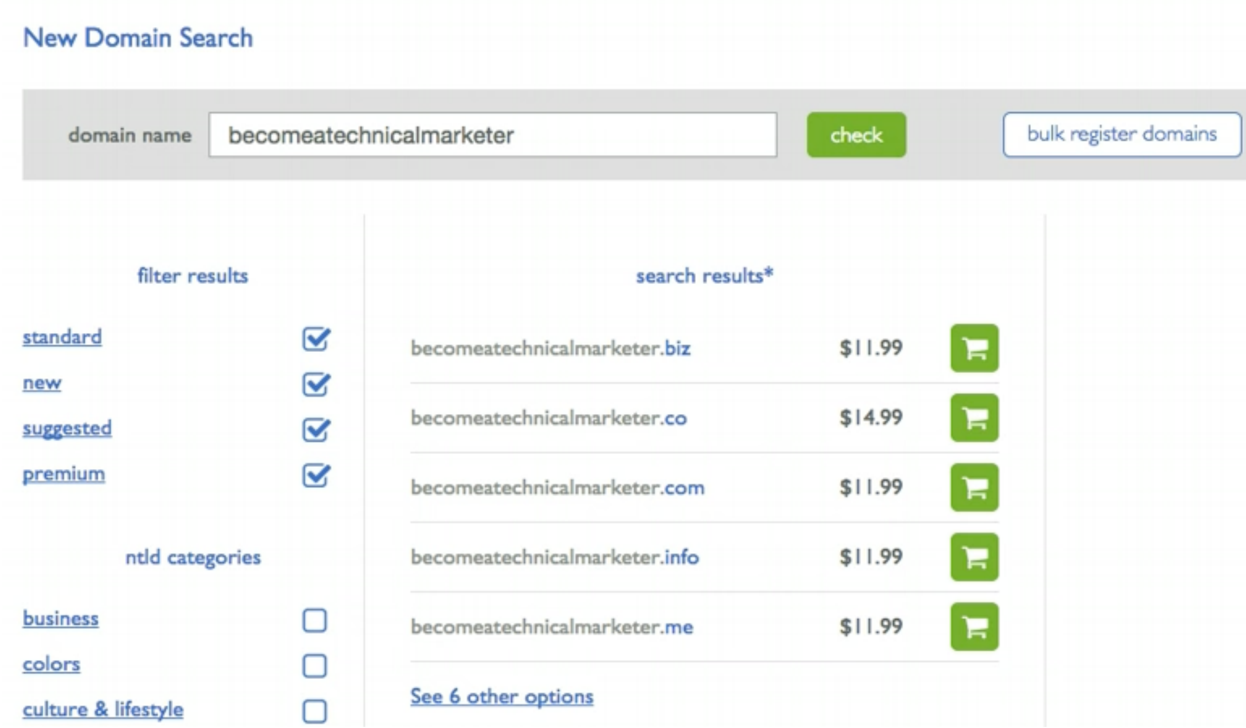
BlueHost is great for hosting in this case because it lets you do easy 1-click installs of

WordPress, without having to know any programming or doing complicated installations directly on the server. The combo of WordPress and BlueHost is what we used to make the Programming for Marketers website, and it will serve you well.

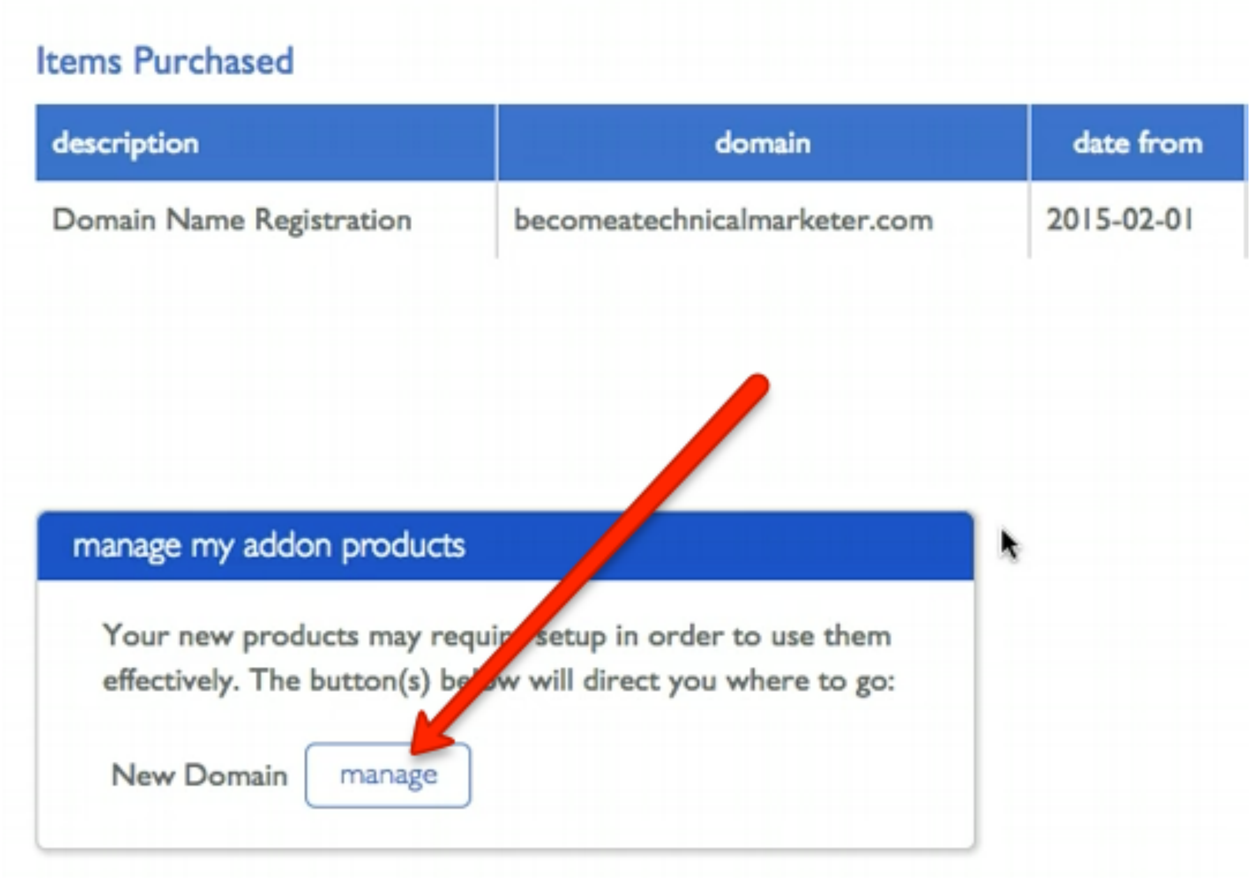
Once you go to [BlueHost]() and sign up for hosting, you'll need to buy a domain. Click on the "Domain" tab in the top, then click on "register."



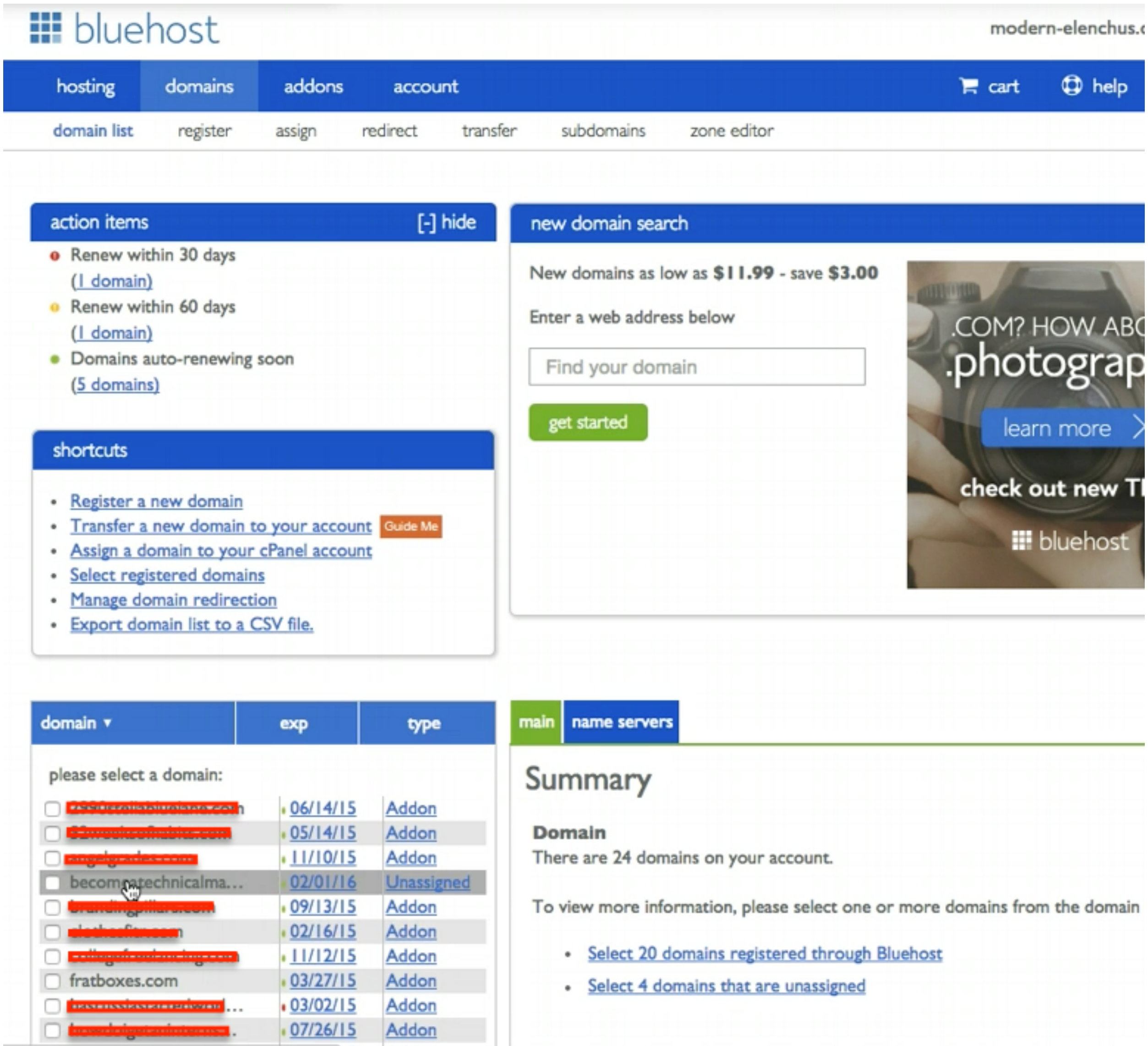
On this page you can type in the URL you want, with or without the suffix. If you do it without then BlueHost will show you all of the suffixes available for that URL. Since these lessons are about becoming a technical marketer, I'm going to go ahead and pick up [www.becomeatechnicalmarketer.com]()



Once you've found one you want, you can go ahead through the checkout, and then it'll present you with a confirmation screen that has an option in the lower left to manage your new domain.



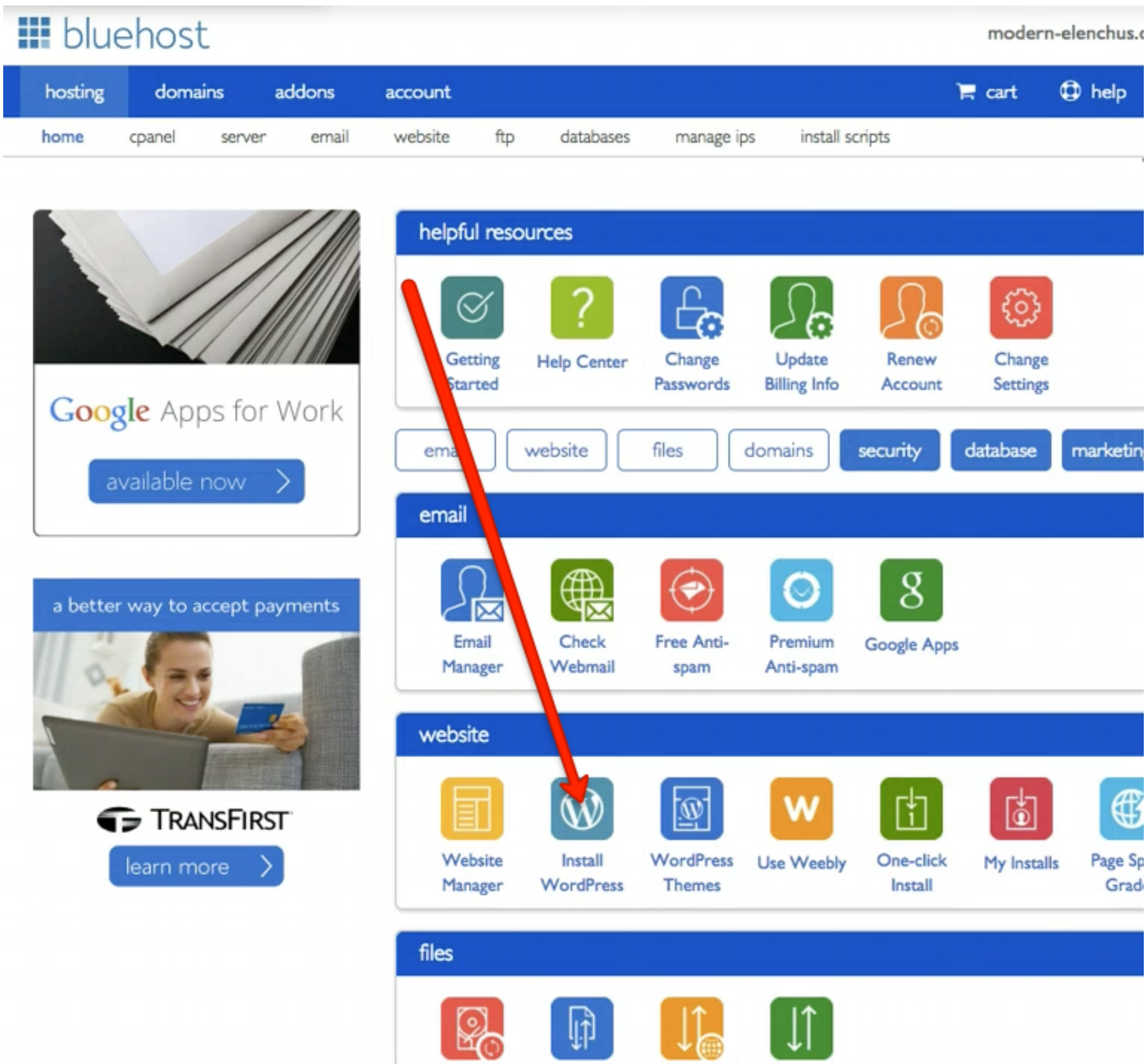
Go ahead and click that. Now on this page you'll be able to see your new domain in the bottom left, but it will be Unassigned. You need to turn it into an "Add-on" which you can do by first clicking on "Unassigned."



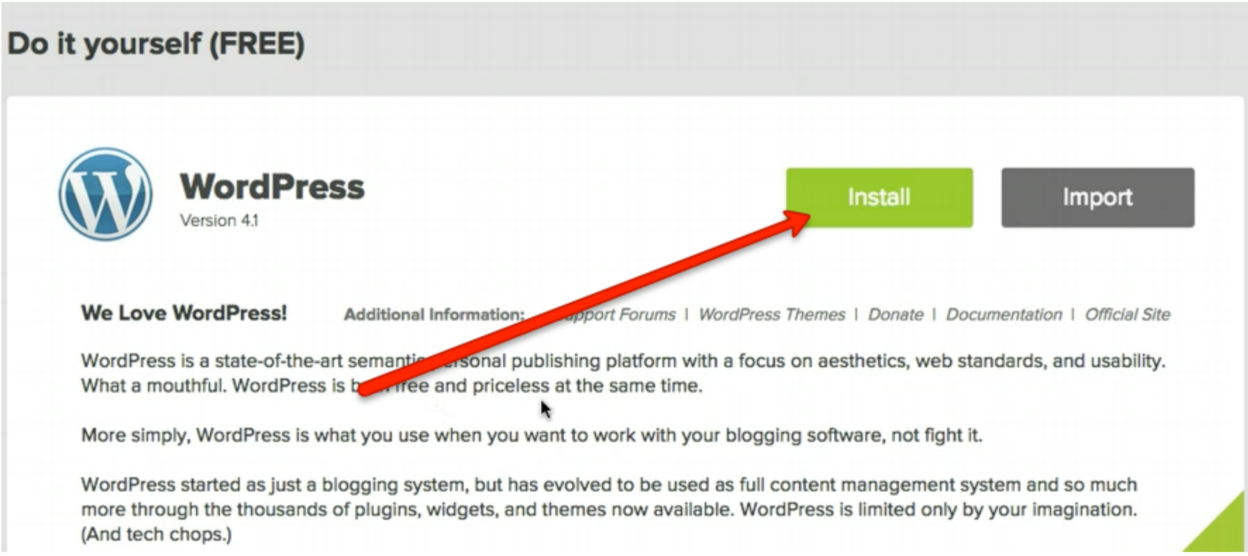
On the page that comes up, just leave "Add-on Domain" selected, and hit the confirmation button at the bottom. After a short wait your domain will now be set up as an independent site! Check it out. There won't be much there (probably just some BlueHost logos), but we're just getting started.

Step Two: Adding WordPress to Your New Site

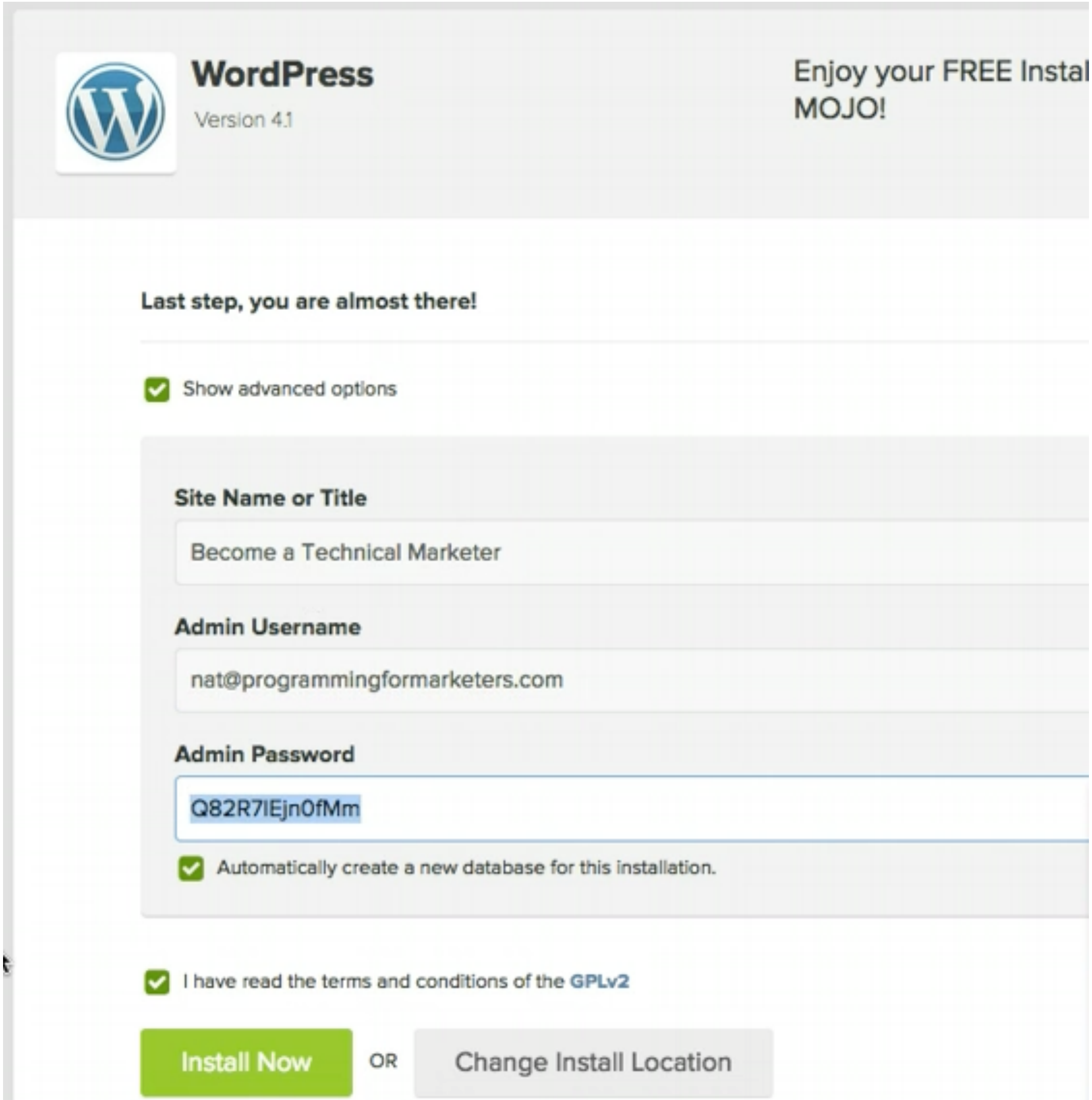
Now we're going to add WordPress, the framework that your site will be built on. To start, go back to the BlueHost home by clicking on the logo or clicking on "Hosting" in the top left. You should get a page that looks like this, on it, click on the "Install WordPress" button.



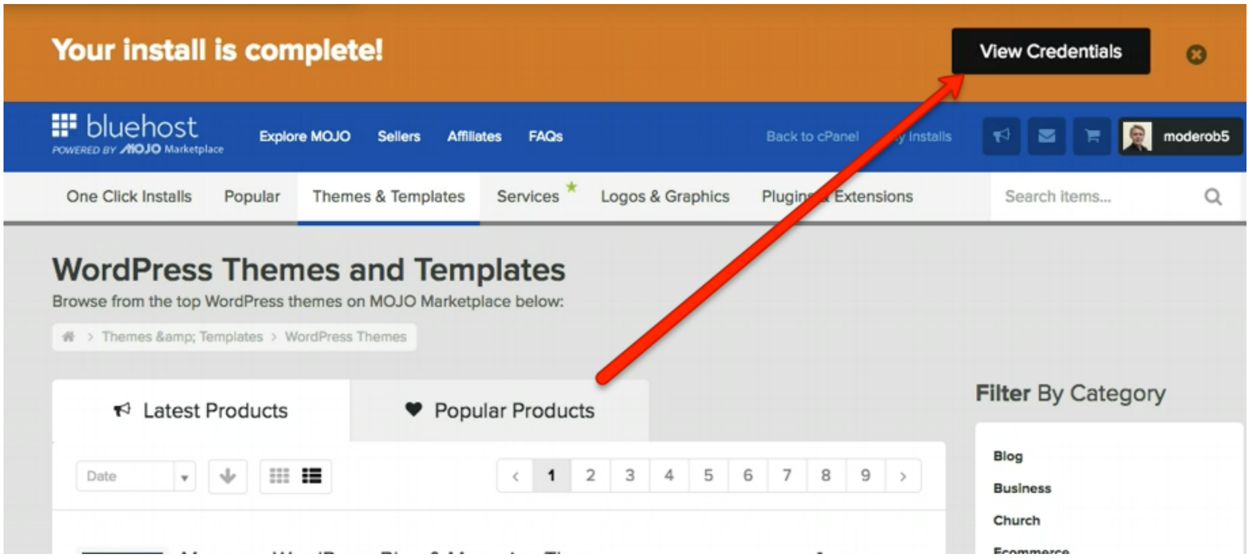
On the next page, go ahead and hit the big "Install" button.



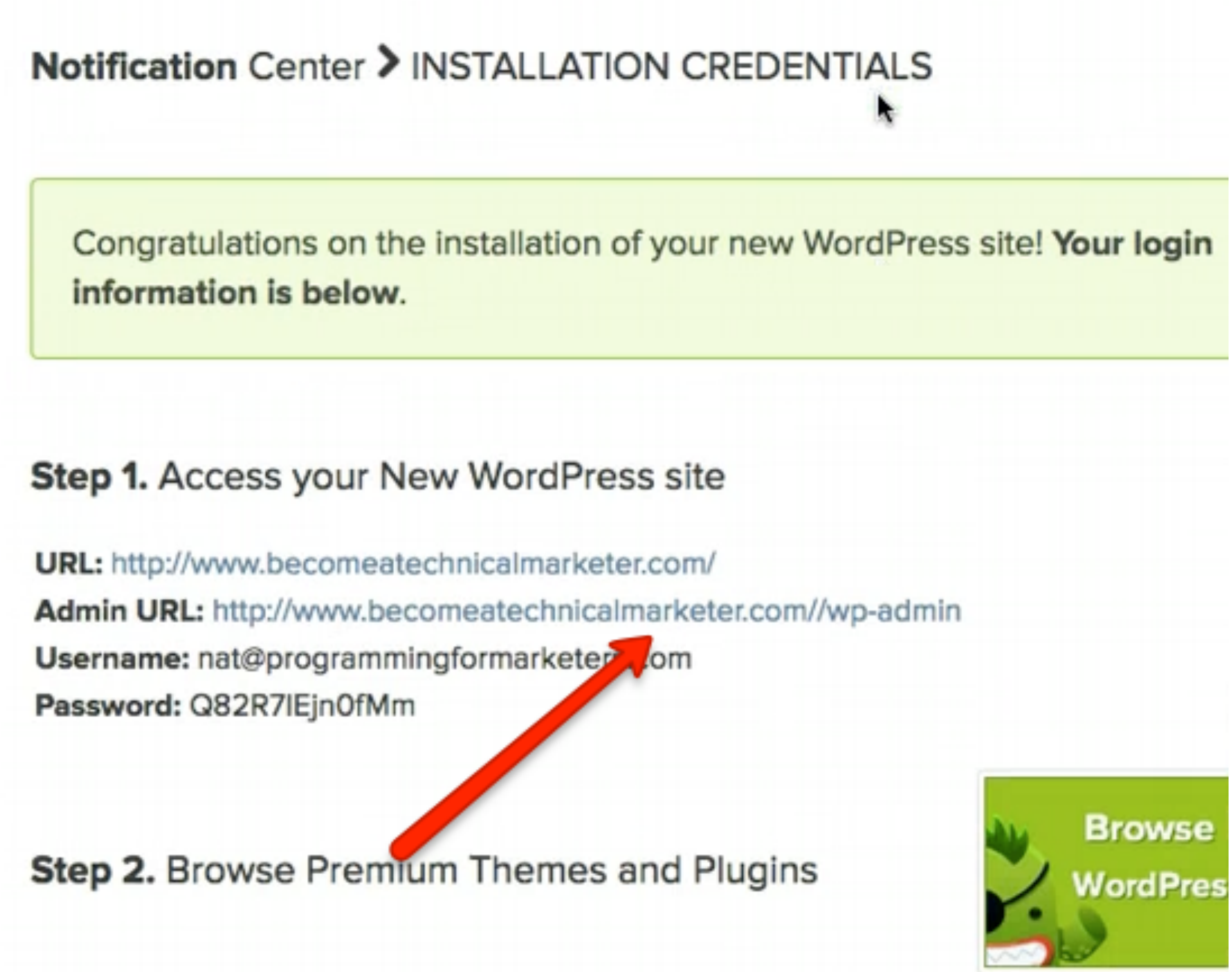
Then you'll just need to select the domain you just purchased, click "Check Domain" and then fill out your own credentials in the Advanced Options of the window that pops up.



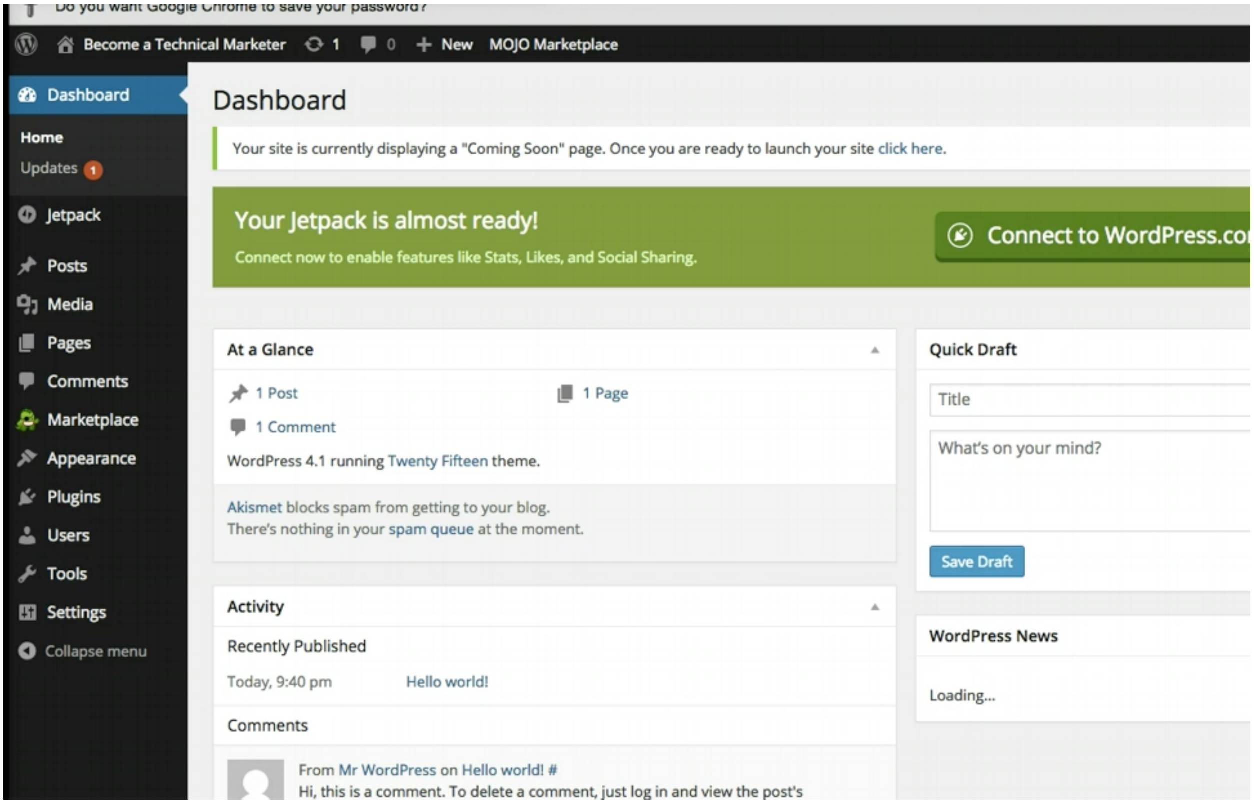
Once you hit "Install Now" you'll be taken to a page with a progress bar at the top showing you how close they are to being done installing WordPress. It should take less than a minute. Once that's done, click on "View Credentials" at the top so you can access your new WordPress site.



Now to access your site from now on, instead of coming to BlueHost, you can go directly to the URL they show you which should be http://www.YOURURL.com/wp-admin and then log in with the credentials you created.



Let's do that now. Go to the Admin URL for your site, and log in with your credentials that you created. If everything went smoothly, you'll now be in your site's WordPress Dashboard and can fully customize it! Check out your site's public URL again. Still looking pretty bare, but there's a lot more potential now.

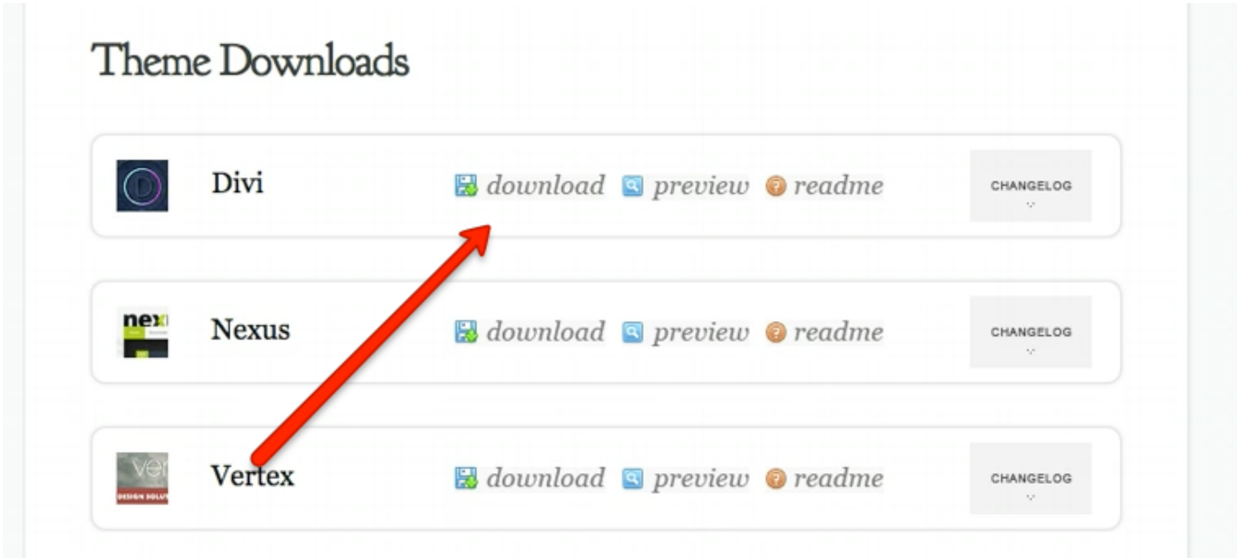


Step Three: Add a Theme

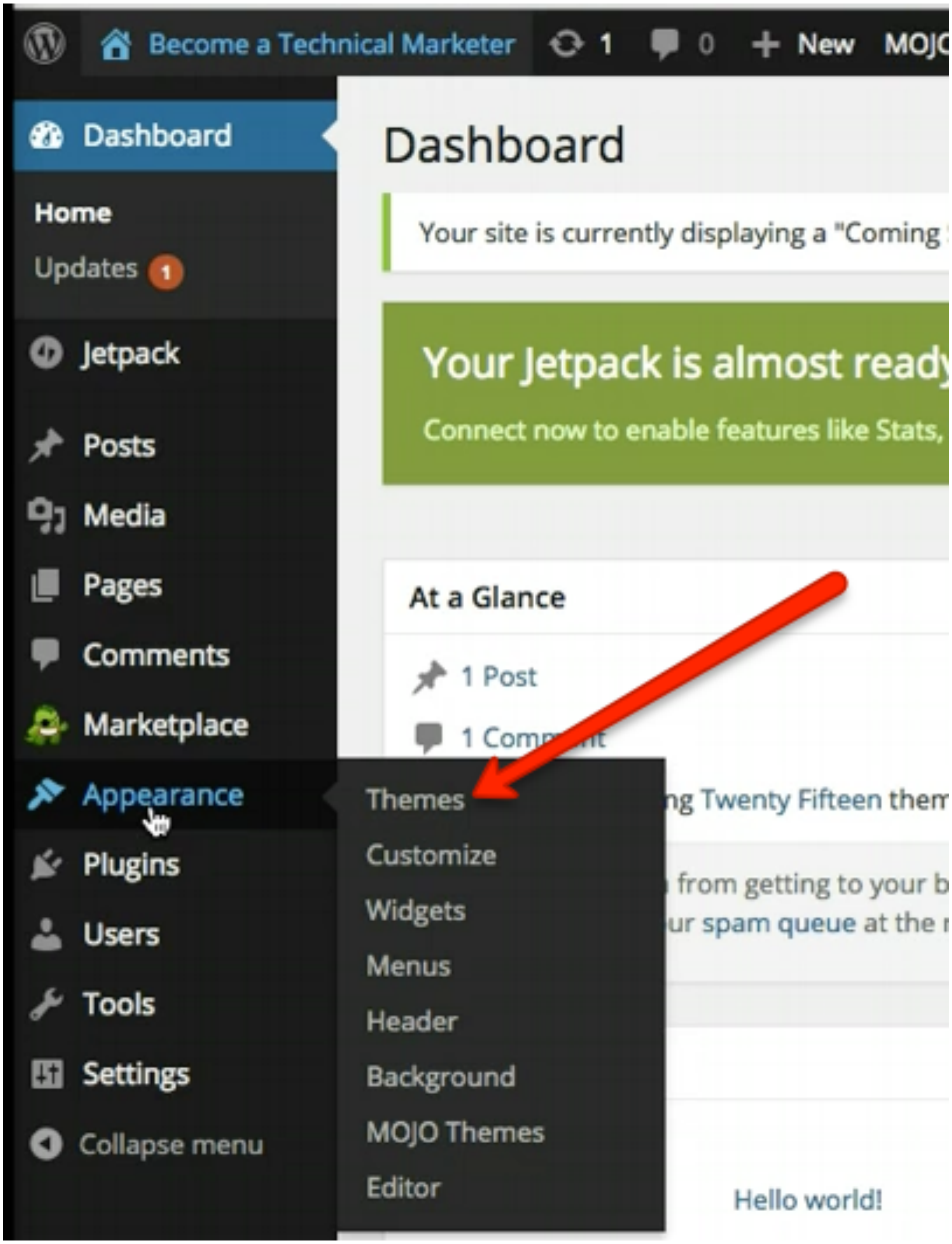
The themes that come pre-installed with WordPress aren't particularly exciting. There are some free ones out there, but for the most part the free themes leave a lot to be desired and it's worth it to just buy a good one.

I (Nat) have used Elegant Themes for years and recommend it to most people. They give you access to a ton of themes for the price of what one would be elsewhere, and their flagship theme "Divi" is the best one I've found on the market. It's what I used to design the Programming for Marketers landing page.

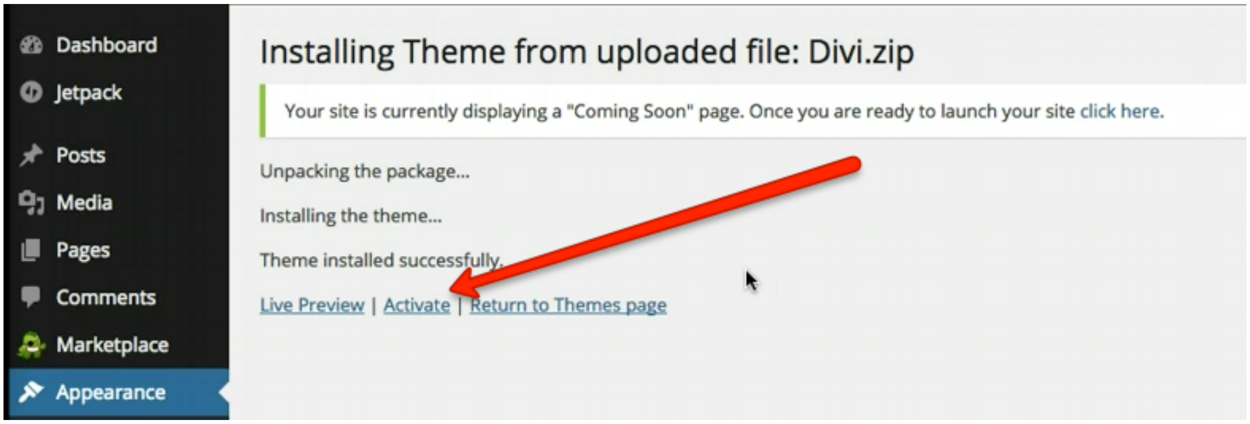
First, you need to go to Elegant Themes and sign up for an account. Once you do, you'll be able to go to the Member's Area where they have all of their themes and plugins. Don't worry about the plugins for now, what we want is the Divi theme. Scroll down to where the themes are, and next to Divi, hit "download."



It will download as a Zip, but don't un-zip it. Now you can go back to your WordPress site, and then hover over the "Appearance" tab on the left and click "Themes."



Then you just have to click "Add New" at the top, and then select the Divi.zip file for upload on the page that comes up after that. Once the Zip has been uploaded, there will be a confirmation page with an option to "Activate Theme." Go ahead and click on that.



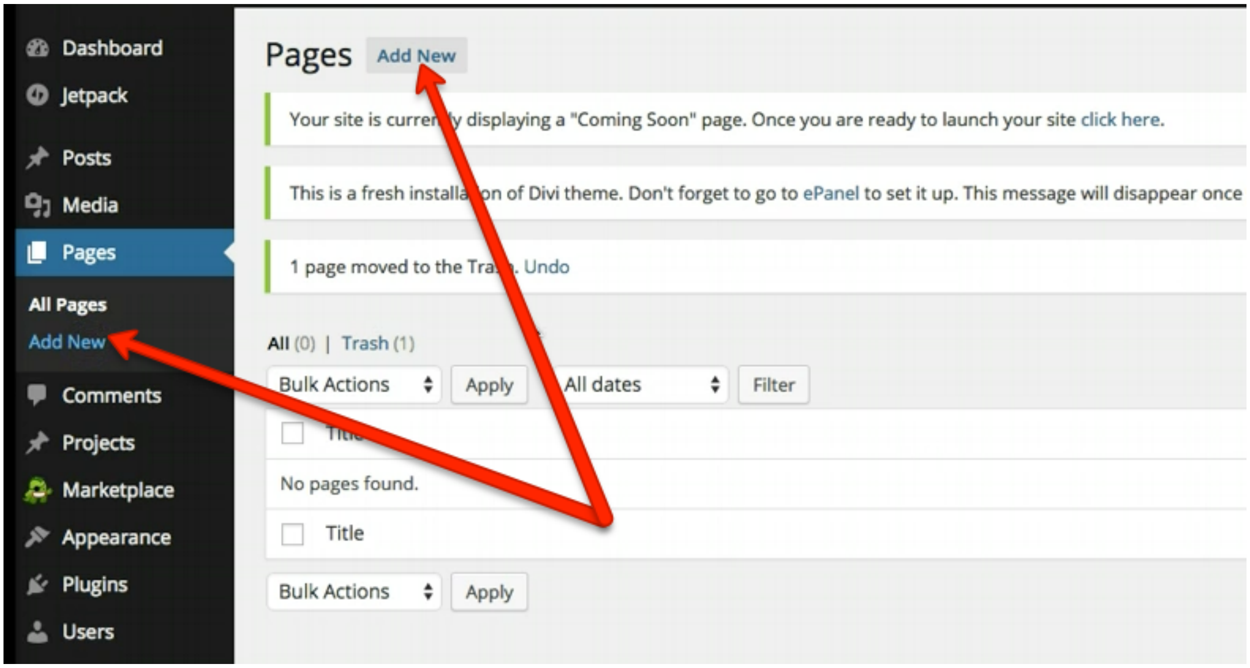
Now if you look at your website again (just type in the URL in a new tab) it should look pretty different. It won't be perfect yet, but you now have the Divi theme installed and can start customizing it.

Step Four: Customization

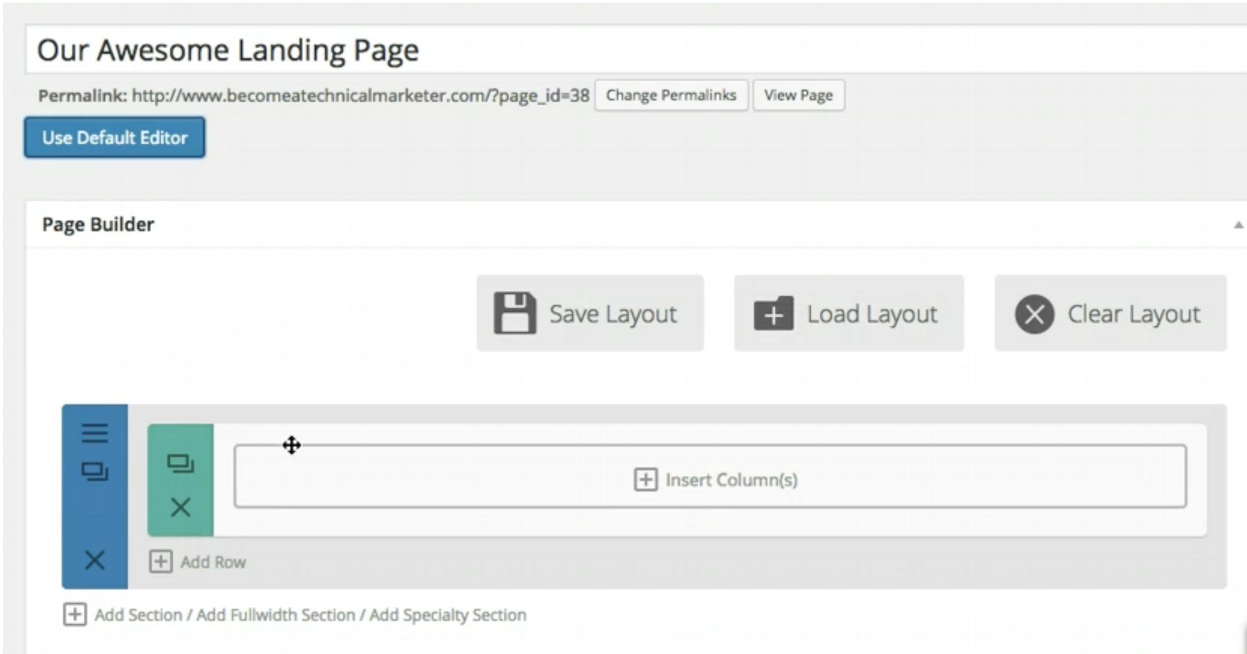
You now have a fully functional web page, running WordPress, that you can design any landing page on top of. Now it's up to you to come up with an interesting design and layout.

But Divi doesn't leave you totally on your own. If you check out the Divi "preview" page you can see some cool examples of ways you can use the theme by hovering over "Page Layouts" and clicking on different options.

To use these on your site, you should first create a page that will serve as the landing page. Go to the "Pages" tab on the left, and then click Add New.



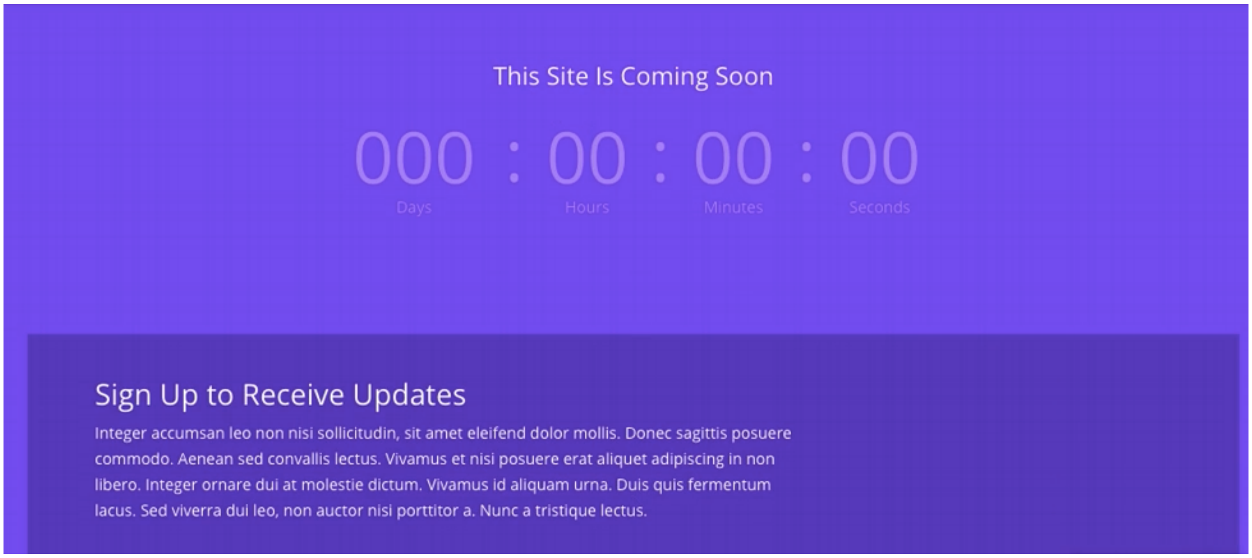
When the new page comes up, you can name it whatever you want, then click "Use Page Builder." This is a special Divi function that lets you design entire pages using a drag and drop editor instead of having to try to arrange widgets or use a set design like you would have to do with most other WordPress themes. One you click on Page Builder, it should look like this.



Now you can use the drag and drop editor to build a page from scratch, or you can load one of their existing layouts and customize it. Click on "Load Layout," and then let's load the "Coming Soon" page

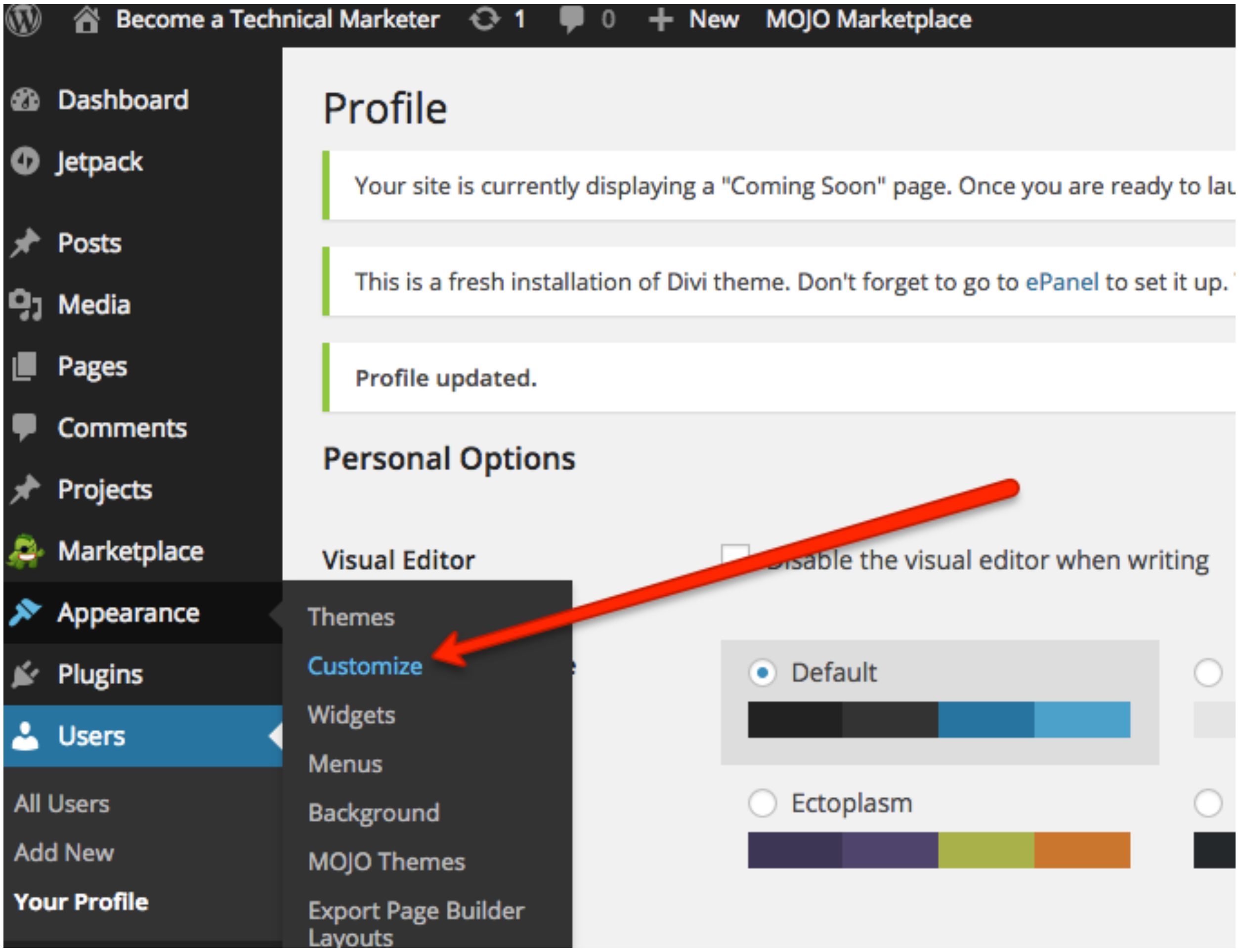
This will load in a generic version of the "Coming Soon" page. If you click on "Preview

Page" in the top right, you can see what it will look like on your site. It'll look a bit like this right now.

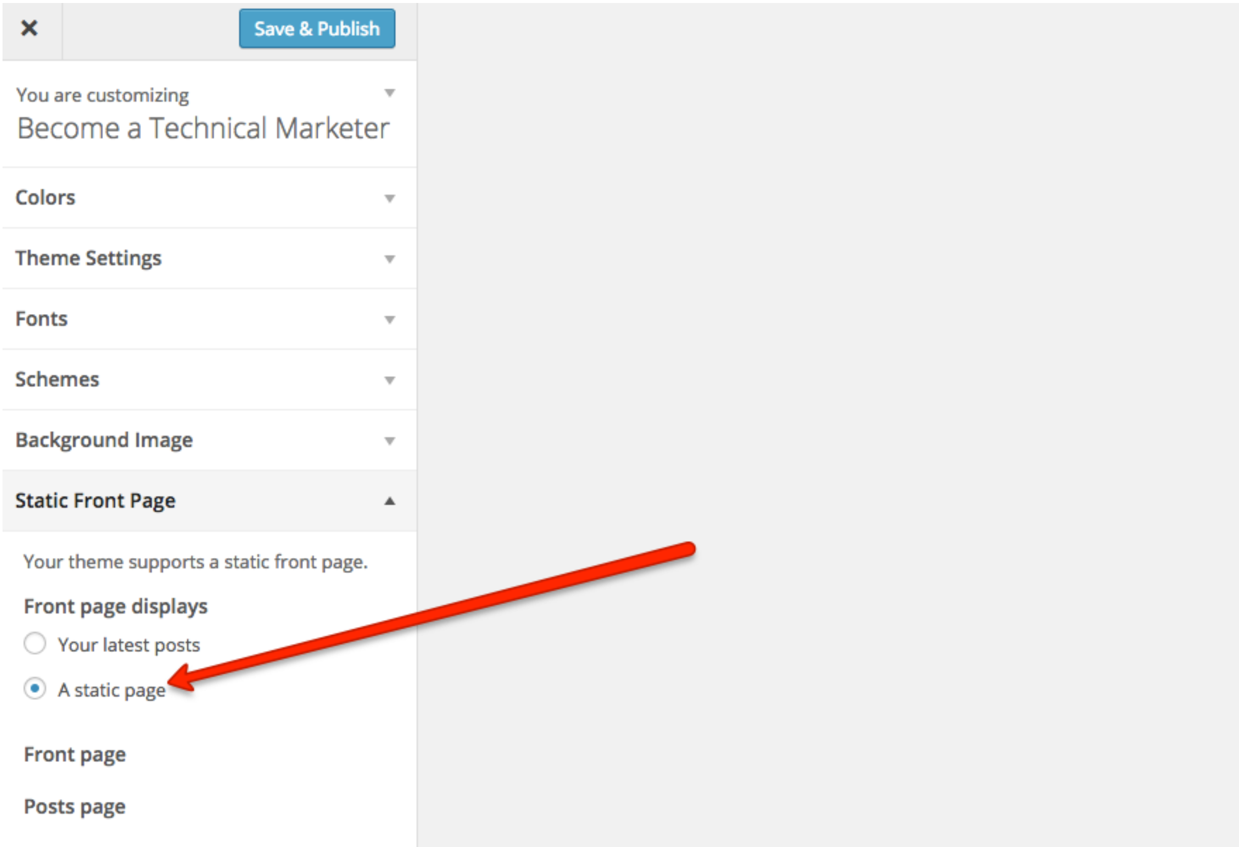


Now you can go back to the page builder and edit the content blocks to say what you want, set when the countdown timer will go to, and hook up your email marketing tool for when they sign up for updates. You can also customize the colors, add blocks, remove blocks, whatever you want really.

Once you're done, you should make this your home page. To do that, go to "Appearance -> Customize"



Then on the screen that comes up, scroll to the bottom, and select "Static Front Page" then "A static page" and select the page you just made which should be listed right below it. Once you hit "Save & Publish," the page you just designed will become your new homepage.



And there you have it! You signed up for hosting, installed WordPress, found a great theme, designed a landing page, and pushed everything live. You're now ready to test any idea that crosses your mind quickly and easily with a minimal amount of time spent on building the page.

Lesson 2: How to Create a Viral Referral Campaign

Virality is tough. Everyone wants their content to go viral, but there's no guarantee that it ever will.

At the end of the day, virality comes down to two things:

1. Having great content
2. Getting people to share that content with their friends

We can't help you with #1, but this lesson should give you a lot of ideas for ways you can do more of #2.

We'll start with some quick and easy tactics, then work our way up to the more programming-heavy content. Feel free to skip ahead if you already know something!

And if you haven't read it yet, check out the Programming for Marketers Motto. It'll help put everything in perspective.

Basics: Getting People to Share Your Content

This first section will be broken down into a few sub-sections:

1. Encouraging and Incentivizing Sharing
2. Encouraging and Incentivizing Emailing
3. Tracking Your Success

1.1 Encouraging and Incentivizing Sharing

Social media is the best place to start for creating virality. The great thing about a Facebook post or a Tweet is that you could get dozens of new signups from that one share, especially if an influencer is kind enough to share your link.

But you have to get them to share it! For the purpose of this post, we're going to focus on Twitter, Facebook, and LinkedIn. In our experience those three sources account for 80% or more of social media traffic (excluding sharing sites like reddit) and assuming you're not promoting something highly visual (like clothing, where Pinterest might be a better bet).

In general, people will be more willing to Tweet something than post it on Facebook, and LinkedIn is a toss-up depending on their profession.

The easiest and most effective way to get someone to tweet something is to use "Click to Tweet."

Click to Tweet site will give you a piece of HTML which you can enter into the raw-text of your posts or pages to create a link which will open a Twitter dialogue where they can tweet something that you've pre-written. You can also use a picture of the Twitter bird as the link. Like this one:

Click to Tweet is a good start, but you'll want to go a bit further. To the extent that you can, you want everything you're suggesting other people do to be driven by incentives. So instead of just asking nicely for someone to tweet for you, give them an incentive with "Pay with a Tweet."

This service will let you create a link that gives your visitors a bonus if they promote you on social media. It could be a PDF, video, really anything you can host on your site or on Dropbox.

And it doesn't have to just be a Tweet! You can let them share it with Facebook, LinkedIn, Google+, and a few others, and you decide which ones are allowed. The half-life of a tweet is usually less than an hour, but the half-life of a Facebook post is about a day, so you might consider only letting them post to Facebook. You'll have to experiment to see what gives you the best results.

1.2 Encouraging and Incentivizing Email Referrals

Although email doesn't scale as well as social media (one tweet might be seen by 100 people, whereas an email is only seen by one), it's still effective for generating new signups. A personal email has a much higher chance of converting since your users will typically only email people whom they're fairly confident will be interested.

If you sign up for the email course at Programming for Marketers, you'll receive a welcome email that looks something like this:

> I'm super excited to help you become a more technical marketer over the next two weeks.
>
> Until, then, I have a little bonus for you. Nat (my partner) and I pulled together a resource with 15+ automation hacks non-technical marketers can use today without writing a line of code.
>
> If you click here to email a friend about this free course, I'll send this bonus to you right away. Just leave "bonus@programmingformarketers.com" BCC'd so we know you sent it :)
>
> Love,
> Justin and Nat

Around **one third of our signups** refer a friend after receiving that email, so we had to share how to do it.

The great thing about incentivizing email referrals this way is:

1. It's personal (an email from Justin, not a robot / no-reply)
2. It gives people an incentive for emailing their friends

It's much less likely that people will blindly share your content, so it helps to have that extra bonus in case they do. In our case, it was a PDF of some great hacks you can use without any programming knowledge, that you might think you need programming knowledge for. But since you bought the ebook, you'll get it for free when you get the bonus content at http://programmingformarketers.com/ebook

Here's how you can do this yourself.

1.2.1 Set up the invitation to share

For your email list manager of choice (Mailchimp, AWeber, etc.) You'll be able to set up an "autoresponder" that activates when people join your list. This is simply an automated message that goes out at a certain time (for us, instantly) and can be used for drip campaigns as well as referral campaigns. You should be able to find a guide to setting it up with your mail client in its help docs.

We set up an immediate email to go out to new people with that text above. But there's something special about it. When you click on the link, the email is already pre-filled for you! How convenient, right? This is very easy to do.

For the URL that you attach to the call to action, you can plug in some variation of this code:

```
mailto:?bcc=bonus@programmingformarketers.com&subject=I%20think
%20you%20will%20like%20this&
body=I%20found%20this%20free%20course%20on%20becoming%20a
%20more%20technical%20marketer%20
and%20thought%20you%20might%20be%20interested%20-%20
http%3A%2F%2Fwww.programmingformarketers.com%2F%0A%0AIt
%20teaches%20you%20how%20to%20use
%20technical%20and%20automation%20skills%20to%20become%20a
%20better%20marketer.%20I%27m
%20taking%20it%20and%20think%20it%20could%20be%20really%20cool.%0A
%0A
```

Here are a few important things about it:

- You can put your own email in the "BCC" field, or you can do a "to" or a "CC" field instead (as well as in addition to)

- You'll want to put in your own subject. %20 stands for a space (since you can't put a space in a URL)
- "Body" is the body of the email. Change the text there to change what gets sent

You can build it manually based on that one above, (tedious) or just go here to have W3Schools create the text for you (much easier).

1.2.2 Set up the bonus

You just need to find a PDF, Video, or something else you think your readers / users will like that you can send to them. Once you've found it, (or better yet, created it) host it either on your site at a private URL (like www.programmingformarketers.com/handouts/secret-doc.pdf) or put it in your public DropBox folder then copy the sharing link (easer, and the way we did it).

Whichever you choose, hold on to that URL for step 3:

1.2.3 Set up the reward

To be honest with you, we skipped this step in the beginning. We weren't sure how many people would do the email referral, so we figured Justin could just email each of you with a link to the PDF after you shared the course with a friend.

BIG MISTAKE. We blew our signup goal out of the water, and Justin woke up the day after we launched with 400+ emails in his inbox and it kept growing. We had to automate it quickly.

First, you'll want a separate email address specifically for the giveaway. We used "bonus@programmingformarketers.com"

Second, you'll want that email address to be the one CC'd or BCC'd in step 1.

Third, you'll want to create a rule in Gmail to apply a tag to emails going to that address that match criteria from your suggested share text. Or maybe just whenever that email address is CC'd or BCC'd. If you're not sure how to do that, check out this guide to Gmail filters.

Fourth, you'll go to Zapier and create a Zap for automatically replying to emails that have that tag on them with some thank-you text, and a link to the giveaway. This way you don't have to manually respond to each one, and you'll save a ton of time. Here's a link to the template we created for doing this.

And you'll be good to go! Now whenever someone signs up, they'll be asked to share your site, and be given a reward for doing so. And it all happens automatically.

1.3 Tracking Your Success

Tracking your links and how well they're working is **critical** for optimizing your different marketing campaigns, whether they're PDFs, Emails, or anything else.

There are two easy ways to do this: URL shorteners, and UTM codes.

URL Shorteners

A URL shortener is any service like Bit.ly that will take a long link and then shorten it for you, such as this: http://ctt.ec/gNw0X, or a service like Buffer that will share content for you with the links automatically shortened.

The great thing about these shortened URLs is that not only do they fit better on Twitter, but they come with analytics attached so you can see how many people are clicking on the link! You can also see where those people are coming from, how it's changed over time, and more, which is useful for seeing how different campaigns of yours are performing.

UTM Codes

UTM codes are similar to link shorteners, but they make the URL longer instead of shorter. What's great about UTM codes is that they add identifiers to the URL such as the "campaign" the "source" and more so that if you're in Google Analytics or another analytics provider you can see which campaigns are driving the most traffic.

We like to add UTM codes to the URLs in this course so that if we refer you to a certain site frequently (like Buffer, Zapier, etc.) in time they'll notice that we're driving them a lot of traffic not just from our site, but from our emails and PDFs as well. You can also use them for your own analytics to see how well your different content strategies are working, like how many people come to your site after reading a certain slideshare.

You could add them manually, but it's much easier to just use Google's UTM generator.

Alright, let's move on to some bigger stuff.

Advanced: Creating a Master Referral Page

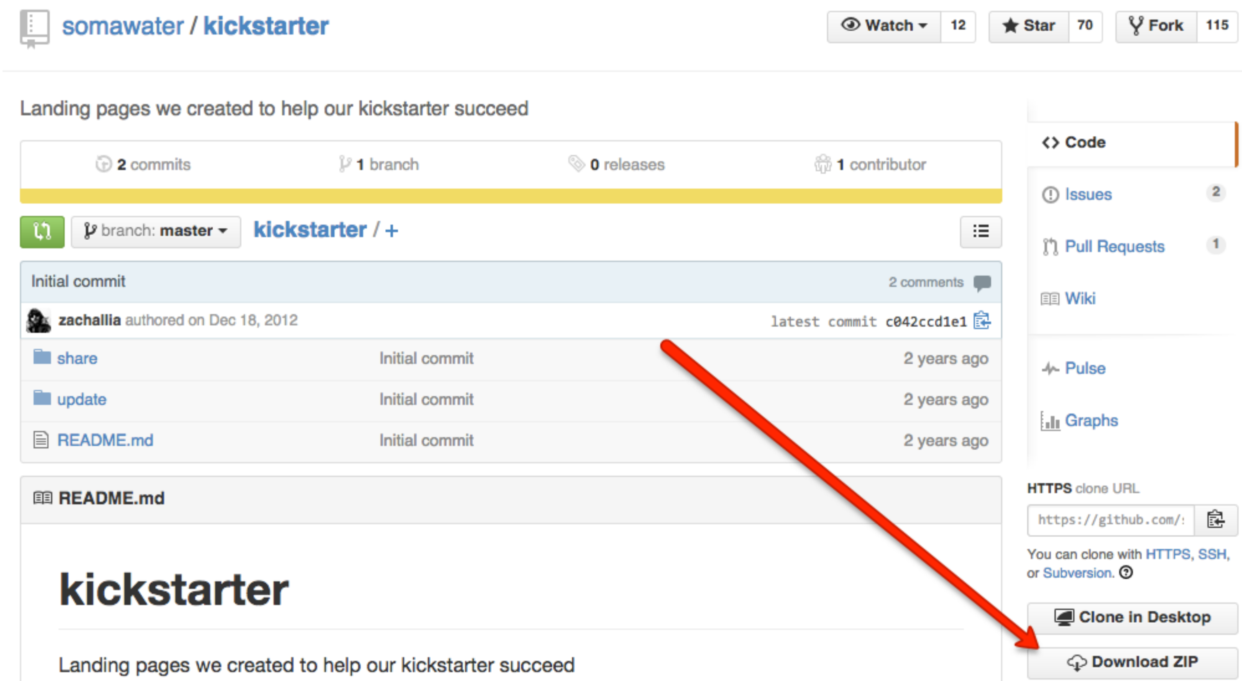
SOMA is a water filter that entered the market about 2 years ago after a very successful Kickstarter campaign where they raised ~150% of their goal.

Afterwards, they were kind enough to open-source the code they used to build the referral page that helped them attract most of those buyers.

Now we'll teach you how you can implement it on any site to incentivize different kinds of shares, and help boost your product's reach significantly.

2.1 Getting the Code

This is the easiest part. SOMA open-sourced their code on GitHub, so all you have to do is go to this URL: https://github.com/somawater/kickstarter and click "download ZIP" in the bottom right.



Once that's done you'll have a ZIP folder in your downloads that you can extract to a new folder, and in that folder you'll have a README, a "share" folder, and an "update" folder. Ignore the README and the "update" folder for now, and go ahead and open the "share" folder.

Select the four items in "share" and open them with your text editor that we set up in the "Setup" section.

2.2 Customizing the Code

This code is still set up for SOMA's referral page, but it's easy to adapt to your own purposes.

You'll notice three types of files: .html, .css, and .js. In order, those are:

1. The HTML file which determines the high-level structure for the site (how things are ordered).

2. Two CSS files that determine how the site looks on people's browsers of phones. HTML sets the order, and CSS sets how it looks.

3. A Javascript file which makes the buttons on the site do things. We'll explain what they do when we get to that file.

If you don't know any HTML, CSS, or JS, don't worry! We'll walk you through each file so you can get an understanding of it and so you know what to change. But if you do want to learn more HTML/CSS/JS, we recommend Codecademy which has some great courses introducing you to the concepts, and which you can finish in a weekend.

First, to edit the HTML, you'll need to change these lines:

1. Line 4 for the title of your site
2. Line 12 for the header of the site
3. Line 15 for the body text. A note here: anything inbetween and will be bolded,
 adds a blank line, and anything between <p> and </p> is just paragraph text. You can look at other HTML operators on W3Schools
4. Line 29 for the Facebook text (just what's on the button)
5. Line 34 for the Twitter text (just what's on the button)
6. Line 39 for the Video text (if you don't want to link to a video, we come to that later)
7. Line 44 for the copyright

Second, to edit the Javascript, you need to change these lines:

1. Line 18: "'http://www.facebook.com/share.php?u=http://bit.ly/somawater'" change the url after the "=" to the URL you want them to share
2. Same for line 20
3. Line 25: Change the Twitter share link and the title (line 26)

4. Line 35: Change the URL for the video

Okay, now if you were to load all of those files in your browser or through brackets you'd have a fully functioning referral page! But maybe you want to take it a step further? You can also:

1. Customize the CSS to fit your design scheme, in share.css and mobile.css

2. Change the third button to something besides a video, such as an email referral. [Here's an example of how to do that.]

2.3 Uploading the Page

Now you just have to upload it. You have a few options for this:

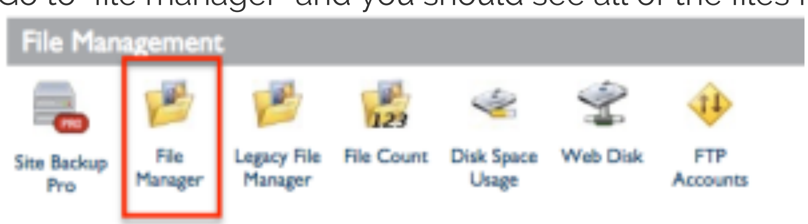
If you built your website from scratch, simply add it to the sitemap.

If you're working for a startup or larger company and someone else manages the site, shoot this over to them, maybe have design touch up the visuals of it, and let them push it live.

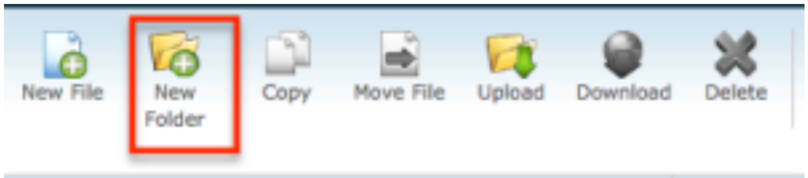
Or if you're using a service like Wordpress, you can manually add it as well. Here's how:

1. Go to your hosting provider (we use [BlueHost], but the steps will be similar for other provides.) If you set up a landing page in Lesson 1, then this will work for you.

2. Go to "file manager" and you should see all of the files for your site



3. Create a new top-level folder called "referrals" or whatever you like. If you stop there then the link will be "www.yoursite.com/referrals". Add more folders to get the referral path you want



4. Upload all four of the files to that folder. Go to the URL you just created, and your page will be live! Very cool :)

Now you know how to set up your individual sharing incentives, as well as how to set up your own page for asking for shares from your audience.

In the next lesson we'll show you how you can use some different scripts to automate generating new Twitter followers, which will significantly help with sharing your content and demonstrating that you have social authority.

Lesson 3: How to Programmatically Acquire 50k+ Twitter Followers

In this lesson we'll cover what you need to know to grow your Twitter following as organically as possible, without investing a huge amount of time.

In the basics we'll cover tools that can do some of the work for you, and then in the advanced section we'll show you how to build most of them yourself.

Let's get started!

Basics - Tools and Techniques

The basics will be broken down into five sections:

1. Setup
2. Following people
3. Favoriting
4. Joining Conversations
5. Timing

1.1 Setup

Before you do anything to try to grow your following, **you need to have a Twitter account that's worth following.**

That means that when someone sees you pop up in their feed they're interested in learning more about you and can clearly see why they should follow you.

This breaks down to three areas: Content, Bio, and Pictures.

1.1.1 Content

You have to create a unique voice on Twitter. I bet you don't like following people who sound like robots, so you can't sound like one either. Your word choice, the content you share, the people you retweet, it should all fit the voice of you and your Twitter account. If you like to swear, swear! Don't feel like you have to follow any arbitrary rules about what to post aside from **be yourself**.

Some parts of the content to make sure you're keeping consistent:

- People you retweet
- Articles you tweet
- Quotes you tweet
- Personal anecdotes you tweet

1.1.2 Bio

The first thing that someone will look at when you show up in their feed, on their followers list, or elsewhere is your bio. You only get 160 characters to tell the world who you are, so make them count! Don't do a boring list like: "Entrepreneur, growth hacker, writer, marketer, lover of technology." It sounds generic and dull.

Here's [Justin's Twitter bio](#):

> Co-author of Traction Book, the guide to getting startup traction. http://tractionbook.com.

It's clear, to the point, and highlights the thing that he's most likely going to be discussing on Twitter (since a lot of his followers came from the book). Make sure that you have a similarly clear Twitter bio that focuses on a couple of the most important things about you, while making you seem interesting and inviting.

Feel free to add a couple hashtags, but don't over-do it. Hashtag soup is a big turn off and people will assume you're going to spam them with articles they don't care about.

Last, make sure that you have a location (people sometimes look for others in their area) and a URL. If you don't have a personal site, you could put your LinkedIn or a link to your profile on a company page.

1.1.3 Pictures

Have a nice profile picture and header that show a bit of your personality. Keep it fun, and don't make it overly promotional. If you're using Twitter for a company then you should have the logo and a header that represents them. Feel free to use one CTA in the header, but don't make it sound spammy or it will turn people off.

Now that you have a great Twitter account, let's build some followers for it.

1.2 Following People

Assuming you're not a celebrity, published author, startup founder, journalist, etc. you're not going to have a ton of exposure where people will naturally want to follow you. That means that to build your following, you have to get in front of people and pique their interest. The most reliable way to do that is to follow people.

Anywhere from 10-50% of people will follow you back if you follow them. A small portion of them will be using auto-follow-back programs, but a lot of them will just follow you back because you seem interesting and they want to return the favor.

But you don't want to follow randomly. You should only be following people who would be good candidates for being interested in your content, and who will be most likely to follow you back. The best way to do this is to try to "steal" other people's followers. Here's how it works:

1. Go through your followers, find people who write about similar things as you and who have a lot of followers

2. Go to their page

3. Start following their followers in the hopes that they'll follow you back

But doing it manually sucks. Instead, check out [Tweepi] and [Manage Flitter,] which both offer services to make it quick and easy to follow a lot of people at once, and to target who you're following.

For example, with the paid version of Tweepi, you can pick a user who's followers you want to follow, then add filters like:

- Only people who are following more people than they're being followed (these people will be more likely to follow back)
- Only people who've tweeted in the last few days (no inactive accounts)
- Only people who have at least 50 followers (less chance of spammers / totally new accounts)
- Only people who I haven't followed / unfollowed before
- And so on

The other important step is to unfollow those who don't follow you back. You won't have a 100% success rate so it's important to prune the people who don't reciprocate. Tweepi and Manage Flitter can do this for you, and we'll give you some code to automatically do it later. Or you could do it manually... but that's no fun :)

WARNING: Don't get too eager with your following! For more active, established accounts, Twitter will let you follow up to 1000 people a day. For a brand new account you shouldn't do more than 100, and you may never want to go much above 200 to be totally safe. Twitter does give you one warning though, so if you want to see how far you can push it go right ahead!

1.3 Favoriting Tweets

Before we explain this, go to Twitter, and tweet something like:

"Test tweet: I love growth hacking, startups, and growing my social media following" (click it to tweet)

Here's what happened when I posted that:

Those 18 favorites showed up in less than 5 minutes--that's an absurd engagement considering none of these people follow me.

Sadly it's not really engagement. All of these accounts are using auto-favoriting scripts or programs to find people tweeting about certain topics, then favoriting their tweets to hopefully get them to follow them.

Depending on your industry, this is a very effective tactic. It's good in an area less inundated with "social media gurus" (for example, fine tea curation is probably low-competition) but it's bad in the case above where everyone and their mother in the growth hacking arena is running an auto-favoriting script. I'd say avoid doing it for things like social media, growth hacking, startups, and anything tangential to those.

Even though it's not quite as effective as following people, it is nice because it doesn't require you to inflate your following number. That would be desirable if you don't want to look like you have an artificially high follower count just from doing follow / follow back.

Again, you could do it manually, but tools are better and we'll show you how to write code for it later. In this case you might want to use a tool like Twitfox. They'll let you pick certain keywords to track, and then they'll go out and favorite tweets related to those topics for you. But we'll also teach you how to write a script to do it.

Try it for a bit, see how it converts, change your keywords around, and go from there!

1.4 Joining Conversations

Another way to gain exposure is to join conversations on Twitter. These conversations revolve around a certain hashtag, and if you look at a company's Twitter page you can usually figure out pretty quickly if they do a regular Twitter chat.

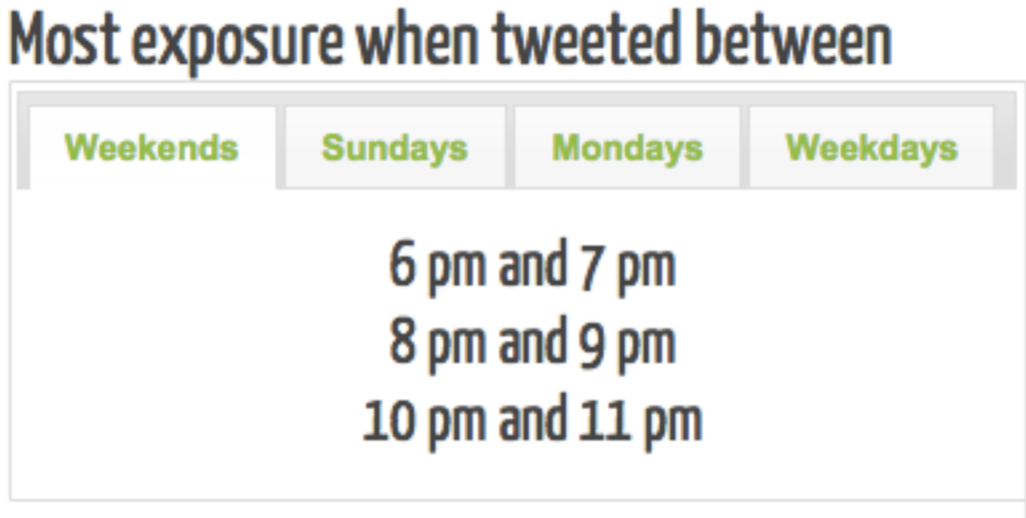
Although you won't get as many followers from Twitter chats, you will get your name out in the field and you might be followed by some high influence people. If you're followed by someone with influence, they're a great candidate for "stealing" followers from (a la 1.2) since when you follow their followers, their followers will see that the influencer they're following follows you and assume that means you're important and an influencer yourself.

There are also some conversations that aren't hosted by companies but that come up organically. Search for certain hashtags related to your industry and see what comes up!

1.5 Timing

Those three things will be the main way you get discovered, but if you want to keep your followers, you have to not only have good content but well timed content.

In order to do this, you can combine Tweriod and Buffer to build an optimal sharing schedule.



First, go to Tweriod, sign in with Twitter, and see when your followers are most active.

Then, take that schedule, go to Buffer, and set your tweets to go out during those high-activity times.

Now so long as you post your tweets through Buffer **you're guaranteed that they're hitting your audience when they're most active!**

Alright, so that's the basics. Now let's cover how you can do some programming and do all of this on your own (and for free!)

2. Advanced - Running Your Own Twitter Scripts

WARNING: These techniques can be a bit risky! Don't accidentally follow 1000 people at once (though we did our best to make that difficult to do). And there's always the chance that you could get banned if you use them too much, so be smart! Play it safe and slowly escalate to what you're comfortable with. Remember the limits from before.

In this section, we'll cover:

1. Installing the Twitter tools for Python

2. Favoriting with Python

3. Following with Python

4. Un-following with Python

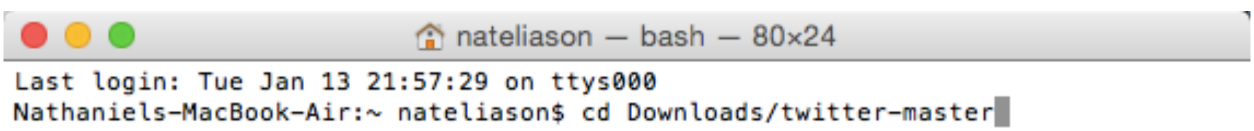
2.1 Getting Set Up with Twitter's API

Make sure you have Python installed as we covered in the “Setup” section. Once that’s done, you need the code for working with Twitter. Head to this link https://github.com/sixohsix/twitter/ then:

1. Click "Download ZIP"

2. Unzip it to your Downloads folder

Once it's in your Downloads folder, go to Terminal (or Powershell if you prefer on Windows) and type in: "cd Downloads/twitter-master"



Assuming you left it in your downloads folder, you'll now have "navigated" to the twitter-master folder using Terminal.

"CD" tells the prompt where you want to go within your system, but it can only go down and up within folders. If you wanted to go back up a folder, you could type "cd //" and then you'd be in "Downloads" instead of "twitter-master". Stay in twitter-master though.

Now, in terminal, type "python setup.py install". This will take the twitter-master code and add it to your Python installation so you can access it from other Python scripts.

Don't worry about all of the text that appears. Assuming nothing went wrong you should now have the Twitter tools installed!

twitter-master — bash — 80×24

```
Processing twitter-1.15.0-py2.7.egg
Removing /Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-pa
ckages/twitter-1.15.0-py2.7.egg
Copying twitter-1.15.0-py2.7.egg to /Library/Frameworks/Python.framework/Version
s/2.7/lib/python2.7/site-packages
twitter 1.15.0 is already the active version in easy-install.pth
Installing twitter-log script to /Library/Frameworks/Python.framework/Versions/2
.7/bin
Installing twitter script to /Library/Frameworks/Python.framework/Versions/2.7/b
in
Installing twitterbot script to /Library/Frameworks/Python.framework/Versions/2.
7/bin
Installing twitter-follow script to /Library/Frameworks/Python.framework/Version
s/2.7/bin
Installing twitter-stream-example script to /Library/Frameworks/Python.framework
/Versions/2.7/bin
Installing twitter-archiver script to /Library/Frameworks/Python.framework/Versi
ons/2.7/bin

Installed /Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-p
ackages/twitter-1.15.0-py2.7.egg
Processing dependencies for twitter==1.15.0
Finished processing dependencies for twitter==1.15.0
Nathaniels-MacBook-Air:twitter-master nateliason$
```

Now we just need access to the Twitter API. The API is the "Application Programming Interface" which is basically a fancy way to say "the data." By having access to the API you're really saying you have the ability to use Twitter as if you were on www.twitter.com, but with fewer birds and more lines of code.

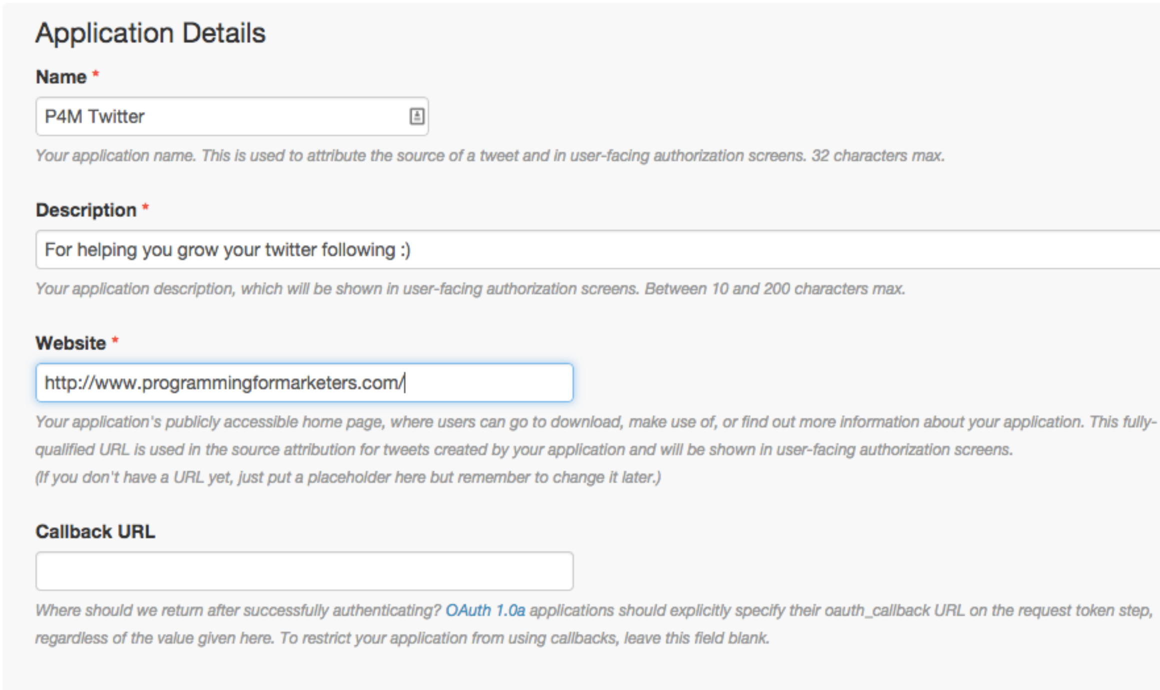
Here's how you get API access:

1. Create a new folder (anywhere you want) to store the code for this project in

2. In that folder, create a new file called "twitter_info.py" based on our example here: https://github.com/ProgrammingforMarketers/grow-twitter-following

3. Now go to https://apps.twitter.com/ and click "Create a New App"

4. Put whatever you want in the name, description, and URL. Don't worry about the callback



5. Once you hit accept, go to "permissions" and set them to "Read and Write" not just "Read Only"

6. Now go to the "Keys and Access Tokens" tab, and generate a new access token (there will be a button towards the bottom)

7. Once you have that, you can copy over your Consumer Key, Consumer Secret, Access Token, and Access Token Secret to the file you created, twitter_info.py, based on our set up

8. Save the file, and you're done! Your computer is now set up to work with your personal Twitter account

2.2 Getting the Code for Automating Twitter

Now the real fun can begin. We're going to download a Python package that someone was kind enough to open source so that we can manipulate our Twitter accounts without having to build everything from scratch.

You can find the original code here: https://github.com/rhiever/twitter-follow-bot but we're going to recommend you download our slightly modified version of it which will allow you to keep your personal Twitter info in the separate file you created. This is a good practice in case you share the code with someone else, since you don't want to give them access to your Twitter account. Go here to find our version: https://github.com/ProgrammingforMarketers/grow-twitter-following

First, hit "Download ZIP," unzip it to your downloads folder (unless you did this already earlier for the "twitter_info.py" file) and then copy the "twitter_follow_bot" and "sample_twitter_codes" files to the same folder as where you put your "twitter_info.py"

And that's it! You can now run a bunch of different functions to automate your Twitter account, just by changing what's "commented out" in the "sample_twitter_codes" file. We'll explain in the next few sections.

2.3 Auto-Favoriting with Python

Now instead of having to pay for someone else's service, you can run your own auto-favoriting robot right from the command line.

Here's all you have to do:

1. First, open "sample_twitter_codes" and un-comment #3 (we explain commenting in the file)

```
'''
#3 This code will let you favoite things that are relevant to you. Just replace "phrase" with the
you want to favorite for, and set the count to how many things you want to favorite.
'''

from twitter_follow_bot import auto_fav
auto_fav("phrase", count=100)
```

2. Then, open your terminal and navigate to where you stored the .py codes using the same "cd XYZ/XYZ" syntax from before

3. Last, type in "python sample_twitter_codes.py" and hit enter! You'll start favoriting everything related to that topic to a certain limit.

2.4 Auto-Following with Python

So this starts off the same as for auto-favoriting, but you'll either un-comment #1 if you want to follow based on a phrase, or #2 if you want to follow someone's followers. Once you've uncommented one, save the file again.

```
'''
#1 Here you can automatically follow people who tweet about a certain phrase. Just replace the phrase
with something relevant to you! Also you can set the count to whatever makes you most comfortable.
'''

from twitter_follow_bot import auto_follow
auto_follow("phrase", count=100)
```

Go to terminal, navigate to the folder, and type in "python sample_twitter_codes.py" and you'll start following!

2.5 Auto-UnFollowing Non-Followers

For this free course we won't go too in-depth with modifying the python files, but we did include the function to wipe out all of your non-follow-backs. You can un-comment and run #4 if you're feeling aggressive.

And there you have it! You now have a set of Python functions set up for your personal (or company) Twitter account that lets you automatically favorite relevant tweets, follow people who would be good leads for you, and even unfollow people who aren't following you back. [Go brag about it on Twitter!]

Lesson 4: Tools and Scripts to Capture More Emails and Engage Your Signups

So far we've covered some strategies for setting up a viral referral campaign, as well as ways to programmatically increase your twitter following.

In this lesson we're going to talk about how you can significantly improve your email capture rates through existing tools, and then in the advanced section we'll talk about programmatically sending custom HTML emails.

1. Basics - Getting More Email Addresses Using Tools

We're going to do things a little differently in this lesson: we're going to break it down by how you can get more email addresses over the life of anyone's visit to your site.

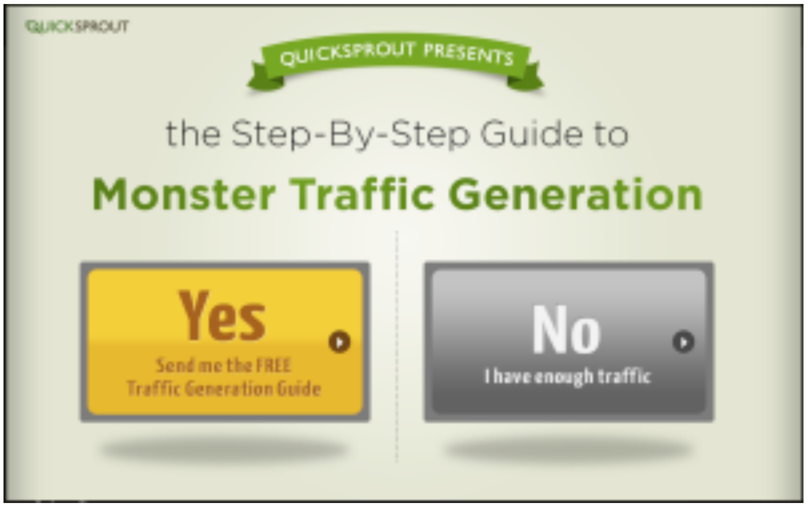
First, the arrival.

1.1 Getting Email Addresses Upon Arrival

The simplest and most obvious way to boost email signups right at arrival is through email popups.

Though somewhat annoying, they do work, so long as you word them well and provide a clear reason why someone should complete them.

QuickSprout does a great job of this. They make it clear as day what you're going to be getting when you sign up, and since it takes over the entire page it's very hard to simply ignore and click out of.



Unless people are already arriving at a landing page, it will help your opt-in rate a lot if you set one of these up. You have a few options. QuickSprout uses Bounce Exchange, but we're going to recommend SumoMe for reasons that will be apparent in a second.

If you're on WordPress you can add SumoMe directly from the plugins. If you're on a custom site, all you have to do is install this code in the <head> of your site.

Once you install SumoMe you'll be able to very easily set up a popup for when your visitors arrive. To make it less annoying, we recommend:

- Adding a delay so it doesn't show up immediately
- Making sure there's something in it for the user, not just a "hey sign up plz"
- Have it wait a day before showing up again for the same user

The benefit of SumoMe is that while you're at it, you can also add a "Scroll box" which creates another popup when they reach the bottom of the page. And finally, you can add a top bar that has a similar call to action, which means you're getting their attention in three different places. Ryan Holiday does a good job of using 2/3 of these in a very un-annoying way.

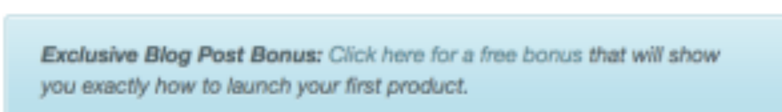
1.2 Getting Email Addresses On-Site

Alright, we've got pop-ups out of the way, let's move on to some more interesting stuff.

With pop-ups you can only expect to get ~1-5% conversion rate. Maybe in the 7-10% range if you're highly respected or have an awesome CTA.

But Bryan Harris of [Video Fruit](#) **averages a 20% conversion rate! WTF?**

According to him, the #1 thing that does it is **lead magnets.** A "Lead Magnet" is an additional part of the article that requires people to take some action in order to get access to. If you've noticed, we use one when we ask you to email your friends about us and we used one in the last email for some extra code to automate your twitter functions.



Lead Magnets are simple. In addition to your blog post, you create anything related to it that the reader might find useful. This could be a video of you walking through it, a PDF explaining it more in-depth, a spreadsheet that will help them get set up faster, etc. Then you embed it in the post in a way that requires the reader to enter their email address in order to gain access to it.

There are a few ways to do this. You can use SumoMe's "Leads" plugin which might be easiest since you already have it installed. If you want a bit more customizability though, you can also check out LeadPages which is what Bryan Harris uses for all of his. Both options will let you easily implement a lead magnet into your blog posts, and increase your conversion significantly.

1.2.1 Video

There's a third option. If you wanted to, say, put up an instructional video on how to do what you talk about in the post, Wista is the best option for that. They're a more SEO friendly video hosting option than Youtube, they let you seamlessly embed videos in your site, and they'll let you require visitors to enter their email address before watching. Once someone enters their email address, Wistia will automatically add them to your email list manager.

1.3 Getting Email Addresses When People are About to Leave

You'd be amazed at how many people you can convert at the last second by installing a simple exit script. When your page detects that someone is moving their mouse quickly towards the "X" in the top, it brings up a pop-up encouraging them to sign up again.

You might not want to combine this with the welcome popup to avoid being overly aggressive. If you're choosing just one, then you might want to just use this one since it lets them look around for a bit first. This is also a good option on landing pages where you don't want a pop-up at the beginning, but you do want a popup if they didn't submit their email address.

Exit Monitor will help you get set up on custom sites, and wBounce is a plugin for Wordpress that will do the same thing.

This is what shows up when you get too close to the X button on Exit Monitor

1.4 Getting Email Addresses Elsewhere

There are some other ways you can capture email addresses, not directly related to people showing up, engaging with, or leaving your site.

1.4.1 Giveaways

Giveaways are a fantastic way to get more email addresses, and they're extremely simple to set up. Pat Flynn explains it best here. He grew his email list by a massive amount (200,000!) just from doing a simple giveaway.

The easiest way to get set up with one if you're on Wordpress is to install "KingSumo" which will let you set up a private page specific for your giveaway. Just for you guys, we got a special KingSumo referral code that will give you **51% off**. Why 51%? Because everyone else was giving away 50%! Just enter "programmersrule" at checkout.

If you're not on WordPress you'd have to code your own, and while we'd love to explain how to do it, this would turn into a very long lesson.

1.4.2 Slideshares

[Buffer](#) does a great job of this. They put a number of great presentations up on [Slideshare](#) and then at the end they have a CTA to signup for their email newsletter. It's a simple way to generate more email addresses off-site, and is also a great way to build links to your site.

Slideshare lets you add the CTA automatically, so all you need is the presentation to put up there!

1.4.3 Text Messages?

This is the oddest one of the list, but say you're at an event, you meet someone cool who's interested in your work, and they want to join your email list. Well you can actually add them by text message, and we'll show you how at www.programmingformarketers.com/ebook if you haven't already signed up.

2. Advanced - Sending Personalized HTML Emails for

New Signups

If you signed up at www.programmingformarketers.com/ebook, then you'll have received the PDF with a bunch of automations you can use to automate tasks you might be doing manually.

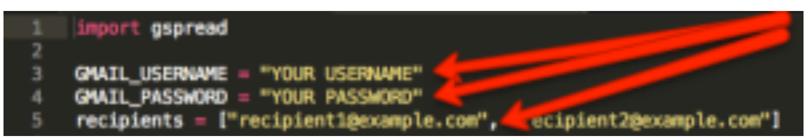
One of those automations explains how you can use Zapier to send a personal welcome email when someone joins your email list. This is great for sending basic emails, but what if you want to do something more stylized?

Unfortunately, Zapier and other services don't let you send programmatic HTML emails through Gmail. So we're going to use a program that does it for us.

Here's what you'll need:

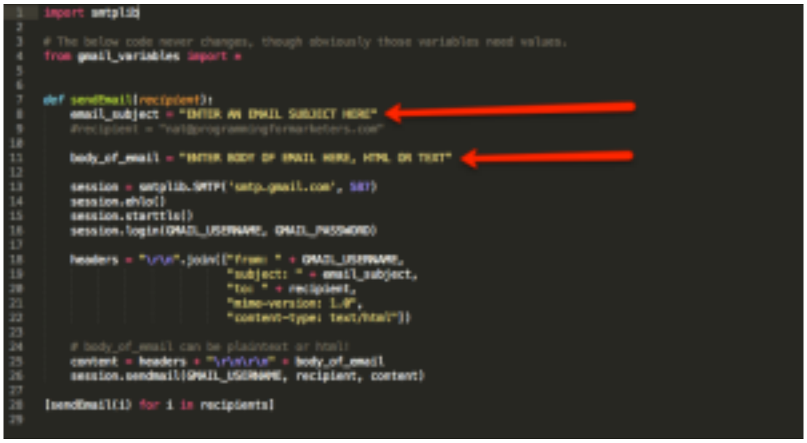
- You need to have Python set up (we explained this in the Setup)
- And you'll need this code. Just hit "Download ZIP" and extract it like we explained in lesson three

Now sending a bunch of emails from your command line is simple. To start, go to "gmail_variables.py" and add your Username and Password where it says to. Then in the "recipients" array (stuff between square brackets) type in your email address twice within the quotes replacing the example emails.



Go to the terminal and run "send_html_email.py" (remember the instructions in the setup), and then check your email. You should have two emails from yourself! Very cool. But the message is a little boring...

Open up "send_html_email.py" and you should see some more all-caps prompts to change the subject and the body of the email. Set the subject to whatever you want, and then the body gets interesting.



You can just send a regular text email if you prefer. Enter text, and use "\n" to indicate a new line (for example: "Hi there!\n\nHow have you been?\n\nBest, \nNat")

But the great thing about this system is that you can send HTML emails as well. So for example, you could design an email in Mailchimp, export the HTML (they give you a link for that once you have it designed), and then send that. Alternatively you could just add some simple html like using The text you want hyperlinked to create a hyperlink.

Not sure how to do something in HTML? [Check W3Schools to start.]

Once you have your email designed, test it with yourself, then try putting in some email addresses for other people.

3. More Advanced - Using Google Sheets to Determine Who to Email

Instead of manually entering the addresses of people you want to email, you can just query a Google Spreadsheet, pull down an entire column, and then email everyone who's in that column. You could even have Zapier send every new subscriber you get in to a Google Spreadsheet to make the email outreach easier.

To do it, you first need to add this code to your computer using the same method we've covered a few times now (download as Zip, extract, go there in terminal and type "python setup.py install").

Once you've done that, you can go in to your "gmail_variables.py" script and uncomment the bottom of the code (delete the triple back-ticks (`).

You also want to add a "#" before the first "recipients" line. This will "comment it out" so that it gets ignored, and the code pulls email addresses from the Google Sheet instead.

Then just replace the all-caps text with the information it calls for, and your program will pull email addresses from your spreadsheet instead of from the "recipients" variable!

Lesson 5: How to Hack Your PR for Massive Exposure

By the end of this lesson, you'll have learned:

- How to quickly and easily find all news relevant to you and your business
- How to get other people to mine data for cheap while you sleep
- How to set up an email robot to send a personalized email to hundreds of people at once, for free
- How to design an HTML email without knowing any HTML

Let's get started!

Starting Off

In this lesson we'll be teaching you a system to hack your PR outreach. It's an extended version of the technique outlined here: http://customerdevlabs.com/2013/09/24/google-news-api-mturk-press/

It will let you find a ton of reporters who know your space, and send them a personalized email suggesting they write about your launch, your company, your new product, whatever you want!

This is imperative if you want to get a large amount of traffic to your site or product on launch and ensure its success. Reporters won't come find you, you have to go to them, and this will make it easy.

Part One: Find Relevant News

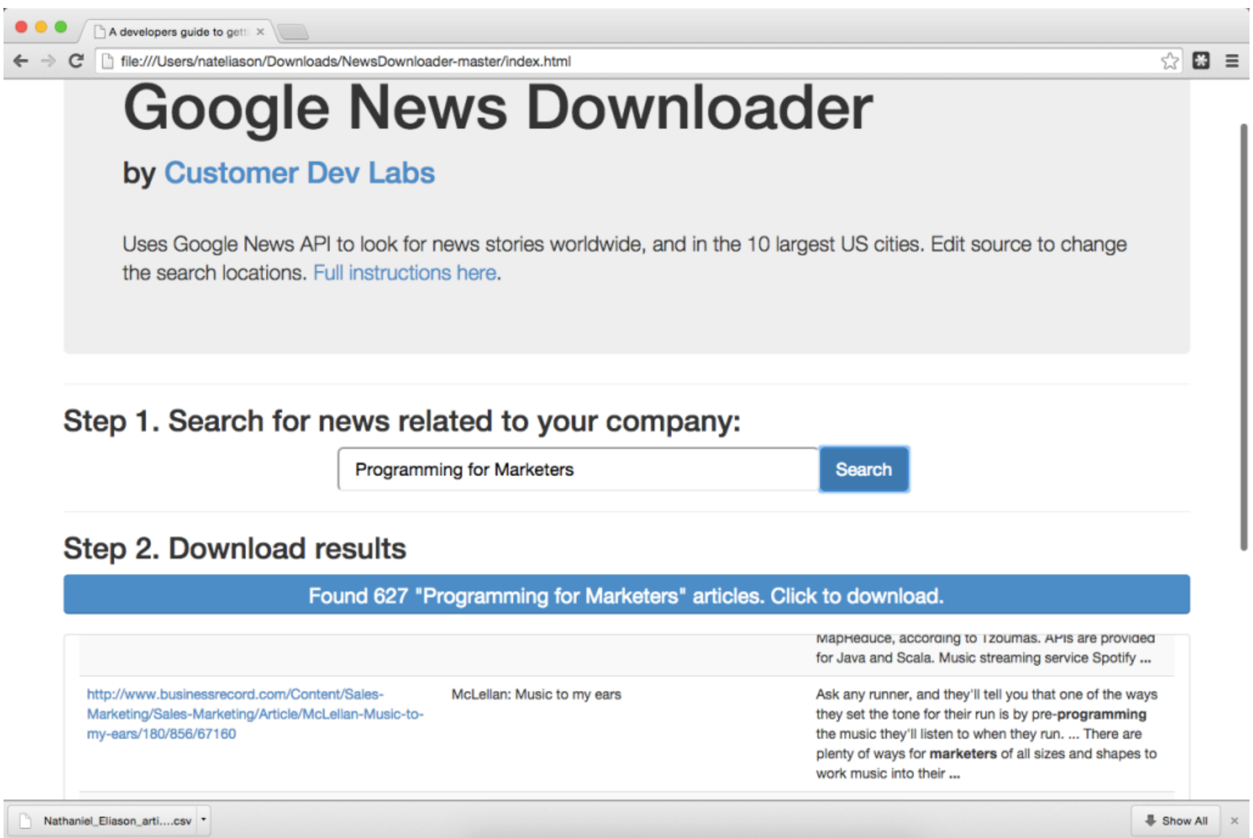
Before you can do anything else, you need to find articles that are similar to what you want reporters to talk about.

The team at Customer Development Labs built some awesome code that makes this part really easy.

You can download it here: https://github.com/JustinWi/NewsDownloader.

Once you have the code downloaded, extract it, go to the folder you extracted it too, open it up, and there will be a file called "index.html." That's the main HTML file. You'll want to open that with your browser of choice (i.e. Chrome) and NOT with Brackets, Sublime, etc.

Now you'll have a new window that looks like this:



Search for something related to your company (for us, it would be "programming for marketers") and the page will return everything it can find in

Google News related to that topic.

Once that's done, go ahead and download the CSV.

Now you have a CSV (spreadsheet) with all of the news relevant to you from recent history. Let's get someone to go through that and find all of the reporters' email addresses.

Part Two: Find the Reporter Email Addresses

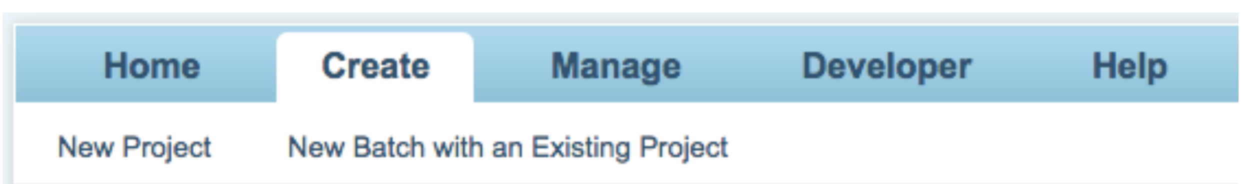
Now you're armed with all the news related to your industry, which means you're also armed with all of the reporters who have written about your industry. These reporters will be the most likely to write about you again

We're going to use a tool called "Mechanical Turk" to find these reporters' email addresses.

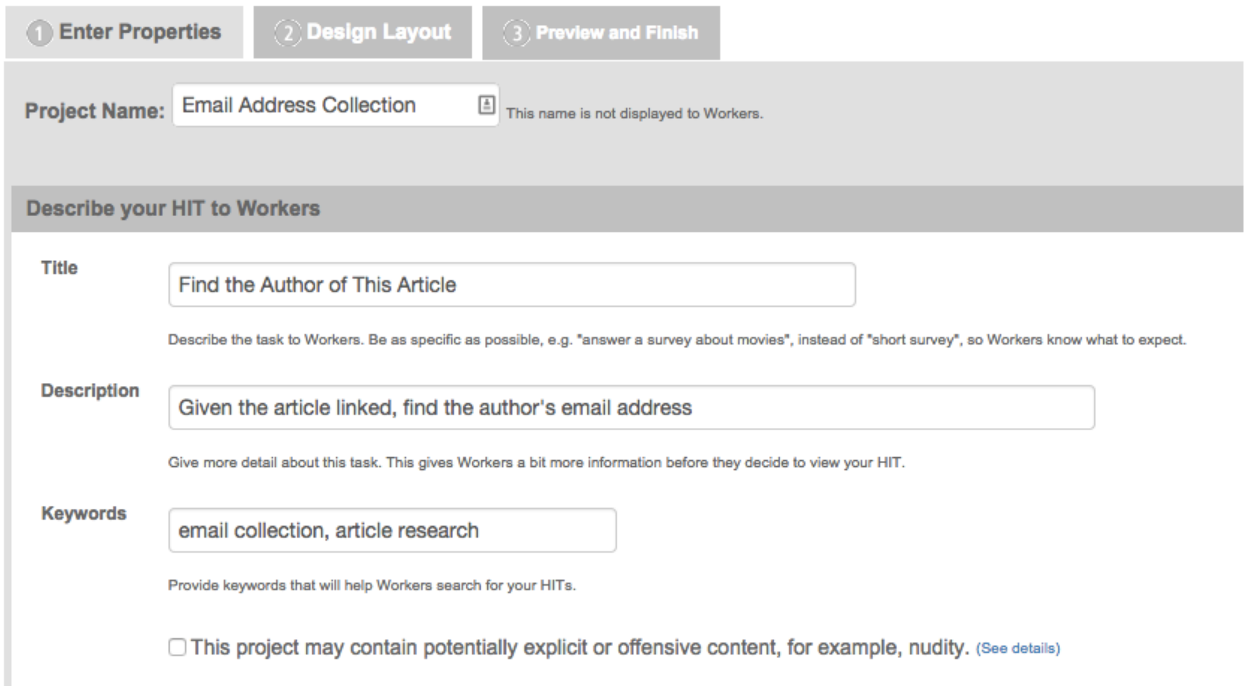
If you're not familiar with it, Mechanical Turk, or MTurk, is a web service where you can pay people small amounts of money to do small tasks, called "Human Intelligence Tasks." The idea being that while we can automate a lot, we can't automate quite everything yet.

First, go to www.mturk.com and sign in with Amazon.

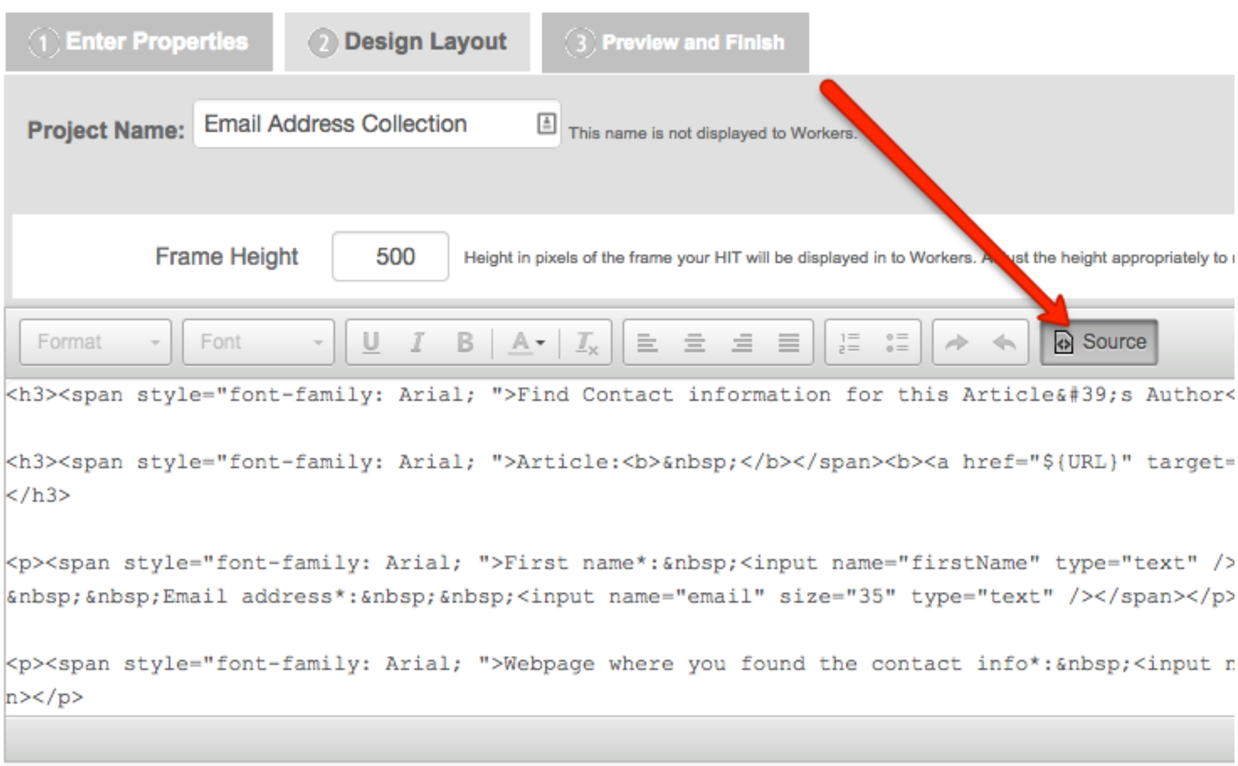
At the top, go to "Create" and then "New Project"



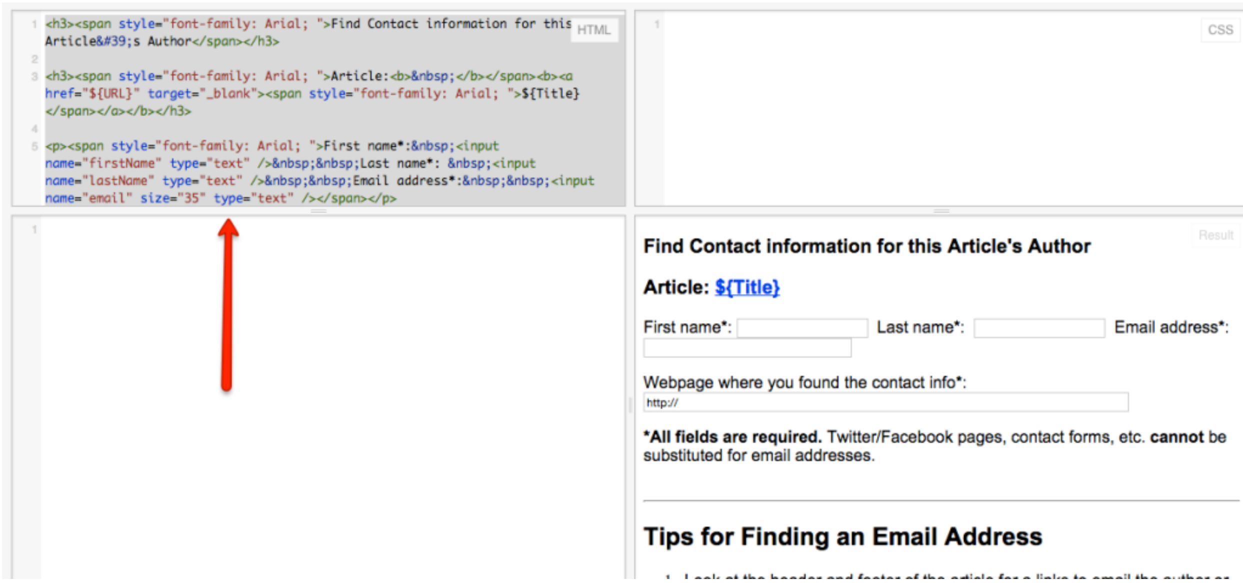
Put anything you want in for the description, this is how the MTurk workers will see it.



Then go to "Design Layout" (step two), click on "source,"



and paste in this code: http://jsfiddle.net/nateliason/455mcqzk/

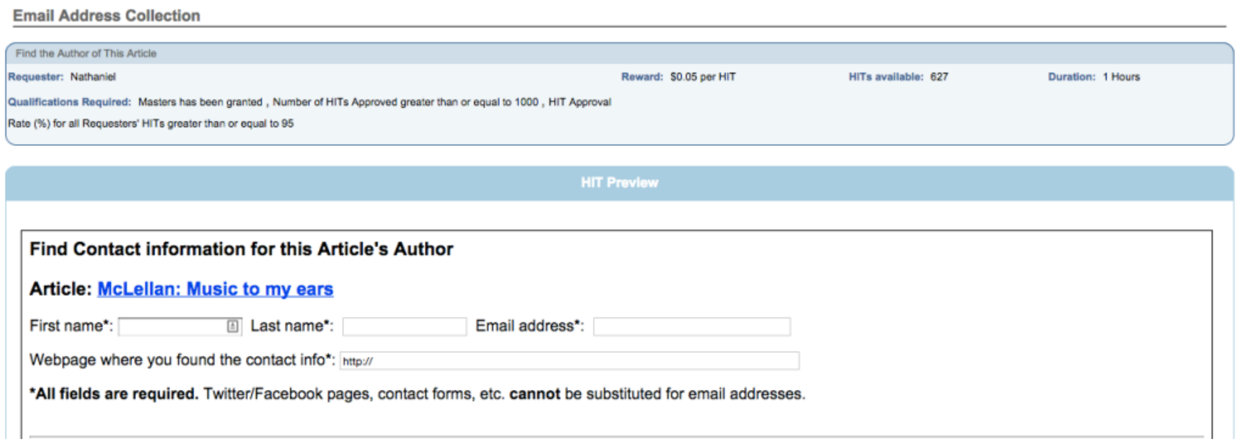


It's a slightly modified version of the one that Customer Development Labs used since their "Step 4" doesn't work anymore.

Then click preview to make sure it looks nice, and hit Finish.

Once you do, select "publish batch" on the next page and upload your CSV when the window pops up.

If everything went smoothly, you should have a window like this where the number of HITs available matches the number of rows in your spreadsheet or the number of articles you found.



Click “Next” assuming the Preview looks good.

Now if you haven’t used MTurk before you’ll need to add funds to your account, they give you a scary red bar telling you where to go to do it though so it’s pretty easy.

Then hit publish! Your HITs are going live, and you can sit back while the email addresses pour in.

If your results start slowing down, you can cancel the unfinished ones and republish them to “refresh” your position in the MTurk list of HITs.

Once the job is done you'll be able to download a CSV with all of the reporter information added to it.

Part Three: Set Up Your PR Email Robot

While the Mechanical Turk results roll in, you want to get ready for reaching out to these reporters.

You have a few options here:

1. Add all the reporters to a Mailchimp list and email them that way. This will get the job done, but it’s clearly automated and less personal. Also

it might end up in their spam.

2. Use a Spreadsheet + Zapier to email everyone. This is better since it's more personal, but it's not perfect since you can't use hyperlinks or images to get their attention.

3. Send all the emails from the command line! This will give you the most customizability, and is the most in the spirit of the course.

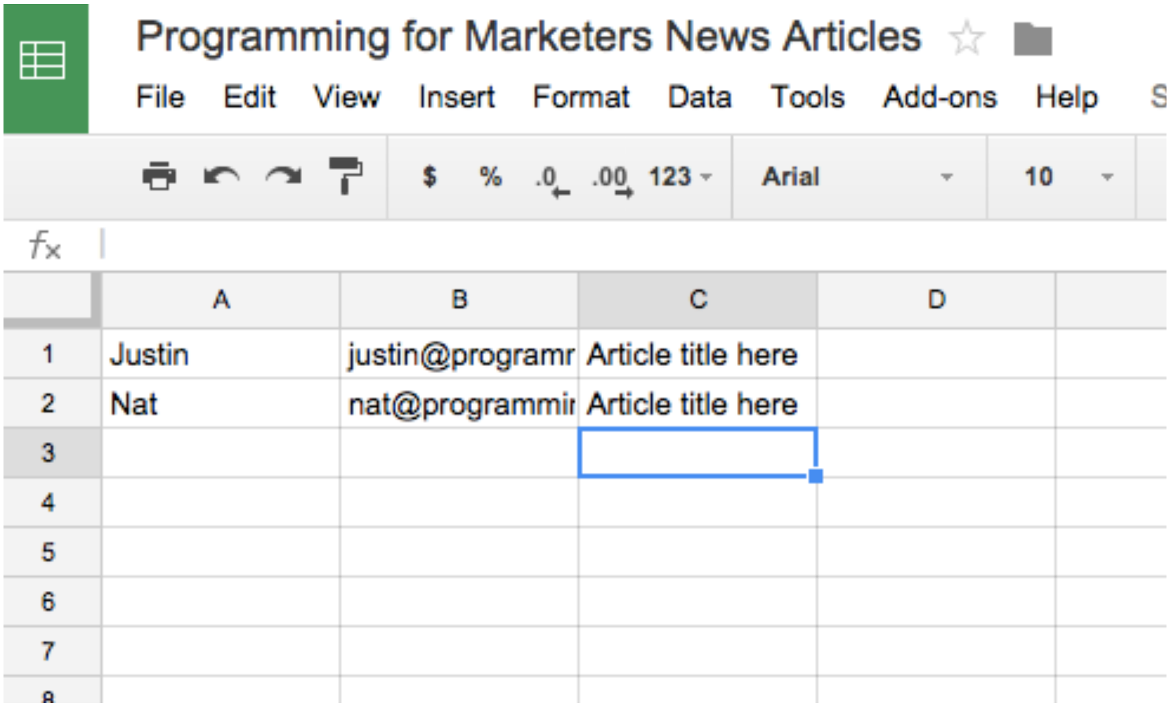
We took the code from lesson 3 and expanded on it for you. You can get it here: https://github.com/ProgrammingforMarketers/pr-email-robot

What this is doing is it's taking a google spreadsheet, pulling down the important data, then using the names and email addresses to send personalized emails to everyone on your list

Here's how you get it working for you:

1. First, in "gmail_variables.py" add in your Google Username and Password (inside the quotes)

2. Second, go to Google Drive, and create a new spreadsheet that you'll use for all of the reporter information. It's absolutely necessary that column A be for their Name, column B be for their email, and column C be for the title of the relevant article. If you lay it out differently the code won't work (unless you edit the code)



3. Third, in "pr_email_robot.py", on line 7, add in the name of your spreadsheet you just created

4. Fourth, you'll need to create a body for the email. We'll get to that in a minute.

Part Four: Design Your Emails

Now, the great thing about using this email robot is that it will let you put in a custom HTML email which means you can have hyperlinks, images, the whole shebang.

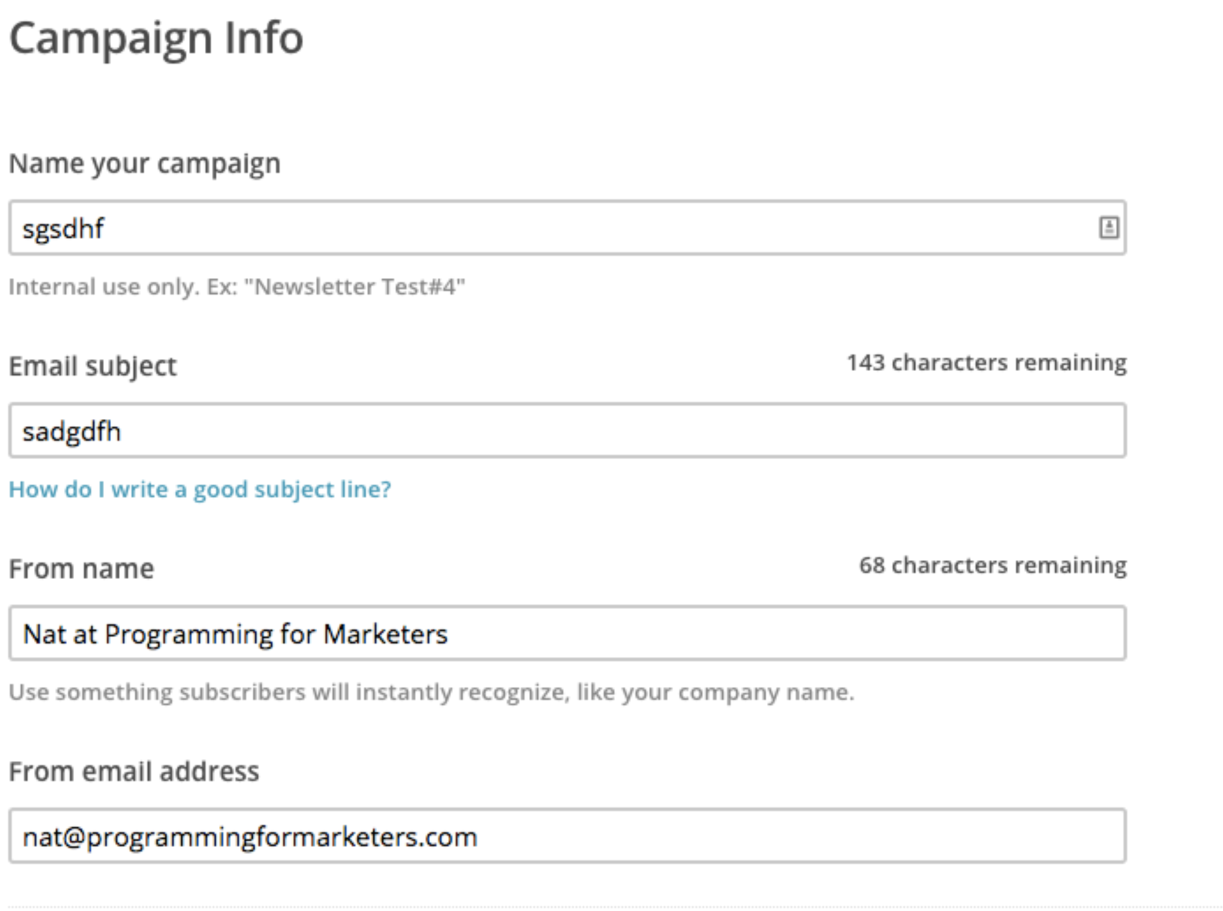
"But I don't know HTML!" Don't worry, you don't need to. Here's what we'll do instead.

First, go to Mailchimp if you don't have an account go ahead and make one.

Second, click "create campaign" (you might need to make a list first, but that only takes a few seconds).

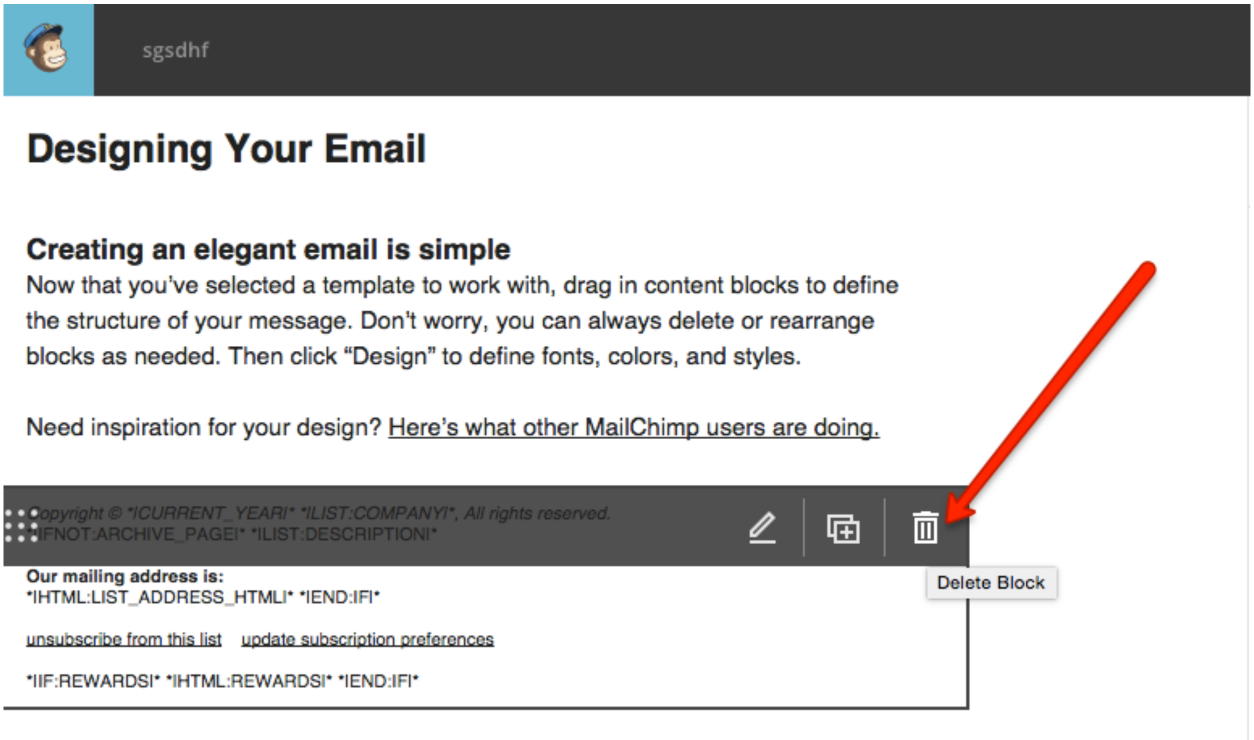
Third, just put gibberish in the "Setup" stage (name of campaign, subject, etc.)

since we're not actually going to send this campaign.

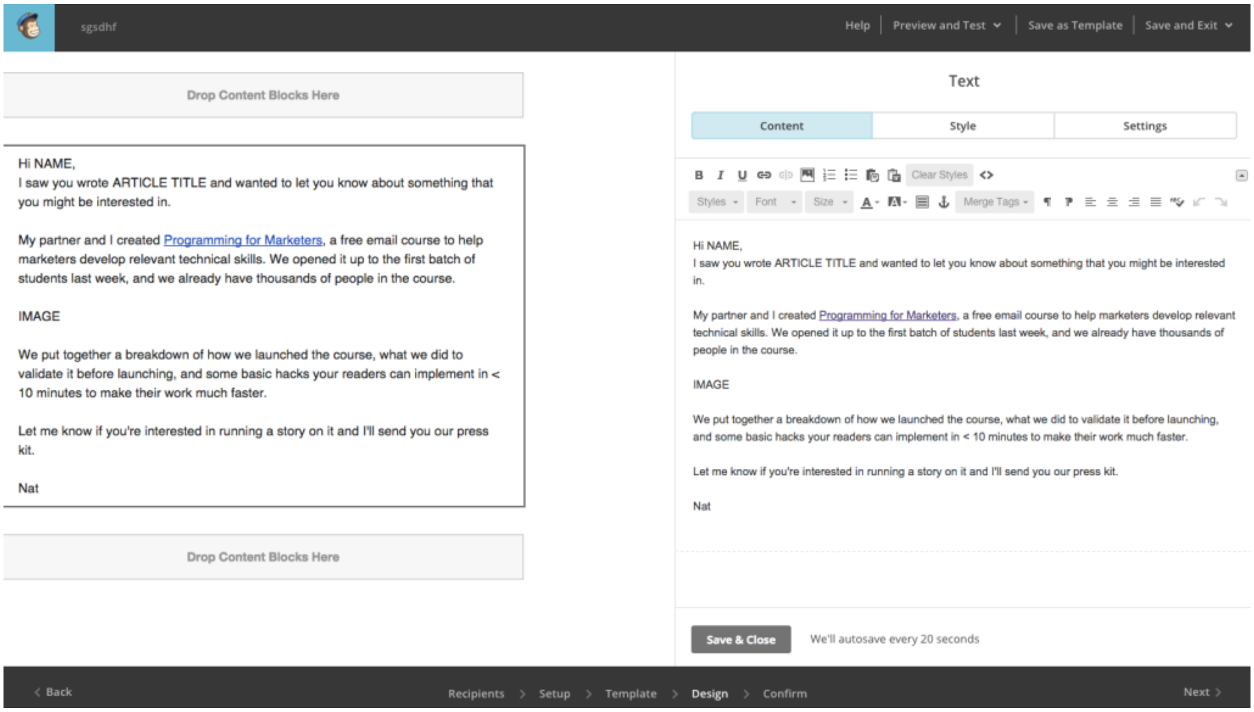


Fourth, for the Template, scroll to the bottom and select "Simple Text."

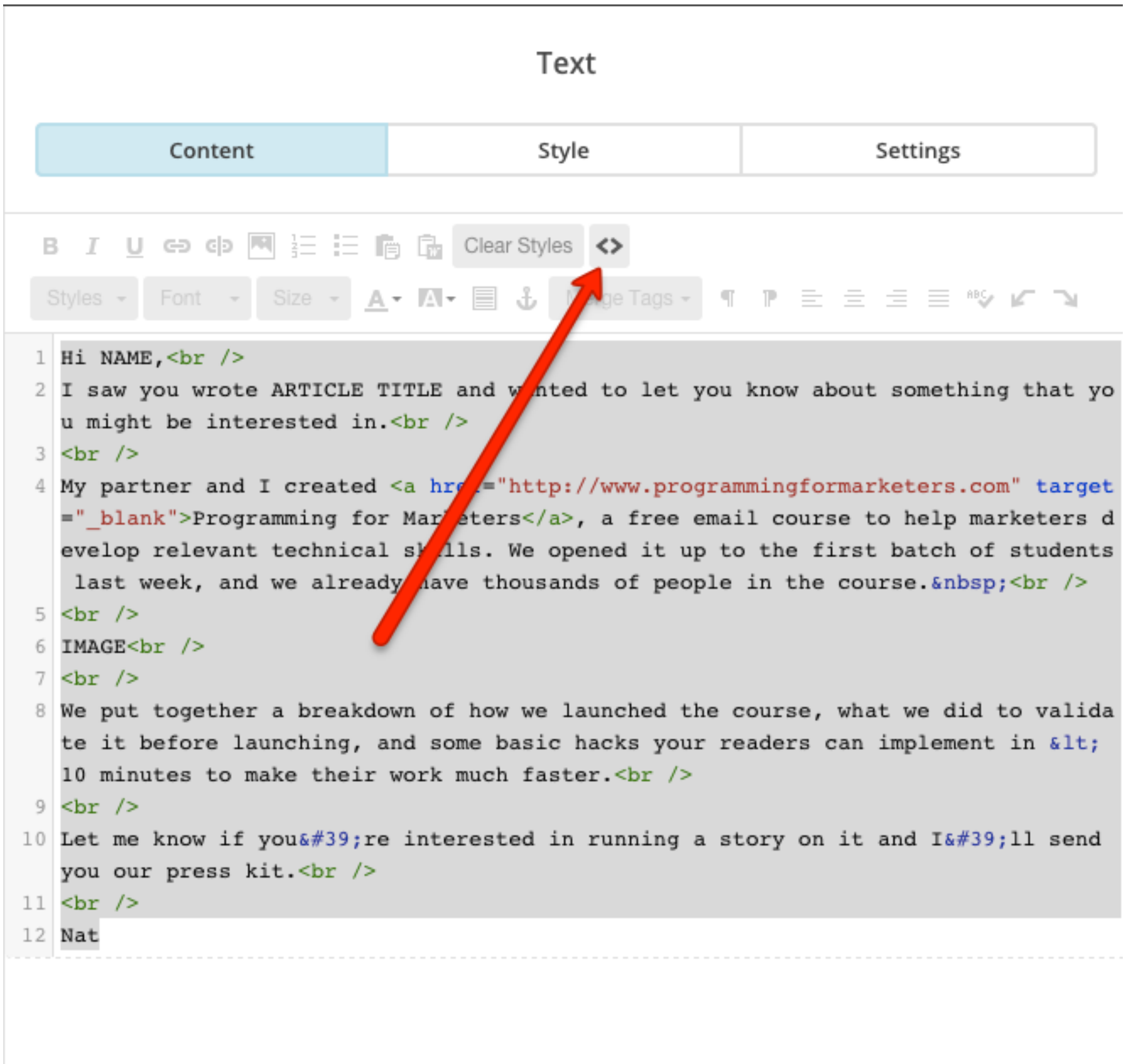
Fifth, in the designer, delete the top and bottom block, you won't need them.



Sixth, fill out the middle block with the email you'd want to send. To make things easier later, write "NAME" where you want their name to go and "ARTICLE TITLE" where you want the title of the article. You can also use "IMAGE" if you want to add in an image.



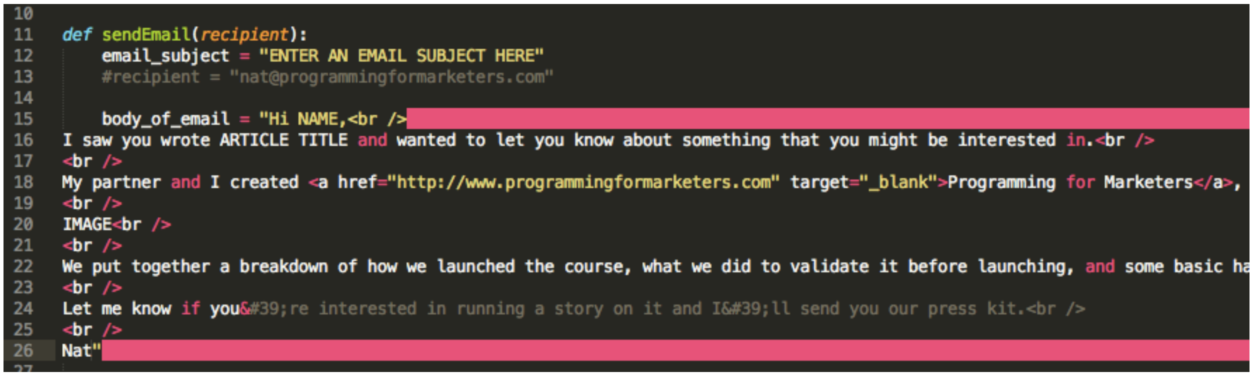
Seventh, click on the "<>" to access the source code of the email. Copy all of it down. We're going to plug it into your email robot.



Part Five: Add The HTML to Your Email Robot

Now that you have your code from Mailchimp, go back to “pr_email_robot.py” and paste in that source code between the quotes after “body_of_email = “ on line 15

It’s going to look REALLY MESSY don’t worry, we’ll fix it.



```
def sendEmail(recipient):
    email_subject = "ENTER AN EMAIL SUBJECT HERE"
    #recipient = "nat@programmingformarketers.com"

    body_of_email = "Hi NAME,<br />
I saw you wrote ARTICLE TITLE and wanted to let you know about something that you might be interested in.<br />
<br />
My partner and I created <a href="http://www.programmingformarketers.com" target="_blank">Programming for Marketers</a>,
<br />
IMAGE<br />
<br />
We put together a breakdown of how we launched the course, what we did to validate it before launching, and some basic ha
<br />
Let me know if you&#39;re interested in running a story on it and I&#39;ll send you our press kit.<br />
<br />
Nat"
```

The horror!

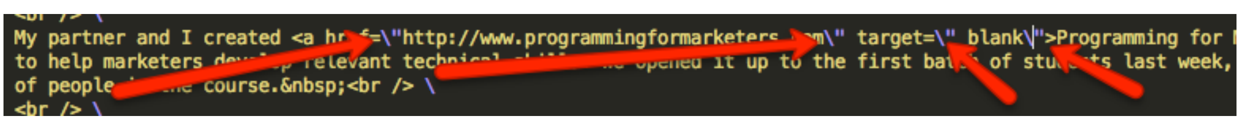
First, at the end of every line, add a "\". What this does is tell the code to ignore the new line and treat the next line as part of the line above.

```
def sendEmail(recipient):
    email_subject = "ENTER AN EMAIL SUBJECT HERE"
    #recipient = "nat@programmingformarketers.com"

    body_of_email = "<body><p>Hi NAME,<br /> \
                    I saw you wrote ARTICLE TITLE and wanted to let you know about something that you might be interested in.<br /> \
                    <br /> \
                    My partner and I created <a href="http://www.programmingformarketers.com" target="_blank">Programming for Marketers</a>, a free email course \
                    to help marketers develop relevant technical skills. We opened it up to the first batch of students last week, and we already have thousands \
                    of people in the course. <br /> \
                    <br /> \
                    IMAGE<br /> \
                    <br /> \
                    We put together a breakdown of how we launched the course, what we did to validate it before launching, and some basic hacks your readers can \
                    implement in &lt; 10 minutes to make their work much faster.<br /> \
                    <br /> \
                    Let me know if you&#39;re interested in running a story on it and I&#39;ll send you our press kit.<br /> \
                    <br /> \
                    Nat</p></body>"
```

Second, select all of the email text except for the first line and hit Tab until it lines up under the first line better like the picture above (this isn't for functionality, just aesthetics).

Third, if you have quotes anywhere (like around a URL) then you need to put a "\" in front of those too. This "escapes" the quotes so that they don't end the quotes defining what the body of the email is.



Fourth, we're going to add in your info from the spreadsheet. Where you have "NAME" you need to replace it with "" + recipient[0] + "" (quotes at both ends)

```
"<body><p>Hi " + recipient[0] + ",<br /> \
I saw you wrote ARTICLE TITLE and wanted to
```

You have to end the quotes so that the code pulls in the first column of your spreadsheet (column 0, since python counts from 0), instead of just writing out “recipient [0]”. The “+”s add the two sets of quotes together so that it makes one cohesive statement.

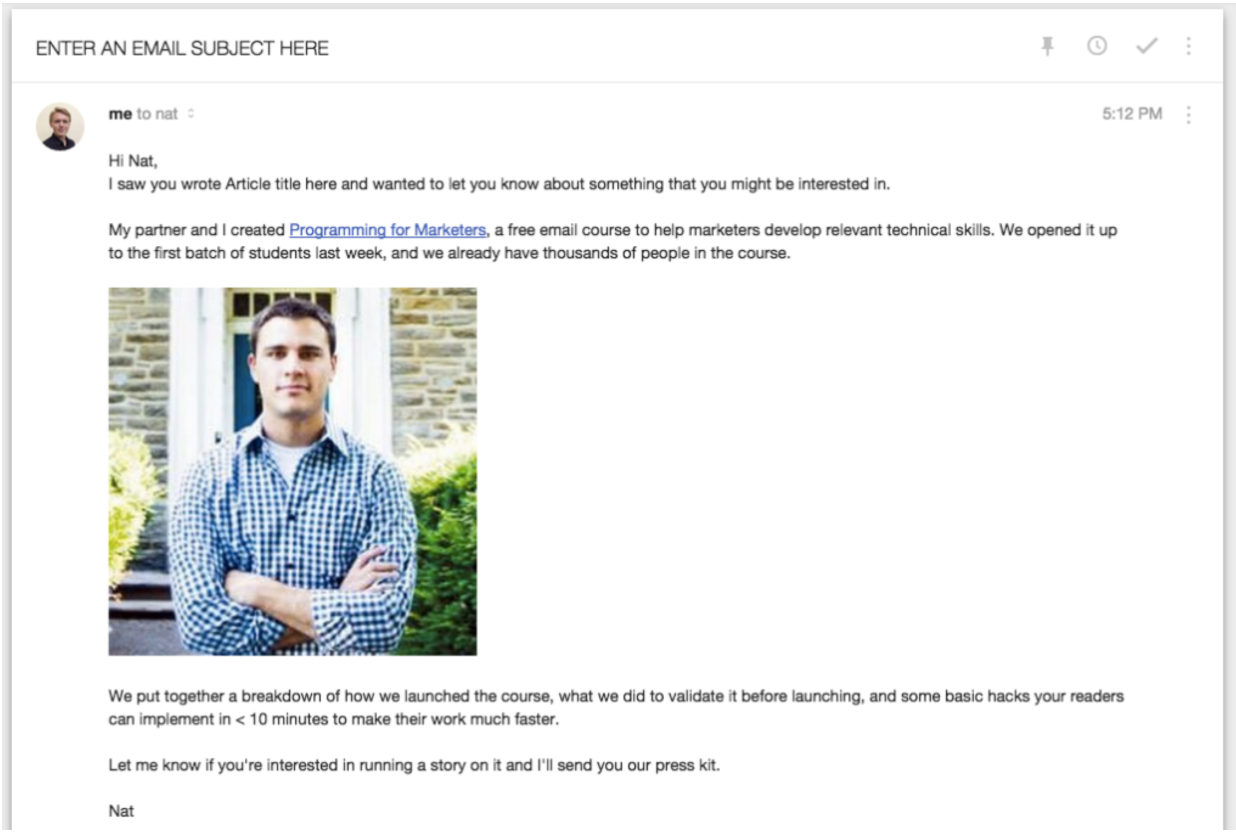
Fifth, you’ll do the same thing for ARTICLE TITLE, but use “recipient[2]” instead of 0, since the title is the third column in the spreadsheet.

Sixth, replace any use of “IMAGE” with this code:
<img src="IMAGE URL" style="width:304px;height:228px"> (you can set the dimensions to whatever you want, also remember to put “\” before the quotes!).

```
<br /> \
<img src=\"http://i0.wp.com/www.programmingformarketers.com/wp-content/uploads/2014/12/d52d0c9f5ea11c51cd650e509217bc9a_400x400.jpeg?w=1080\" \
style=\"width:304px;height:228px\"><br /> \
<br /> \
```

Note: for this to work, you need to put the image somewhere online. It could be your own site, or you could just upload it to IMGUR. To find the URL for a picture, right click on it, then click to “Open Image in New Window.” The URL for that new window is the URL you put in to the code.

Seventh, you're done! You now have an email that you can automatically send to hundreds of people at the same time, while having it look personalized.



Part Six: Finishing Up and Next Steps

Just to be safe, you'll want to send a test email to yourself to make sure it works. Put some sample data in the spreadsheet (your name and email) and then run the program. If everything goes smoothly you should get an email from yourself.

Then all you need to do is copy in all of the data you get back from your Mechanical Turk job, and run the program! You just automated all of your PR outreach.

For going further, you could copy your “pr_email_robot” file into a number of different files for a staggered outreach leading up to the launch or big event. You also might want to go through your list of emails and make sure there are no duplicates.

Lesson 6: How to Use SQL to Find Your Best Users and Customers

In this lesson we're going to get you up and running with SQL, the most popular database-manipulation language in the world.

What does that mean? Well, your site stores all of its information in tables, which are essentially giant excel spreadsheets. The collection of all of those tables (plus a few other things) makes up the database that your site runs on.

Within these tables is all of the information you need to run highly targeted marketing campaigns, engage your best users, target people for up-sells, and more. While you can read these tables like an Excel spreadsheet, they might be hundreds, thousands, or millions of rows deep. SQL lets you quickly manipulate them to spit out whatever data you need.

How Does that Help Me as a Marketer?

Your best customers are the people more likely to buy more from you, and your most active users are the most likely to give you meaningful and useful feedback.

SQL will let you go through all of your sales or user history to see who the most active people are, who the biggest purchasers are, and who you'll have the best luck with reaching out to.

It can also give you ideas for who to send free swag to, who to ask for referrals from, and more. With SQL you'll have total insight into all of your company's metrics giving you the power to use them to drive your marketing. The sky is the limit really.

Getting Started with SQL

We're going to focus on breaking your users down by a certain criteria (such as when they signed up or how many things they've bought from you).

This will give you a sense of who your best customers are, who you might want to reach out to, who you could engage with, and it will give you basic SQL tools to go off and look into your own database and create your own queries.

For this lesson, we're going to teach you some SQL in your browser so that you don't need to immediately get set up with your company's database.

Go ahead and open this page: http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all It's an example database that you can run SQL queries against to see how they work and to get some practice.

At the end of this lesson we'll hook you up with an explanation on how to join your company database.

Throughout this lesson we're going to progressively build up your SQL knowledge. But let's start with something fun.

Type this into the SQL editor where it says "SELECT * from Customers";

```
Select orders.customerid, customers.customername,
count(orders.customerid)
from orders join customers on customers.customerid = orders.customerid
group by (orders.customerid)
order by count(orders.customerid)
```

What it spit out is the list of all customers and how many purchases they each made, ordered from fewest to most. So if you wanted to reach out to your best customers, Ernst Handel seems like a good candidate!

Now we'll break it down and explain how it all works.

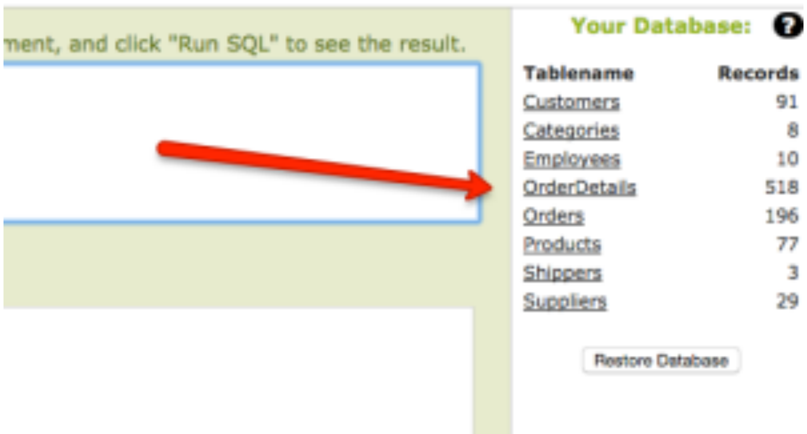
Selecting Data

The most basic SQL command is "SELECT" which lets you tell the database what information you want it to show you.

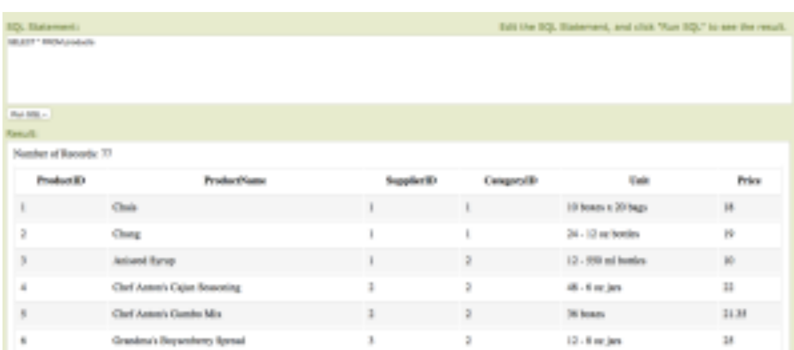
Go back to the prompt, and type in: "**SELECT * from Customers**"

What that did is SELECT(ed) Everything (the *) from the table called "Customers"

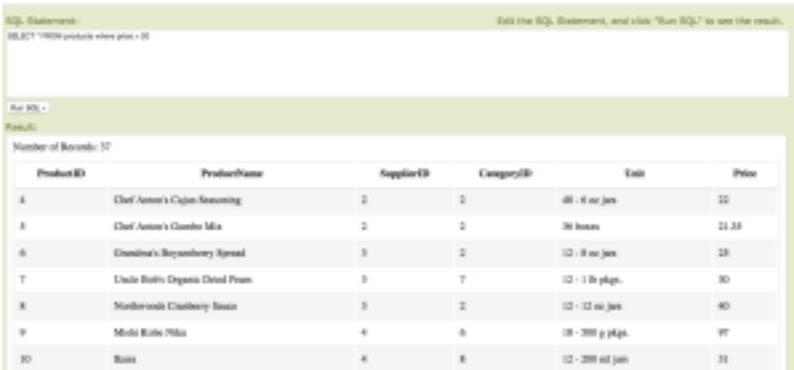
Quick note, a database is a collection of tables, and a table is like an excel spreadsheet. So in this example database there are 8 tables that you can see on the right column.



Now try "**SELECT * from products**" and you can see all of the products in the database.



Let's step it up a notch: try "**SELECT * from products where Price > 20**" And you'll have all of the Products that cost more than 20 dollars. You can filter by any column on the table, using >, <, or =.



Here's another example: "**select * from Customers where Country = "Germany"**" will give you all of your customers from Germany

And you can select specific columns, such as by doing "**select CustomerName from customers**" which will just give you a list of all of the names in the Customers table.

Joining Tables for More Data

But pure select statements are limited. You can open them up a lot more by joining two tables together, to get their combined information in one chart.

Join statements have 3 parts:

1. The data you want to Select, and thus print out

2. The two tables you want to combine

3. The column you want to use to match them up

For our analysis, we need to know how many items each customer is buying. To do that, we want to know the name of the customer, and how many things they bought. But the name isn't enough, we also want their "CustomerID" so we can look them up later in case two customers have the same name.

So where is that data? Well the Customer ID is the "CustomerID" column in the "Customers" and the "Orders" table. The Customer's name is the "CustomerName" column in the Customers table. But there's no column for how many things they ordered. To do this, we'll have to go through everything in the orders table and add it up.

Orders has the same column as Customers called "CustomerID" which tells you which customer placed an order. So all we need to do in order to figure out how many things someone ordered is take their CustomerID, and then count how many times it shows up in the Orders table.

This is easily done with "**count(orders.customerid)**". That counts how many times a given CustomerId shows up in the Orders table. But it will only work as part of the larger function we're building, it doesn't do anything on its own.

So the data we want to select in order to see who our best customers is would be **orders.customerid, customers.customername, and count(orders.customerid)**. If you remember, that was the first line from our example query at the beginning.

But before we can pull down data, we need to join the Customers and Orders tables. We're going to use the "Join" statement. The Join statement lets you temporarily combine two tables, so long as they share a common column.

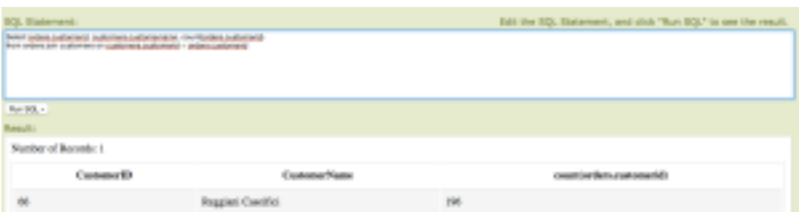
In this case, the "orders" table and the "customers" table both have a "CustomerID" column, so we can use that to match them up.

First, you take the orders table by typing "from orders" then you bring in the join statement, "from orders join customers" and then you tell it what column to use in joining them “**from orders join customers on customers.customerid = orders.customerid**”

Plug in that line in addition to the SELECT statement we made earlier, and you'll get:

```
Select orders.customerid, customers.customername,
count(orders.customerid)
from orders join customers on customers.customerid = orders.customerid
```

When you run it, you should get one line of data. Not exactly what we wanted.



Why? Well the “**count(orders.customerid)**” is simply giving you the total count of orders in the entire database, which isn’t what we want. We want how many orders each customerid has, which we can solve with a group function

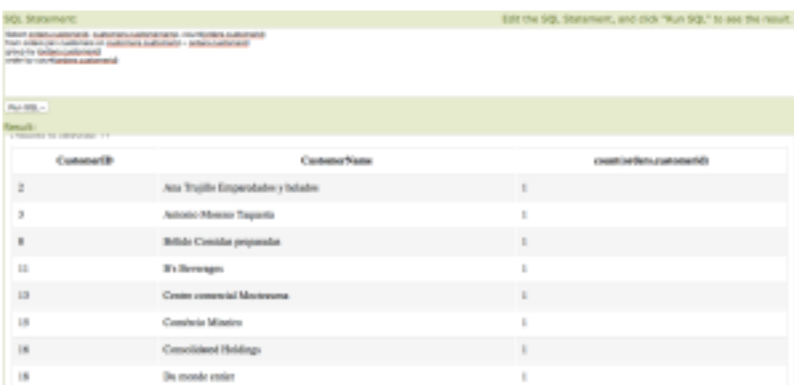
Add a new line saying “**group by (orders.customerid)**” so you get:

```
Select orders.customerid, customers.customername,
count(orders.customerid)
from orders join customers on customers.customerid = orders.customerid
group by (orders.customerid)
```

Now instead of giving you the total count, it will give you the count for each customerid and match it up with their corresponding info.

Last but not least, you want it to look a little bit nicer. Instead of a random assortment of numbers, you can add in “**order by count(orders.customerid)**” which will sort it by how many orders each user has.

Select orders.customerid, customers.customername, count(orders.customerid)
from orders join customers on customers.customerid = orders.customerid
group by (orders.customerid)
order by count(orders.customerid)



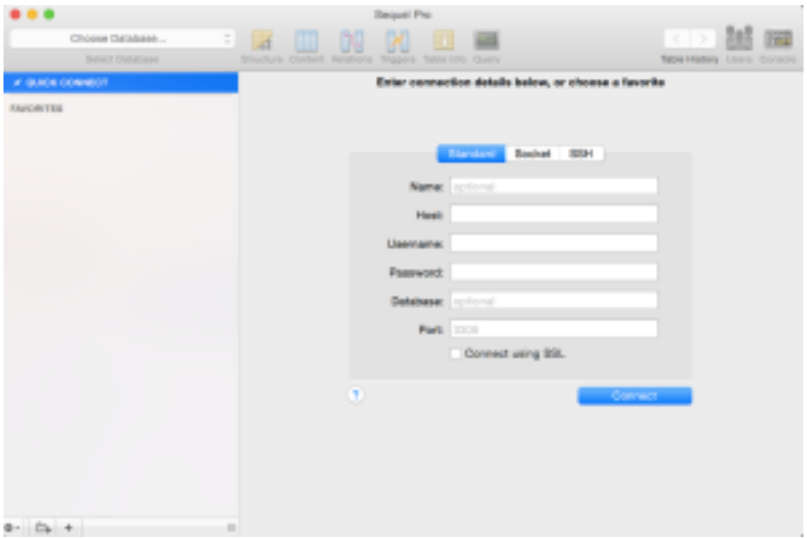
Using it On Your Own

These Select, Join, and Group statements enable you to query your own company database and find out everything you could need to know about your customers. But to do that, you need to get set up with your company database.

You'll need some help from one of the developers you work with since they have to give you access, but once they do you'll be able to do everything on your own.

First, download a SQL editor like Sequel Pro.

Once it's installed, open it up, and you should see a screen like this.



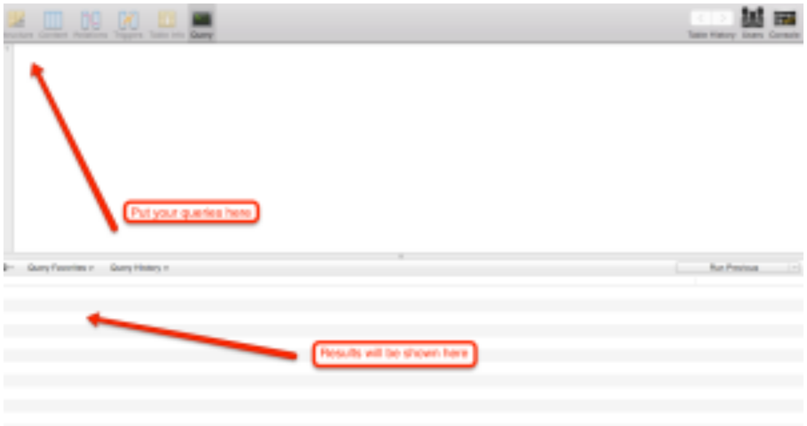
This is where you enter your database credentials. You'll need to ask someone for access to a "read-only" version of the database, and you'll need the:

- Host
- Username
- Password
- Database (possibly), and
- Port

You can name it whatever you want.

You might have to be on your company's VPN to access the database. Be sure to ask about that!

Hit connect, and you should be in! You can interact with your database just like you could with W3Schools, all you need to do is start typing in queries.



Lesson 7: How to Scrape Any Blog to Optimize Your Next Post

By the end of this lesson you'll be able to go to any blog, scrape the URLs for all of its blog posts, add in the social sharing data for those posts for free and then export that information to a PDF so you can run your own data analyses.

We originally got this idea from [Nate Desmond]() who wrote about it over on [OkDork](). Unfortunately, most of the beginning of that post is out of date since the tools have changed.

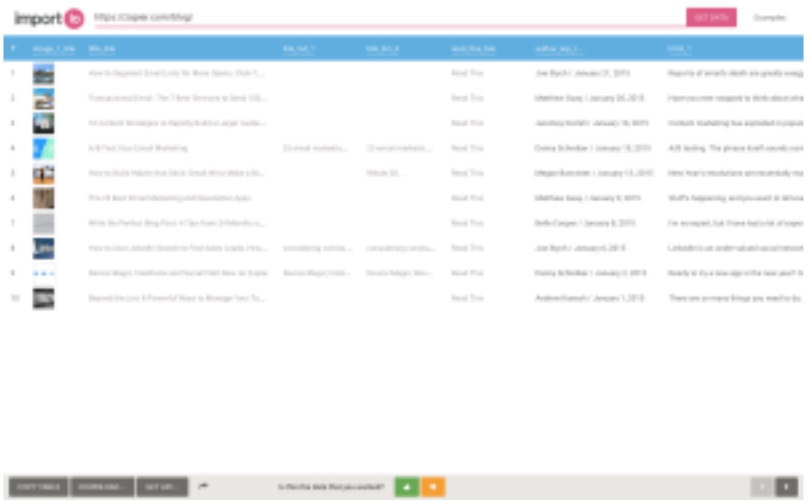
Using this method you'll be able to get a sense of what's working and what isn't on your blog without having to hire a data scientist. Let's get started.

How to Get All of the URLs on a Site

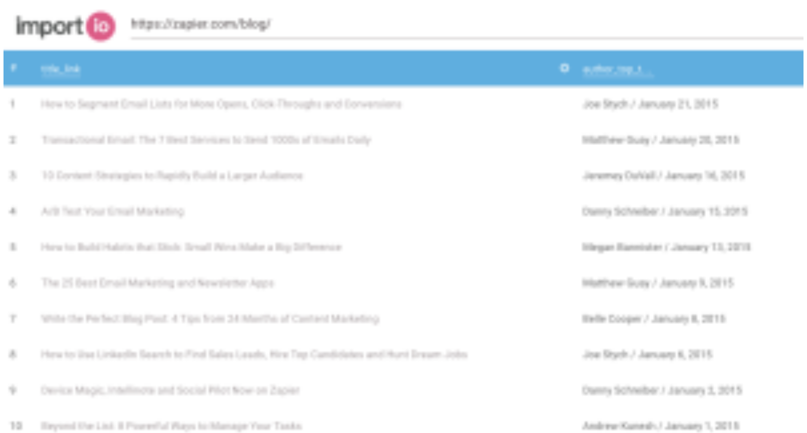
Head to [Import.io](), and type in the URL for the blog you want to get data from. Since P4M doesn't have a blog, we'll use [Zapier's]() as an example.

The URL we'll use then is [www.zapier.com/blog]()

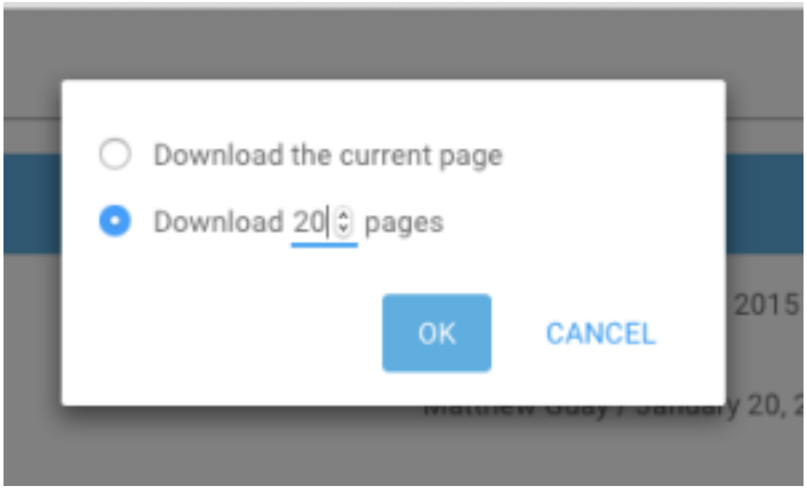
When you plug it in to Import.io, you'll be presented with a page like this that looks kind of like a giant table (yours will look different if you're doing it on your own site).



Next, you want to remove all of the data you don't need. For figuring out the top performing posts, let's just keep the number of the post, the title of the post (which will have the URL), and the author.



Now we can download all the data. Click the download button in the bottom left and then type in how many pages you want. You'll usually be limited to ~20, but in this case that's the last 200 posts which is more than enough information.



Once it's downloaded, open it up, and you have a spreadsheet with the last 200 blog posts, their title, URL, author, and post date!

You might want to clean it up a bit. For example, I had a black column C that I deleted, and then I used "Text to Columns" to separate the Author/Date column into two separate ones.

Intermediate Step: Creating a SharedCount Account

Before we move on to the next section, you need to make an account with SharedCount.

SharedCount is a site that will let you send it a limited number of queries each day to see how many social shares a certain URL received. You used to not have to make an account, but as of January 1st they changed their system and now you do. But don't worry, it's still free, and the free plan lets you do 10,000 queries a day.

Simply go here: https://admin.sharedcount.com/admin/signup.php and create an account.

Once you confirm your email address, you'll be able to see an "API Key." Go here: https://admin.sharedcount.com/admin/user/home.php to find it, and be sure to save it somewhere for the next step.

Getting Social Share Data for Each Blog Post

Now you need to know how well these URLs performed.

The easiest way to do this is to install "OpenRefine" for Mac or PC.

If you have problems installing it on Mac (like I did) then you can go here to fix it.

Once it's installed, open the program, and upload your spreadsheet. It should look like this.

WARNING: The original spreadsheet will have been downloaded as a CSV, but you should save it as an XLS or XLSX in the "Save as..." options to make this less of a hassle.

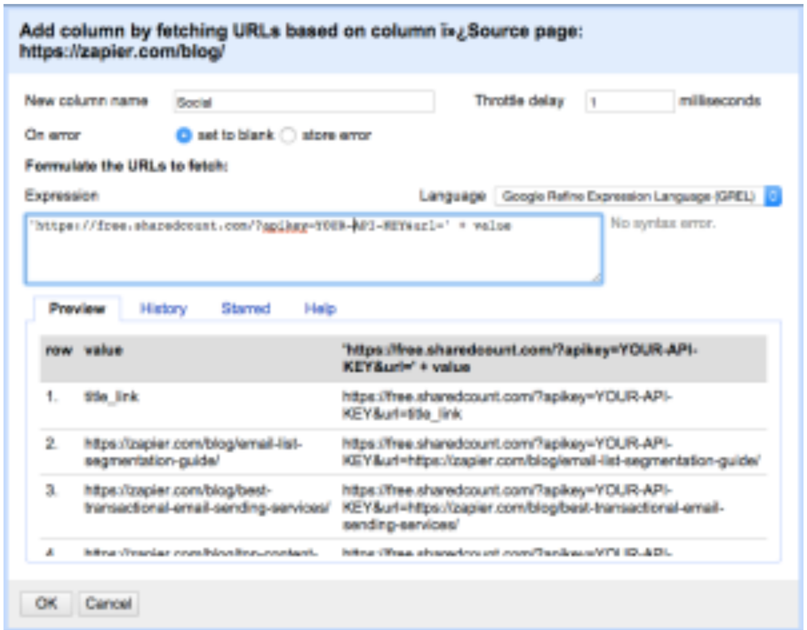
If everything looks fine, hit "Create Project" in the upper right

Now you should see all of your columns again in a page that looks like this:

Click on the arrow above the URL column, then in the drop down menu select "Create New Column by Fetching URL.

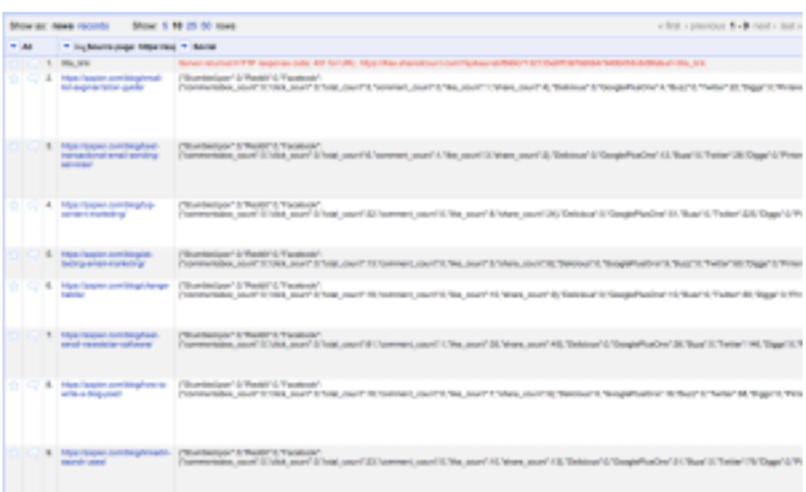
When that box comes up, you're going to:

1. Name the column whatever you want, I went with "social"
2. Change the throttle delay to 1ms
3. And then you're going to put this code in the box (with your API key inserted in where the all-caps text is):
   'https://free.sharedcount.com/?apikey=YOUR-API-KEY&url=' + value



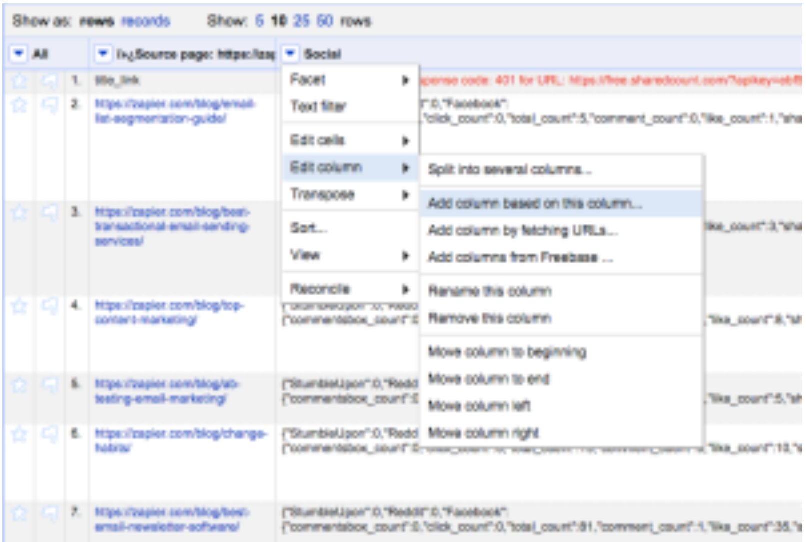
What that does is queries the site you created an account with, Sharedcount.com, to see how much social activity each post received. We'll then break down that information by social source in the next section

If everything went according to plan, you should now have a column that looks like this:



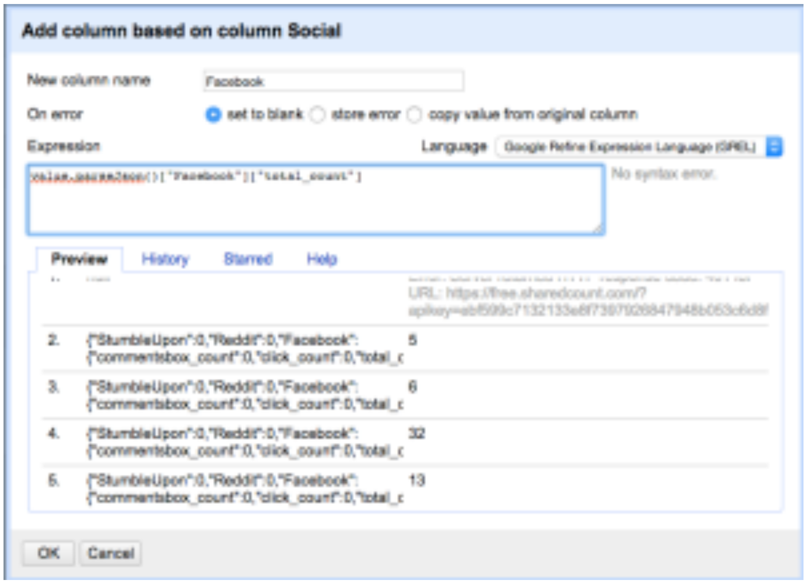
Now we can use all of that data to get the individual social data.

Click the arrow next to “social” and select “add column based on this column.”



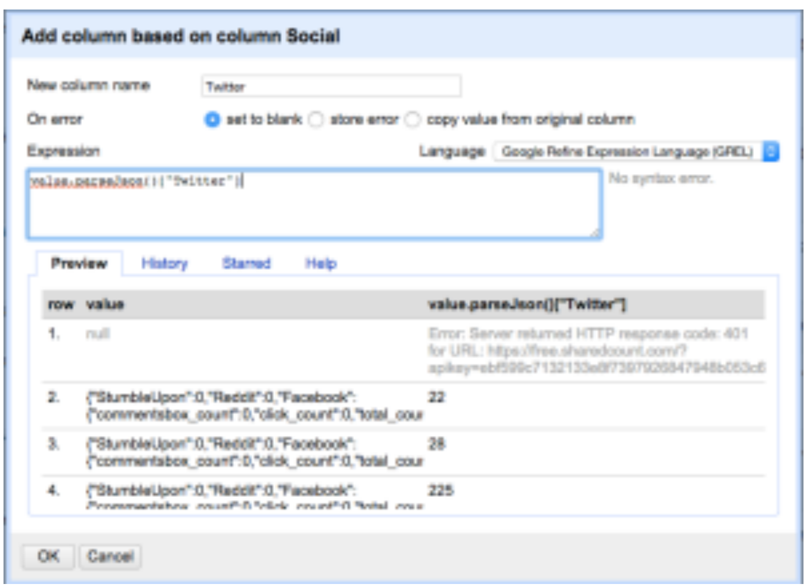
Add in this data to create a column for Facebook actions (shares, likes, comments):

- Name it “Facebook”
- Paste this in the value box:
  value.parseJson()["Facebook"]["total_count"]



And for Twitter (and everything else) this is all you need:

value.parseJson()["Twitter"]



You can also add StumbleUpon, Reddit, Pinterest, LinkedIn, Digg, Delicious, and Google+

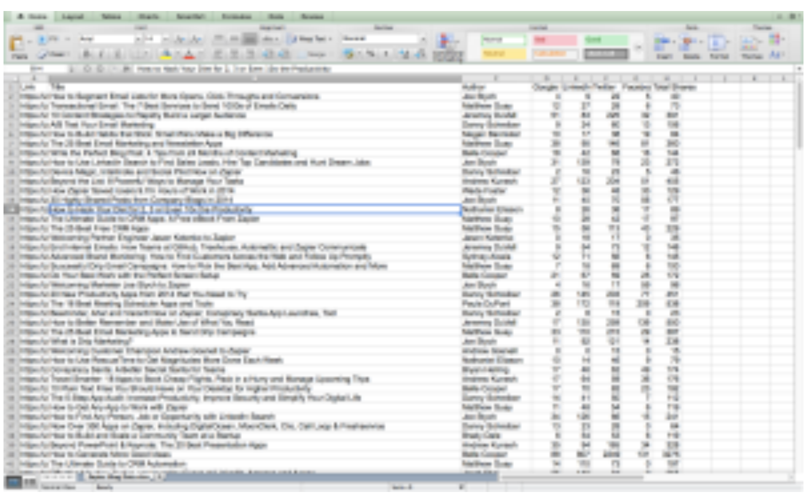
Once you have all of the columns you want, you're ready to export it back to your desktop.

Just hit Export, then Excel in the top right, and open it on your desktop.

Setting Up Your Spreadsheet

Before we get into the comparisons, you might want to clean up your data.

1. You can delete the social column, you don't need that anymore
2. You can move the social counts to the end
3. You can delete the index page number if you haven't already
4. And you can add a new column for adding up the total number of shares.
5. I also went ahead and deleted the date column, something was odd with the data and I'm not as concerned about it
6. And now you should have a nice looking spreadsheet with social data that looks like this!

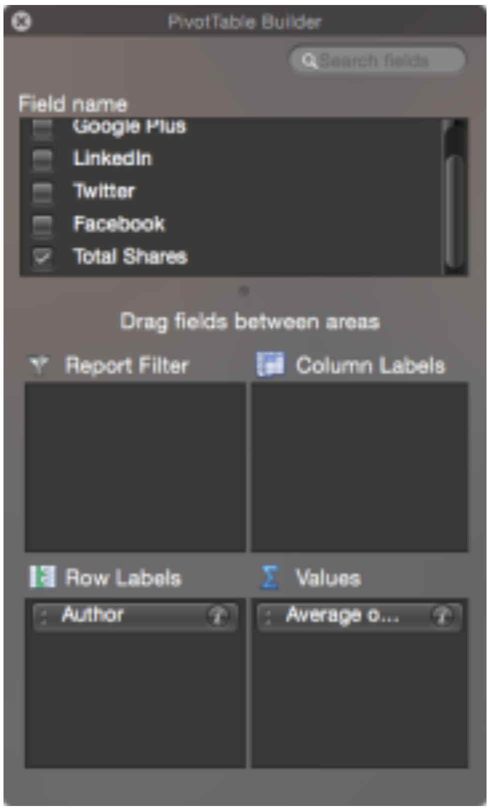


Now we can get into the data. Here are some ideas for playing around with it to help you get started.

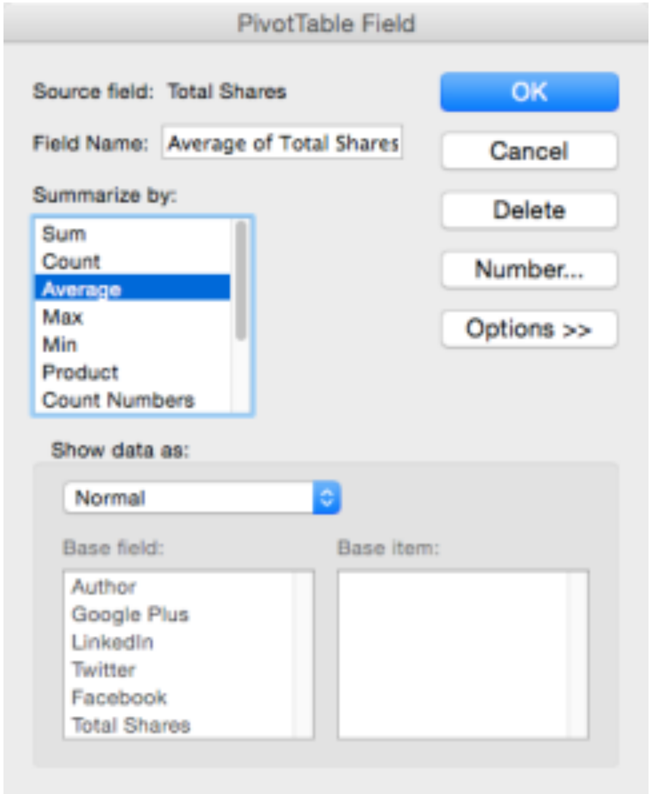
Which Author Has the Best Results?

Open up a new sheet, and create a Pivot Table selecting all of the information from "Author" through "Total Shares." (If you're unfamiliar with Pivot Tables, check out this tutorial.)

Then set the "Author" as the row label, and "Total Shares" as the value.



You'll have to go into the "total shares" selection though and change it to "Average" instead of "count", since count just tells you how many posts they have.



What's the result? It looks like Belle Cooper and Danny Schreiber are leading the pack by a long shot. Brian Smith has the highest average, but he only has one post, so that's not a good indicator of success.

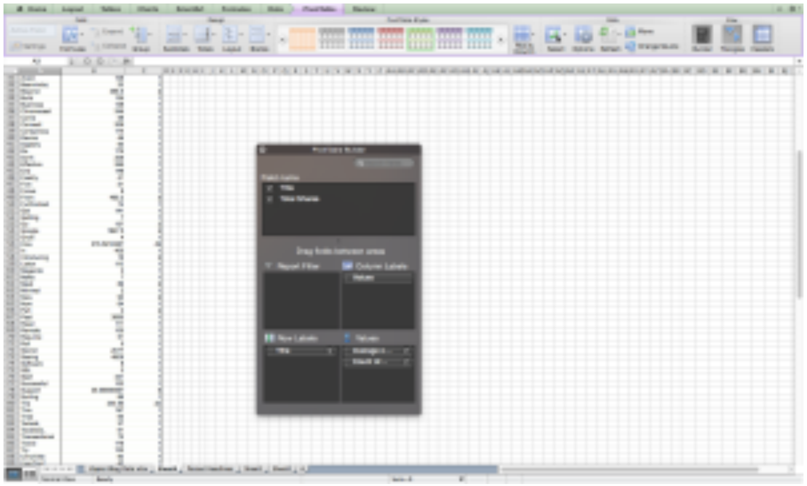
What Word is the Best to Start with for your Headlines?

To do this, we first take the "title" column and paste it into a new sheet.

Then, using text-to-column, split the column based on each space into separate columns.

Now delete every column after the first one, and paste in the share counts alongside the first words of the titles

Then create a pivot table of those two. You'll be able to see the average share count of the Title first words, as well as how many times that word was used first.



In this case the data are not particularly useful since Zapier uses “How” as its opening line with a high frequency, and the only other word that is even close to statistical relevancy is “The.”

However, we can do something further. If we take an average of the averages for all the posts starting with a number, we see that the average there is 496, which is higher than the average for “How” by over 100 shares per post, and higher than the blog-wide average of 329.

Lesson 8: Leveraging APIs to Learn More about Your Customers and Leads

In this lesson we're going to get you set up with a comprehensive spreadsheet where you can track your customers, prospects, and anyone else.

It'll help you decide who to reach out to, how to target them better, and help you make the most of social media.

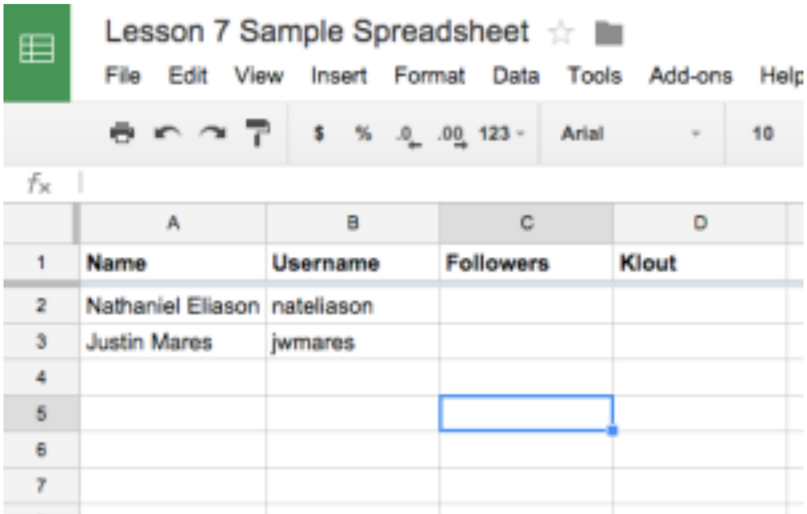
Setting Up Your Spreadsheet

There are a few ways to do this, you can put all of your top prospects and clients in one sheet, or you can have two separate sheets, one for Twitter and one for Email. We're going to go with the latter and use it as the example in the lessons.

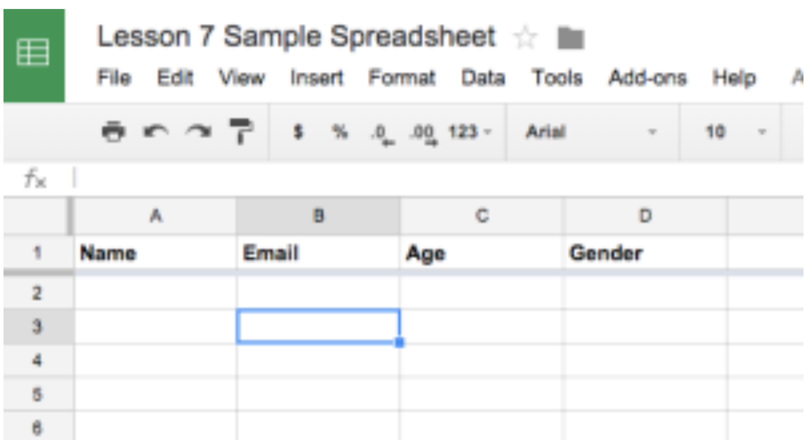
First, go in to Google Docs and create a new spreadsheet wherever you want it.

Then add a second page at the bottom, name the first one "Twitter" and the second one "Email."

On your Twitter sheet, create four columns for Name, Twitter Handle, Followers, and Klout



On the Email page, create columns for their Name, Email, Age, and Gender. You'll be able to get more info with a paid plan, but we'll just show the free one for now.



Alright, now let's fill out that Twitter data

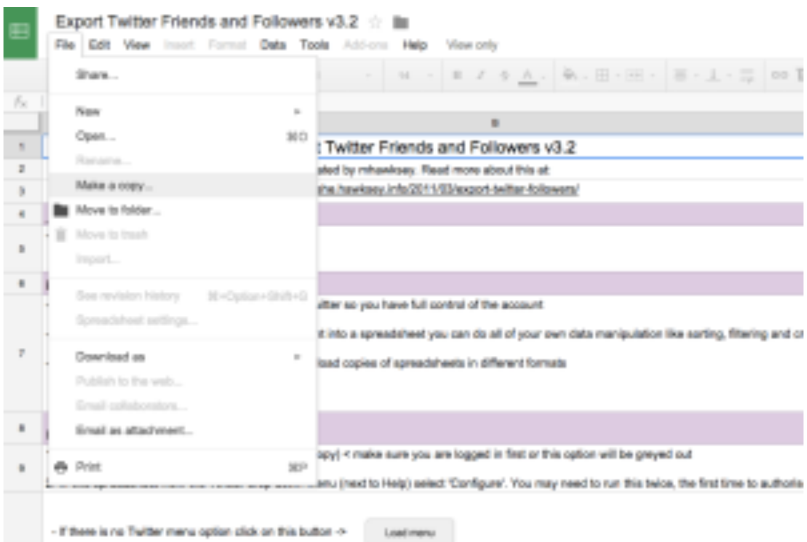
Getting Your Twitter Followers

First, you need to get a list of all of your Twitter followers. Twitter doesn't make this easy, but someone was nice enough to create a free program for it.

WARNING: Do not publicize your Twitter data or sell it. You can scrape your twitter for your own personal use (like this lesson) but not for any commercial use.

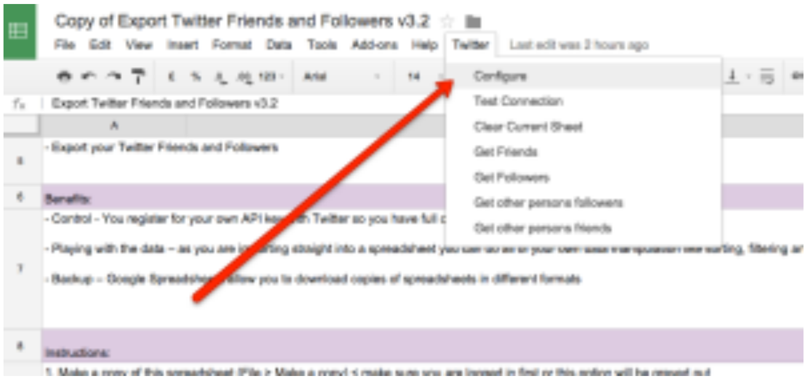
To do it, we're going to use a premade spreadsheet for pulling down your Twitter followers

First, [go here to get the spreadsheet](), then go to "File->Make Copy" to get one for yourself.

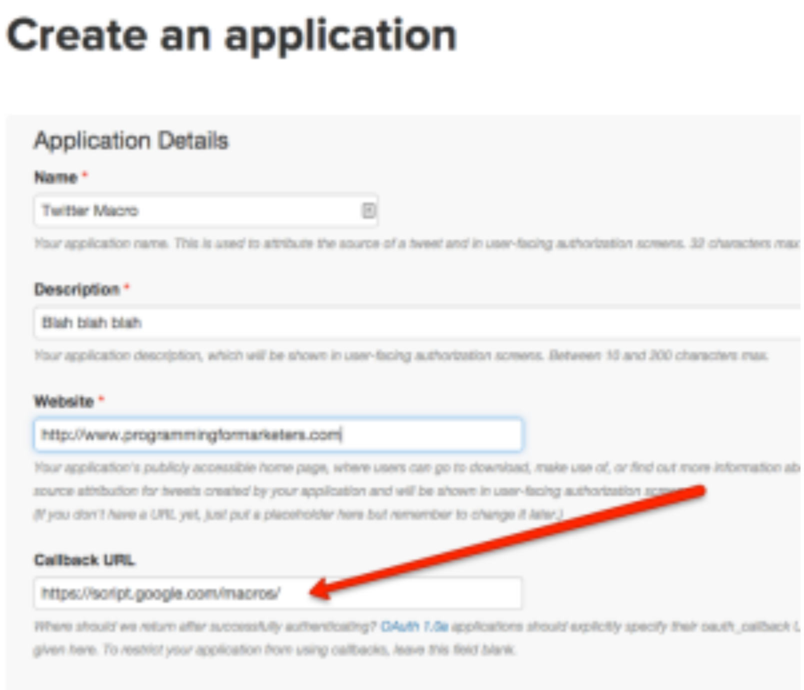


Once that's done, open up the copy you just created (it should automatically open).

Click on the "Twitter" button at the top by "Help" and then click "Configure" and follow the instructions for getting set up (these will be similar to some of the steps in Lesson 2)



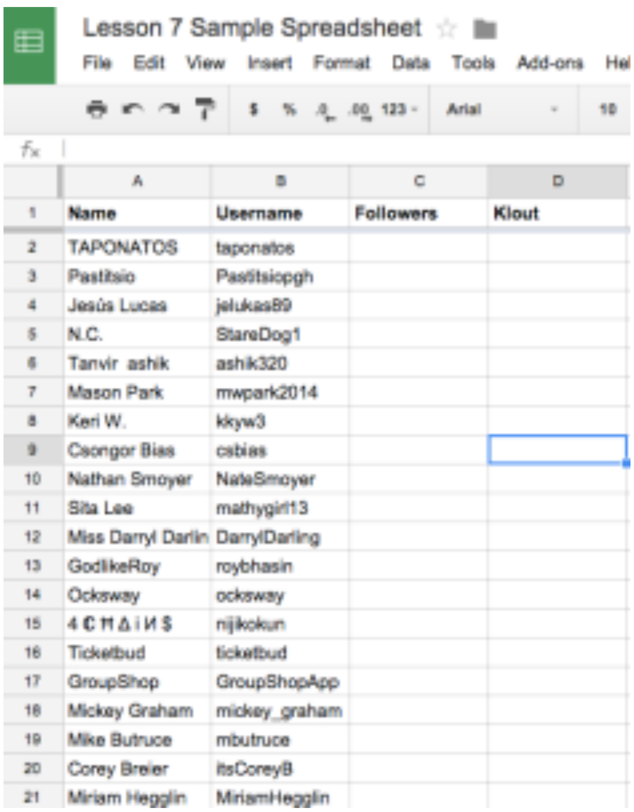
Don't forget the callback URL!



It might not work the first time, just log out of twitter and any other google accounts, and try again. It took me a couple tries.

Then you can go back to the Twitter button at the top and select "Get Followers,"

and it will add all of your follower info to the "Followers" sheet. Swap the "name" and "screen name" columns, then copy them over to your original spreadsheet.

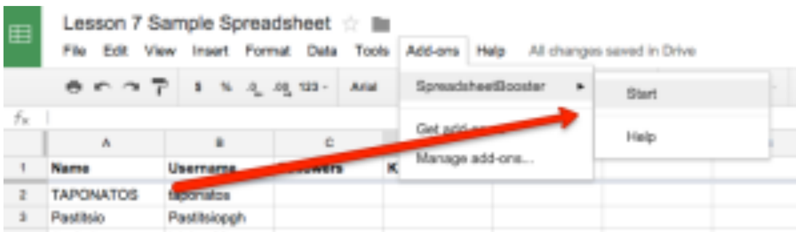


	A	B	C	D
1	Name	Username	Followers	Klout
2	TAPONATOS	taponatos		
3	Pastitsio	Pastitsiopgh		
4	Jesús Lucas	jelukas89		
5	N.C.	StareDog1		
6	Tanvir ashik	ashik320		
7	Mason Park	mwpark2014		
8	Keri W.	kkyw3		
9	Csongor Bias	csbias		
10	Nathan Smoyer	NateSmoyer		
11	Sita Lee	mathygirl13		
12	Miss Darryl Darlin	DarrylDarling		
13	GodlikeRoy	roybhasin		
14	Ocksway	ocksway		
15	4 C H Δ i И $	nijikokun		
16	Ticketbud	ticketbud		
17	GroupShop	GroupShopApp		
18	Mickey Graham	mickey_graham		
19	Mike Butruce	mbutruce		
20	Corey Breier	itsCoreyB		
21	Miriam Hegglin	MiriamHegglin		

Getting More Info on Your Followers

For this next step, you need to install "Spreadsheet Booster" Their on-boarding is pretty straight forward so you shouldn't have any trouble getting it set up.

Once it's installed, go back to your master spreadsheet, and refresh it so that Spreadsheet Booster pops up. If it doesn't, then go to Addons -> Spreadsheet Booster -> Start

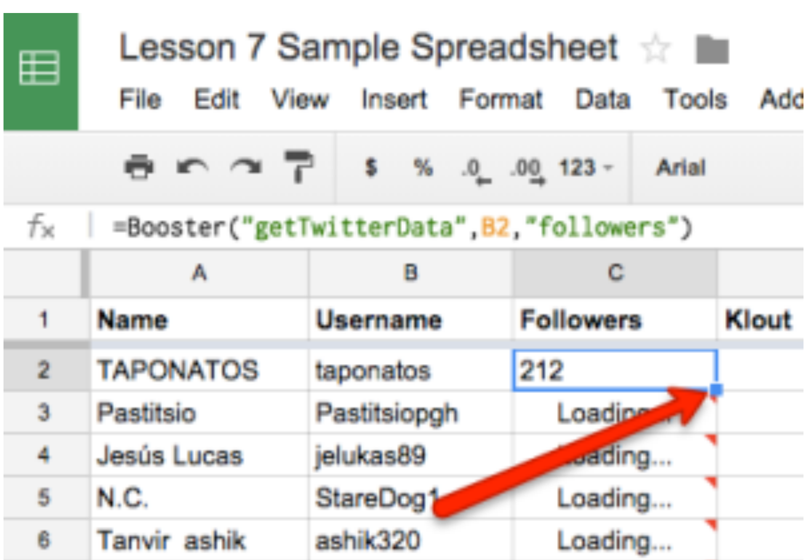


Now in cell C2 (the first one below "followers") paste in this code:

=Booster("getTwitterData",B2,"followers")

What that does is takes the username in the cell next to it, and finds all of its twitter followers.

Assuming that works, double click the little box in the bottom right of that cell to expand the formula down the entire column.



Then for Klout, you want to paste this code into D2:

=Booster("getTwitterData",B2,"klout")

And double click the box in the corner again so that you have the klout for all of your followers as well.

(If you're not familiar with Klout, it's a score from 1-100 that tells you how much social influence someone has.)

And now you know how much influence each of your followers has. This makes it easy to determine who would be best to reach out to and which of your users or fans can do the most to promote you.

You can also tie this in with your code from the earlier lessons. You could start following the followers of your followers who have the highest klout score as a way to follow people who will be more likely to follow you back.

Getting Data on Your Email Addresses

First you want to make sure that all of the email addresses you need information on are making their way to your new master spreadsheet.

So if you have an exisiting mailing list, you could export that to a CSV then copy the info into this new sheet.

You can set up a Zapier integration to send new signups to this sheet, such as this one.

And you can do things like import your FullContact contacts to the sheet as well.

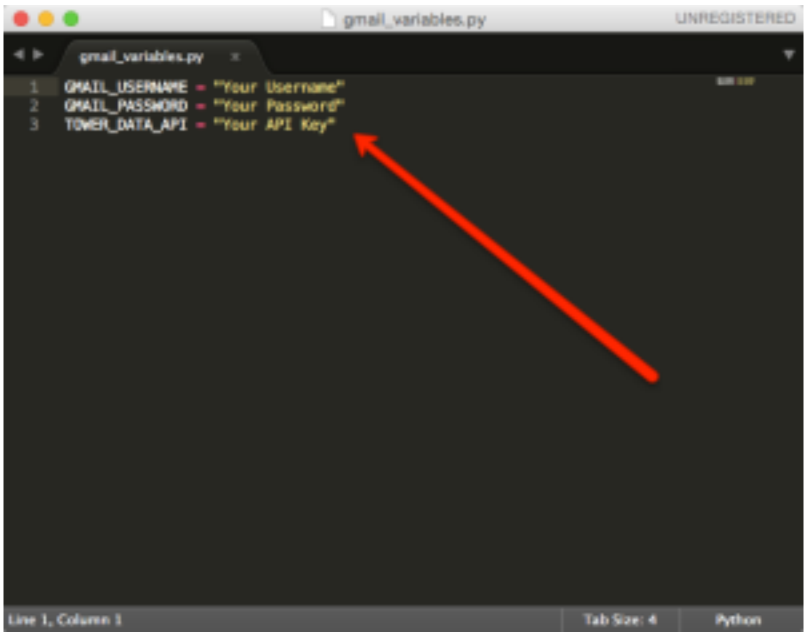
Once that's done, we're going to be using a tool called "TowerData."

To start, go to the site, sign up, and generate an API key.

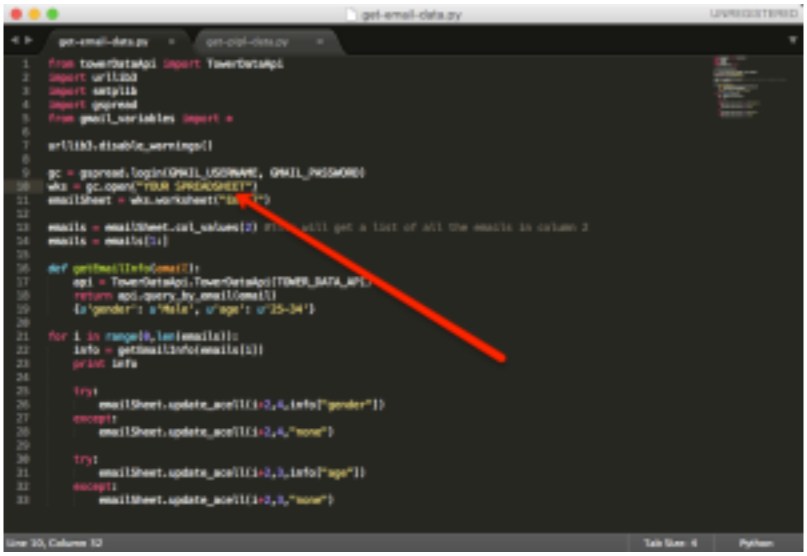
Once that's done, we can install the TowerData API for Python.

Using the guide to setting up python from Lesson 2, go to this page, download the ZIP, and run the setup (python setup.py install) from within the SDK/Python folder, not just the SDK folder.

Then you just download this code from our GitHub, add your credentials to the gmail_variables.py doc (including your towerdata API key),



and then add your spreadsheet info to the get-email-data.py function.



Now you can run “get-email-data.py” and it will take the emails in your spreadsheet, and find their gender and rough age, and add them to the spreadsheet. If you upgrade to the paid version of TowerData you can get much, much more info than that.

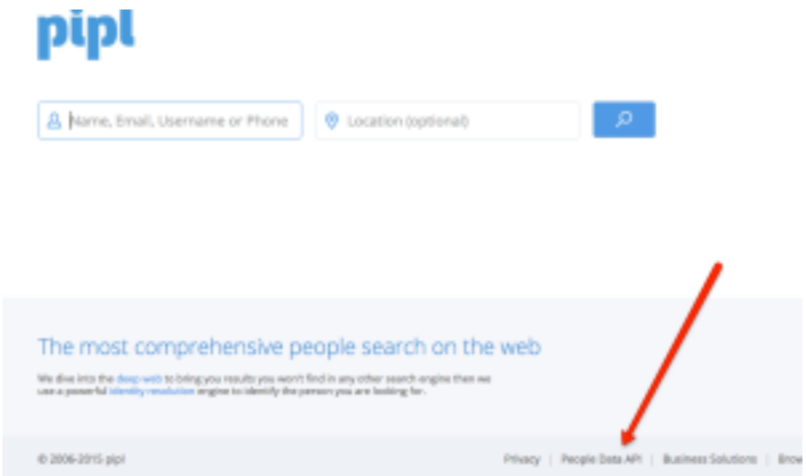
If you’re curious how it’s all working, read the comments in get-email-data.py

Getting Even More Email Information

[Pipl](#) is a great service that you can plug any email address, usernname, or just name into to get more information on someone.

And it has an API! We’re going to use it.

First, go to Pipl, scroll to the bottom and hit People Data API,



go through registration, and get a free demo key.

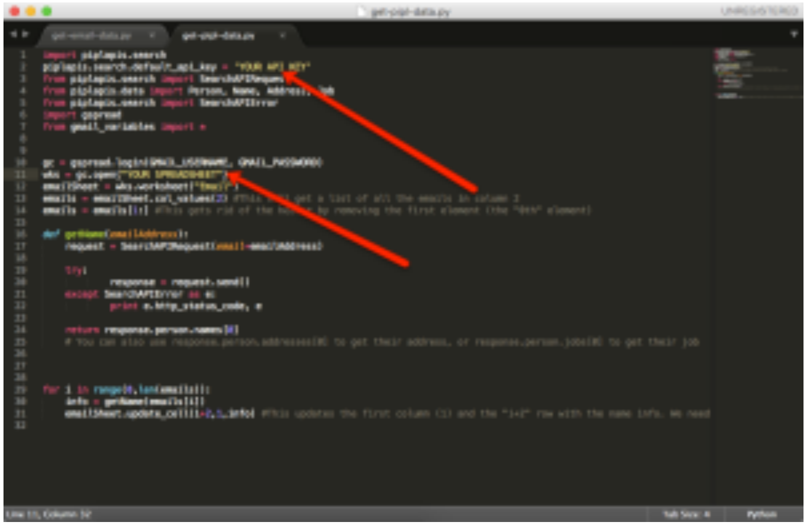
Then, go here to download and install the code that lets us use Pipl with Python.

Once you've done the python setup.py install, you'll be able to query Pipl's API (with your API key) to get information on people in your email list.

We gave you some code to look up their names and add it to the spreadsheet, which is the "get-pipl-data.py" function in the GitHub folder for this lesson.

What this code is doing is it's taking each email in the spreadsheet, running it through Pipl, finding their (most likely) full name, then adding it back to the spreadsheet.

All you have to do is modify it to have your Pipl API key and the name of the spreadsheet in it:



You could also modify it to add in their address and job if you wanted to.

Using This Info

One way to use this info is to cross reference the names on your email list, with the names of your followers

If you have a high influence person on your mailing list (and thus a user/customer) they'd be a good person to keep in touch with and try to please

One way to do this is to set up a [Google Alert]() specific for their name or company.

All you have to do is type their name or company in, hit "create alert" and you'll get an email whenever something important shows up in Google's rankings related to that topic.

Now you'll instantly know if anything big happens in the world that involves a top user/customer of yours, giving you a reason to reach out and engage them.

Pulling it All Together

Now that you have this information, you can do a few more useful things with it:

You could check your subscriber ratings in Mailchimp, AWeber, wherever else to see how good of subscribers each of these people are.

You can start searching Twitter for the names that you got for each of your email addresses in step 6 so you can follow and engage with your customers.

You can reach out to your customers with the highest social klout to ask for testimonials about your product which you use in blog posts, and then get them to help promote.

You could create a separate email list (or group) for high-influence people on your main list.

You could use the job, age, gender, address, etc. data to create new segments within your email list without having to have originally asked them for this info.

And plenty of other things, let us know if you have another cool application for it!

Conclusion

In the course of these eight lessons, you've started building landing pages, programmatically emailing people, automatically growing your twitter following, using SQL, Python, HTML/CSS, and more.

But this is just the beginning. It's meant to provide you a lens into the things you can do as a technical marketer, and we hope that it's whet your appetite to learn even more skills, hacks, and techniques to significantly improve your marketing efforts.

If you enjoyed these lessons, we'd love it if you went to programmingformarketers.com/review and left us an honest review.

If you want to learn more, then join us at http://www.programmingformarketers.com/ebook where you'll get access to all of the lessons from this book in PDF format, as well as other goodies including all future bonus lessons that we send out.

And feel free to reach out with feedback any time! You can reach us at justin@programmingformarketers.com, and nat@programmingformarketers.com

Stay Awesome,

Justin & Nat