Overview of Programming Languages for GIS

An essential element in designing many geospatial systems is the choice of what programming language (or languages) to use. Most of the exciting projects we can envision will involve at least some programming to customize existing tools, or to develop completely new ones.

There is an astounding variety of programming languages that are useful for geospatial professionals today. [Wikipedia lists over 600 languages(link is external)](http://en.wikipedia.org/wiki/List_of_programming_languages), which excludes the byzantine variety of dialects of [BASIC (link is external)](http://en.wikipedia.org/wiki/List_of_BASIC_dialects)past and present.

Characterizing Programming Languages

One important means of characterizing programming language is according to their type systems, that is, the rules by which one can assign meaning to variables or objects. One fundamental divide is between static and dynamic typing. Statically typed languages such as C, C++, C#, and Java evaluate type information at compilation time, and reject code that does not seem well-formed. The advantage of this is that many errors can be caught at compile time instead of run-time, and the earlier errors are caught, the easier they are to fix. The disadvantage of static typing is that it makes the code more verbose as type information is added, and you can sometimes spend a lot of time "making the compiler happy." Dynamically typed languages such as Python, JavaScript, PHP, and Ruby, conversely, delay type checking until run-time. Therefore, you often don't have to specify the type of a variable before you use it. The advantage of dynamic typing is in ease of programming; the disadvantage is that some errors will slip through until the program is running, and these errors can be difficult to interpret.

Common Programming Languages used with GIS

There are a large number of programming languages in use today, and no clear consensus on which language is “best”. We will focus here on those most relevant to GIS, which essentially means the most popular languages today, along with a few GIS-specific languages. I will add in a few general languages that show particular promise as well.

The table below represents several recent lists of languages from 2019 and 2018 (the first 3 columns are based on specific surveys of use/users, the basis of ordering for the last 3 columns was not explicit, but still represents a view by the author/authors. Only one of the lists, the one from GoGeomatics, is specific to GIS, and that one was produced by a single developer. The most important take-aways from this table are that: (1) there are many popular languages and diverse opinions on their relative merits, and (2) Python, Java, and JavaScript are in the top 5 on every list, with C++ and C# on 5 of 6 lists, PHP on 4 of 6, and C, R, Ruby, Rust, Shell, SQL, and TypeScript trailing the others with 2 lists each. For those interested in trends over time in programming language popularity (for most of the languages listed), check out the [2002-present TIOBE Programming Community Index timeline(link is external)](https://www.tiobe.com/tiobe-index/).

| Common Programming Languages used with GIS | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Rank** | **Business Insider - ranking based on GitHub** | **IEEE Survey of members (2018)** | **Stack Overflow Developer Survey (2019)** | **GoGeomatices: Top Languages in the GIS World** | **Full Stack Academy (9 best to learn)** | **Women Who Code** |
| **1** | JavaScript | Python | JavaScript | Python | JavaScript | Python |
| **2** | Java | C++ | HTML/CSS | JavaScript | Swift | Java |
| **3** | Python | Java | SQL | R | Java | JavaScript |
| **4** | PHP | C# | Python | SQL | C/C++ | Rust |
| **5** | C++ | R | Java | Java | Python | Kotlin |
| **6** | C# | PHP | Bash/Shell | C/C++ | PHP | \_ |
| **7** | TypeScript | JavaScript | C# | C# | Ruby | \_ |
| **8** | Shell | Go | PHP | - | C# | - |
| **9** | C | Assembly | C++ | - | Rust | - |
| **10** | Ruby | Matlab | TypeScript | - | - | - |

Below, we provide a brief description of the programming languages that show up on two or more of the lists. The first group (in alphabetical order) are those showing up on 5 or more lists, thus the ones that are generally most popular across the range of developers. Then (also in alphabetical order) we outline key features of those showing up on fewer (but at least 2) lists. These descriptions are followed by a perspective on making application-specific choices of languages that are well suited to particular GIS (and other geospatial) development tasks.

If you see any languages you feel are relevant for GIS but missing here, or you have good examples of an effective geospatial application that leverages any of the languages, please sing out by making a comment in the Canvas General Questions Discussion Forum!

Most Popular Programming Languages

1. **C++** — C++ is a general-purpose programming language (derived from C), with both procedural and object-oriented features, that is often used in embedded systems. Developed in 1985, its popularity has been in general decline since about 2004 (probably due to a steep learning curve and general complexity), but there are signs of a recent resurgence. While less popular than it once was, many software applications you use every day were written in C++ or use C++ for at least part of the implementation (ArcGIS, Windows OS, Firefox, MS Office, etc., etc.), so it isn't going away anytime soon.
2. **C#** — C# was Microsoft's answer to Java (see below), and is the flagship programming language for .NET. So, if you were starting to write a new add-on to ArcGIS, it would probably be one choice (the other is Java). VB.Net is now basically an alternative syntax for the same C# runtime.
3. **Java** — Java is very popular for web programming in general, and is many programmers' general language of choice. It is one of the contenders for the most popular Open Source GIS languages, used in the GeoServer and JTS projects for example. Java is the most commonly taught language in universities and is arguably the current king of the hill for programming languages in general.
4. **JavaScript** — JavaScript is the current leader for Web User Interfaces. Google Maps is a heavy user of JavaScript, Esri’s ArcGIS API for JavaScript supports web map development, and the leading open source web map client (OpenLayers) also is based on JavaScript.
5. **Python** — Python is often used as a scripting language, although many people swear by it for larger systems as well. It is currently gaining a lot of visibility as the primary scripting language for ArcGIS. It kind of functions like a new AML for ArcGIS, in that it provides a high level API to Get Things Done. This was a great choice on Esri's part, because Python makes a great "glue" language, and it is very easy to work with. It has many extensions, such as SciPython and Numerical Python. Beyond explicitly GIS applications, Python has become a key tool for those in Data Science (including Geospatial Data Science) building deep learning applications.

Other Important Programming Languages

1. **C** — C is the granddaddy of the family. When you need top performance, you use C, it is "close to the metal." This is great if you need to code a device driver, not great if you need to create a web app. Many lively open source GIS projects are written in C. For example, the Very Awesome GDAL (Geospatial Data Abstraction Library) is written in a combination of C and C++.
2. **PHP** — PHP is one of the best ways to whip up an interactive website and, thus, quite popular.
3. **R and S** — R and S are scripting statistical languages with a lot of very sophisticated spatial statistics that can use some of the output from ArcGIS. Plus, on its own, there are many spatial extensions to R that support a wide range of geospatial capabilities (e.g., see: [CRAN Task View: Analysis of Spatial Data(link is external)](https://cran.r-project.org/web/views/Spatial.html), [Spatial Data Science with R(link is external)](https://rspatial.org/), and [r-spatial(link is external)](https://www.r-spatial.org/).
4. **Ruby** — Ruby is an older language that has become more popular recently. Ruby got major traction due to Ruby on Rails, which made it easy to set up a database-backed application. This has been extended to web maps by GeoCommons. Some other interesting neogeography sites such as OpenStreetMap and WeoGeo utilize Ruby.
5. **Rust** — Rust is an open source language supported by developers with the Mozilla Foundation. It is conceptually related to C++, but designed to be safer in relation to security threats. The [Women who Code(link is external)](https://www.womenwhocode.com/blog/most-popular-coding-languages-for-2019) site says Rust has “loads of promise for game engines, VR simulation engines, VR controllers”, thus as more GIS-based VR develops, Rust may become more popular in our community.
6. **Shell** — Shell is a scripting language for use in Unix (and Linux and Mac OS) environments. It allows a user to chain Unix commands together, having the system execute them as an event. For one example of using Shell with GIS, see this guide for [Shell scripting with the GRASS GIS(link is external)](https://grasswiki.osgeo.org/wiki/Shell_scripting).
7. **SQL** — SQL is used as a database access and control language. SQL is at the heart of many GIS operations. SQL is a great example of a language that has survived for a long time. It has survived so long because it is declarative instead of procedural. That is, SQL statements tell *what*you want to happen, not *how*you want it to happen. Therefore, implementation details are hidden and can change over time. This means SQL is set to remain relevant into a world of concurrent processing, which we will discuss in this week's tech trend.
8. **TypeScript** — TypeScript is an object-oriented programming language that is a superset of JavaScript. It has been described as “[JavaScript that scales(link is external)](https://www.typescriptlang.org/)”. As with JavaScript, support exists for using TypeScript with the ArcGIS for JavaScript API and the and the Google Maps JavaScript API. There is also support for TypeScript with the Bing Maps SDK and for using Leaflet in TypeScript, among others.

Making Application-Dependent Choices

Many GIS projects leverage more than one of the languages detailed above. This is because different tasks in GIS are better supported by some of the languages than others. While different developers have different opinions on the best match of language to task, one [Task-Language list (link is external)](https://gogeomatics.ca/what-are-the-top-programming-languages-in-the-gis-world/)is worth a look (specific to GIS-related work) was produced in 2016 by Florin-Daniel Cioloboc (Posted on: November 16, 2016, accessed on August 8, 2019).

* GIS Scripting and applications (Python, R)
* Data processing, analysis, and modeling (Python, R)
* Web Mapping (JavaScript, Python)
* Geospatial databases (SQL)
* MapServers (Java, C# .NET, C++)
* GIS heavy-weight development (Java, C/C++, C#)
* Mobile development (Android, iOS, JavaScript)
* Geospatial libraries (JavaScript, Python, Java, R, C/C++)

Learning GIS Programming

**APRIL 28, 2017BY**[**CAITLIN DEMPSEY**](https://www.gislounge.com/author/caitlin/)[**GIS SOFTWARE**](https://www.gislounge.com/software/)

*Advertisement*

*Share this article:*

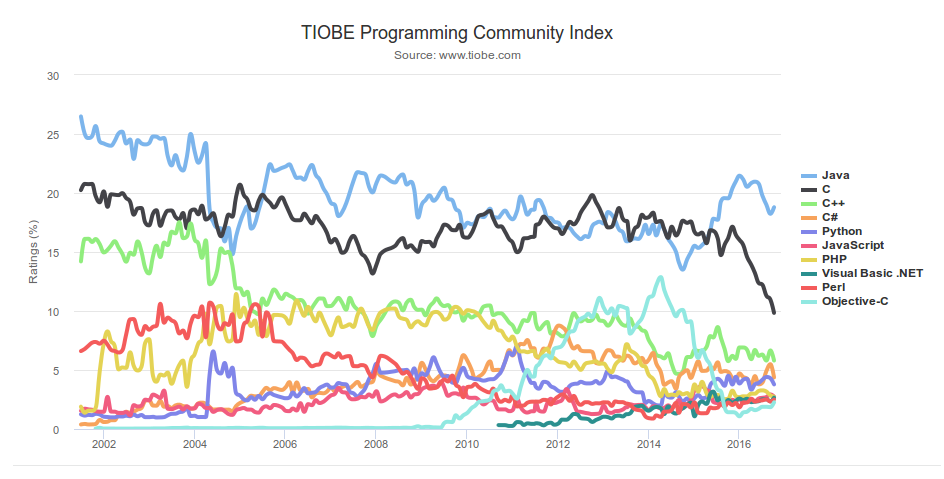
[Facebook](https://www.facebook.com/sharer.php?u=https%3A%2F%2Fwww.gislounge.com%2Flearning-programming-for-gis%2F)[Twitter](https://twitter.com/intent/tweet?text=Learning%20GIS%20Programming&url=https://www.gislounge.com/?p=4303&via=gislounge)[Reddit](http://www.reddit.com/submit?url=https%3A%2F%2Fwww.gislounge.com%2Flearning-programming-for-gis%2F&title=Learning+GIS+Programming)

Having some competence in programming has become a critical requirement for many geospatial positions.  The majority of GIS analyst level positions require some expertise in programming and application development.  So which languages should you learn for GIS programming?

Python is a popular scripting language in the GIS arena, especially for those wanting to work with Esri’s [ArcGIS](https://www.gislounge.com/new-learn-arcgis-book/) and open source [QGIS](https://www.gislounge.com/how-to-make-the-most-of-qgis/) desktop products. Learning Python is the way to go for scripting and creating tools inside of the ArcGIS framework.  C# is another recommended programming language that is relatively easy and intuitive to learn.  Learning .NET will also allow you to customize ArcGIS.  Using VBA is on the decline and [Esri will no longer support VBA](http://downloads2.esri.com/support/TechArticles/ArcGIS_9.3.1_and_10_Deprecation_Plan.pdf) beyond ArcGIS 10 (*Users who have custom functionality built using VBA should actively plan strategies to rewrite their applications using a supported development language such as Python, VB.NET, or C# so their applications continue to be operational and supported in future releases of ArcGIS.*).

For web-based applications development, learn HTML, JavaScript, and CSS as a baseline.  Java, PHP, and ASP.NET are additional programming languages for more advance web development.

**What are the Top Programming Languages in the GIS World?**

[](http://www.gogeomatics.ca/wp-content/uploads/Selection_002.png)*Image source: www.tiobe.com*

Every now and again we hear about the [importance of programming](http://www.directionsmag.com/entry/gis-jobs-skills-in-demand/471742) for Geographic Information Systems(GIS) professionals, that is quickly becoming a key in the evolution of one’s career. It is argued that even the ones that have a limited grasp of [**programming**](http://www.directionsmag.com/entry/should-all-gis-users-learn-to-code/234861) can see the **benefits**.

This article is  addressed to a larger audience of GIS professionals who are not only already familiar with programming, but are for those interested in getting a head start or further their knowledge.

**Why would these be useful?**

Programming is used with many goals in mind, whether it is automating geoprocessing tasks or implementing an algorithm for all sorts of tasks, programming goes a long way in problem solving. It helps us tackle those tasks which require a different degree of operations, where the problems we are facing need some specific manipulation that is not provided.

This is one of the reasons why many GIS software allow access to API for customizing the application according to your specific needs.

How will this go? We will start by exploring what are the most used programming languages in the industry. Then we will look at each separately to see where it is most useful and why. Lastly, we shall see what are the most notable examples of its usage.

**Python**

Whenever I think about GIS programming, [Python](https://www.python.org/) comes to my mind.

My experience tells me that QGIS, and especially, ArcGIS users have heard about the modules for scripting in Python. With that being said,  Python has become one of the most important languages used in the field. How so? It is easy to learn, thanks to having an outstanding documentation and easy to pickup syntax, plus the variety of tutorials available online for free. Moreover, there is a great support for data analysis and processing through the likes of [Numpy](http://pro.arcgis.com/en/pro-app/arcpy/get-started/working-with-numpy-in-arcgis) or [Pandas](http://www.esri.com/esri-news/arcuser/winter-2015/integrating-arcgis-and-scipy).

Python is incorporated into ArcGIS, QGIS, GRASS GIS, gvSIG, and many other open source projects, that make the language worth knowing.  At the same time, there are quite a few libraries [Shapely](http://toblerity.org/shapely/),[Fiona](http://toblerity.org/fiona/fiona.html) and[Rasterio](https://github.com/mapbox/rasterio), [Folium](https://github.com/python-visualization/folium), and many [others](https://github.com/SpatialPython/spatial_python/blob/master/packages.md) that appeared in the past couple of years.

Python can be compared to as the swiss army knife for GIS.

**JavaScript**

Without a doubt, [JavaScript](https://en.wikipedia.org/wiki/JavaScript) is one of the core languages of the web. Being present in pretty much every browser, website or web application, this makes it a reasonable choice for development of **web mapping** applications. Think about it, the most popular applications in the past few years appeared, thanks to the capabilities of this language.

**Notable examples:** ArcGIS Web APIs, MapBox, CARTO, Google Maps API, OpenLayers, Leaflet, the list can go on.

What is becoming more obvious these days is the tendency for applications to offer [more 3D](https://www.geospatialworld.net/article/golden-era-for-3d-gis-data-analysis/) web-based capabilities. And you guessed it, JavaScript comes to the rescue. Just have a look at CesiumJS (Do you have a link for CesiumJS), [OSM Buildings](https://osmbuildings.github.io/OSMBuildings/) project, and [MapBox  GL supports 3D](https://www.mapbox.com/blog/mapping-3d-buildings/) visualization.

It’s pretty safe to say if you are considering web mapping, webGIS or even 3D, then make sure you **add JavaScript to your toolbox**. The upcoming years are looking favorable to this programming language.

**R**

[The language](http://www.r-project.org/) for statistical computing, graphics, data science, and **geospatial analysis**.

It’s becoming more popular in geospatial realm through the [initiative](http://data-analytics.net/cep/Schedule_files/geospatial.html) (link is confusing, with how you are trying to connect it to your article, try explaining it more to the audience) of developers from the R community. As we recently saw that ESRI decided to provide [support R](https://blogs.esri.com/esri/esri-insider/2015/07/20/building-a-bridge-to-the-r-community/) in their ArcGIS suite. We can interpret this as a pretty clear sign that things will only get better for R geospatial community.

You can also get [R in QGIS](https://www.r-bloggers.com/qgis-open-source-gis-r/) as well, so no more worries that you’ll have to run your scripts elsewhere. Thanks to this there are pretty good signs that R will play an important role in the years to come.

**Notable examples:**rgdal, rgeos, ggplot2, ggmap, leaflet for R. You can find a comprehensive [list](https://cran.r-project.org/web/views/Spatial.html)on the R-project’s website.

**SQL**

Do you remember those introductory courses in GIS where you had to find attributes or location? This is where you might have had to input a query that is written in SQL. It’s one of the languages with a long history in GIS, that is being incorporated in many applications. You can have look at OpenStreetMap’s [database overview](http://wiki.openstreetmap.org/wiki/Databases_and_data_access_APIs) in case you want to know more.

To sum up, SQL is essential in GIS as you will find yourself working with geospatial databases and can help you manipulate databases.

**Notable examples**: ArcGIS, PostGIS, CARTO, QGIS, MapInfo, and many other GIS software.

**More heavy-weight languages**

The languages we mentioned so far are probably the most practical from the point of learning and getting quick results. The ones that follow are the ones we might call as “heavy-weight” as they are generally found as the backbone of larger GIS software.

**Java**

Surely if you ever heard of Java, you probably heard about it being used in enterprises, and in various industries. But why here?

[Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29) is used when you develop the backbone of the application, especially for **desktop applications** as an alternative to C/C++ or C#. Nonetheless, Java has proven itself as a particularly useful language even for **map servers**. As you might have known or not, one of the most popular web map server, GeoServer is written in Java.

While, more verbose and not as practical as Python, it is a general purpose programming that can help you go the long way in software development.  Keep in mind, it is one of the most popular languages used today with strong and well-tested APIs, huge code base, so it’s not going to go anytime soon.

**Notable examples**: GeoServer, GeoTools, ArcGIS Runtime SDK, gvSIG, OpenJump, uDig, WhiteBox (GAT), Java Topology Suite etc. Last but not least, Android apps are most of the time built with Java, thus you’d have to know it if you want to go in that direction.

**C/C++**

One of the most well-known languages among developers, and useful for **desktop applications**. Whether it was for Windows or Linux, [C++](https://en.wikipedia.org/wiki/C%2B%2B) has a long line of successful applications it supported and still does. I included C alongside C++ since the examples provided don’t necessarily rely fully on one or the other.

If the software was built-in the ‘90s or early 2000, then it is more than likely that it was built with C++.

**Notable examples**: ArcGIS, QGIS, SAGA GIS, GRASS GIS, MapServer, ILWIS, Mapnik, GDAL, Orfeo Toolbox, and the list goes on. Even today it still remains a popular language among developers, however it is not the sort of language you use on a general daily basis as a GIS user.

**C#**

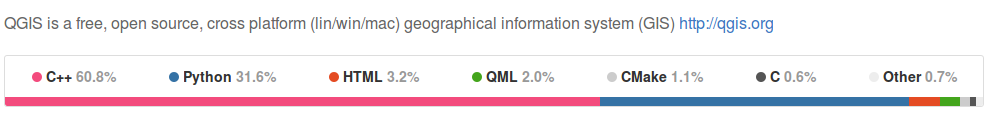
Microsoft’s baby. Together with .NET, it represents a well-known framework for **software development**. You could have heard of it without actually being a coder as with most things mentioned in this article. It is popular among proprietary software companies, especially for ESRI.

**Notable examples**: [ArcGIS](http://desktop.arcgis.com/en/arcobjects/latest/net/webframe) (ArcObjects),  ArcGIS Runtime SDK .NET, NetTopologySuite, MapWindow.

**Let’s sum up the whole thing**

The easiest way to find out what languages are being used in a scenario is to regularly check out the **Open Source Geospatial Foundation**([**OSGeo**](http://www.osgeo.org/)). After that, ESRI provides information about what they work with. Just follow their blogs, webinars, [open source projects](https://esri.github.io/), conference presentations, the same applies to other companies. What I also suggest is following popular GIS websites and specialized magazines. There are so many ways you can tackle this if you are interested.

Keep in mind that **a software is most of the time built with multiple technologies** but in general a few of them had the most influence as you can see below in QGIS’s case.

[](http://www.gogeomatics.ca/wp-content/uploads/QGIS_github.png)

Source: from QGIS [GitHub repository](https://github.com/qgis/QGIS) (as of November 2, 2016)

There are plenty of languages we didn’t include as they do not come as close as the ones mentioned before, nevertheless there some very interesting mentions: Ruby (RGeo library), PHP (GeoPHP).

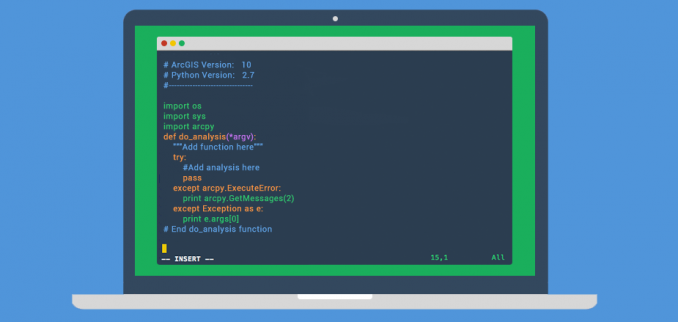
Before going, we would like to show you a summary of the various programming languages that are connected to the variety of technology that is  summed up in the article:

* + GIS Scripting and applications (Python, R)
  + Data processing, analysis, and modeling (Python, R)
  + Web Mapping (JavaScript, Python)
  + Geospatial databases (SQL)
  + MapServers (Java, C# .NET, C++)
  + GIS heavy-weight development (Java, C/C++, C#)
  + Mobile development (Android, iOS, JavaScript)
  + Geospatial libraries (JavaScript, Python, Java, R, C/C++)

All in all, if you want to have a bigger view of what programming languages are being used for specific GIS software, we suggest that you have a look [here](https://en.wikipedia.org/wiki/Comparison_of_geographic_information_systems_software).

**Free GIS Programming Tutorials: Learn How to Code**

**WHAT SUCCESSFUL, SELF-TAUGHT GIS PROGRAMMERS EAT FOR BREAKFAST**

GIS Programming in Python

By: GIS Geography · Last Updated: March 5, 2020

**Learn to write code for free in any GIS programming language**

What do successful self-taught [**GIS programmers**](https://gisgeography.com/gis-developer-job-profile/) eat for breakfast?

We eat healthy dose of Python, JavaScript, SQL, VB.NET, C++, HTML, CSS… In that order, these are the most popular GIS programming languages.

It’s not necessary to have programming skills to [**land a job in the GIS industry**](https://gisgeography.com/10-gis-career-tips-find-gis-job/). But it’s a feather in your cap if you do. And it will certainly help.

If you’re just starting out, here are some of the best, free GIS programming resources available to pave your way to coding competency:

**GIS Programming in Applications – Python, C++, .NET, C#**

Python has been a standard language in GIS because Esri and open source tend to gravitate toward it. Because you can leverage so many [**Python libraries**](https://gisgeography.com/python-libraries-gis-mapping/), many consider this GIS programming language to be the front-runner.



In addition to Python, GIS programmers use C++, C# and .NET. C++ lets you work in multiple environments. While C# and the .NET languages offer you good development tools and interaction with Windows-based software.

We suggest to learn Python first because it’s usually the first language a company looks for.

Here are 3 starter courses to start your journey into Python programming:

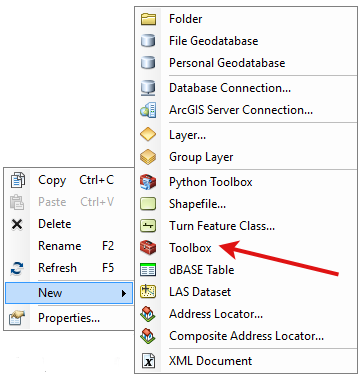
The [**Python Scripting for ArcGIS Free Book by Esri**](http://esripress.esri.com/bookResources/index.cfm?event=catalog.book&id=9) guides you to create custom geoprocessing tools and learn how to write Python code in ArcGIS.

Next, the [**Programming Foundation with Python**](https://www.udacity.com/course/programming-foundations-with-python--ud036) in Udacity helps you actively learn to code in Python for free.

Finally, our [**list of free Esri training courses and MOOC**](https://gisgeography.com/free-esri-training/) teaches you Python scripting language and automate GIS tasks in Esri ArcGIS desktop.

But you won’t truly learn Python unless you apply it:

For example, build your own toolbox to manage, process or display GIS data of your own.



A good example would be to analyze a large Excel file. Import that data into ArcGIS and run an analysis on that shapefile with your Python script.

Show off your new skills on your portfolio page. Add to your CV that you have a working knowledge of Python programming.

**[](https://gisgeography.com/wp-content/uploads/2016/03/GIS-Programming-Banner.png)**GIS Programming – Python, JavaScript, R and More

The big advantage of using Python is this:

You automate workflow and repeat redundant tasks. If you can **save a company time and money** with a working example, you’re as good as gold.

If you get good at Python, it isn’t that hard to transfer your skills to JavaScript. This is the way to go for developing your own [**web mapping applications**](https://gisgeography.com/best-maps-webmaps/).

**Make Your Webpages Interactive with JavaScript**

Web development is all the buzz these days.

…And the GIS industry is no exception as it is gravitating more these days with [**cloud-based platforms like ArcGIS Online**](https://gisgeography.com/esri-arcgis-online-agol/).



Starting with the basics, HTML is the markup language that gives structure to web pages like headings, tables and lists. While CSS stylizes webpages with layouts, it also provides color, fonts and size.

But HTML and CSS are easy to learn. Your biggest hurdle is JavaScript. If you really want to make your webpage dynamic, then take a look at JavaScript.

Because JavaScript is run in a browser, it manipulates the behavior of your web-based content. In fact, [**Esri’s Javascript API**](https://developers.arcgis.com/javascript/) is the successor of Flex and Silverlight.

***READ MORE:*** [**15 High-Tech Webmaps and Webscenes Examples Using Esri’s JavaScript API**](https://gisgeography.com/esri-javascript-api-webmaps-webscenes/)

Actually, HTML, CSS and JavaScript all work together to create dynamic web content. In addition, the [**Dojo toolkit**](https://dojotoolkit.org/) provides language utilities and UI components.

Again, the assumption is starting at zero. These 3 courses deliver a good base in JavaScript and some HTML/CSS:



[**Khan Academy’s Introduction to JavaScript**](https://www.khanacademy.org/computing/computer-programming/programming) are very introductory and won’t take you very long to complete. This one starts with the basics and ends with object-oriented design.

[**Udacity’s Javascript Basics**](https://www.udacity.com/course/javascript-basics--ud804) courses will help expand on the info learned above. Like most introductory classes, it helps you dive deeper into whatever direction you would like.

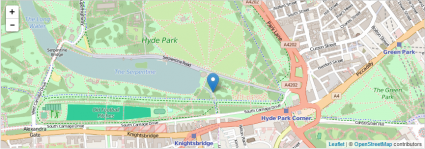
[**HTML/JS: Making Webpages Interactive**](https://www.khanacademy.org/computing/computer-programming/html-css-js) tells you how to include JavaScript libraries on your webpage so that you can use functionality built by other web developers.

From here, it’s time to challenge yourself. Put your knowledge to action with a project of your own. For example, create your own personal web map project by starting simple. Take latitude and longitude coordinates and put them on a web map with Leaflet, ESRI API or D3.

You’ve gone through the necessary courses. You’ve put your knowledge to action. It’s time to give your resume a boost and tell employers. At this stage, it wouldn’t be unreasonable to add ***working knowledge of HTML, CSS, JS*** upon completion.

**Build Sophisticated Webmaps using Leaflet, OpenLayers and More**

Rich web maps often use open-source JavaScript libraries like Leaflet.js to really bring them to life:

**[](https://gisgeography.com/wp-content/uploads/2016/04/Leaflet-Map.png)**Example of Leaflet Map

JavaScript libraries are code someone else has written to make life easier for the rest of us. This is the same how Python modules are distributed. For example, JQuery, three.js and bootstrap are JavaScript libraries in use today.

When you use Leaflet, you can build your own web map from this JavaScript library. This is why JavaScript is so good with interactive viewing in GIS.

[**Leaflet.js**](http://leafletjs.com/) is well documented and easy to learn. You can do almost anything with JavaScript from spatial analysis to robust basemap servers.

…As technology improves, web development is going to require using 3D visualizations tools such as WebGL and Esri’s JavaScript API 4.

Here is a course to fine-tune your GIS programming skills:

Use [**OpenLayers 3 Examples**](https://openlayers.org/en/latest/examples/) gives code examples of clusters, animations, GeoJSON, heatmaps, WebGL and more.

But you won’t truly learn how to create dynamic web maps unless you apply it.

QGIS web map example

Test your knowledge with a more complex web map project. For example, make a web map with at least 2 toggleable layers. From this project hopefully you will find why HTML, CSS, JavaScript, XML, JQuery and AJAX are valuable in web development.

As GIS technology shifts to the cloud, web development may just be the better career choice.

**Databases – SQL and UML**

Almost all modern database systems (DBMS) can use SQL. A big part of GIS is database entry, editing and maintenance. For example, SQL queries, inputs and deletions all fine-tune your data.



You should be familiar with SQL and be able to perform SELECT, INSERT, MODIFY and DELETE statements. Joins, relates and further SQL knowledge is greatly valued in the field.

Here are some courses to brush up your SQL:

Khan Academy’s [**Intro to SQL: Querying and managing data**](https://www.khanacademy.org/computing/computer-programming/sql) helps you manage data in a relational database with SQL. This free course teaches you how to use SQL to store, query, and manipulate data.

Udacity’s free course on [**Intro to Relational Databases**](https://www.udacity.com/course/intro-to-relational-databases--ud197) gives you the necessary knowledge to write code using a database as a back-end.

When given a database, explore the data models within it. Understand the structure of the database. This database design structure are often represented in UML diagrams.

It’s often the case you can use design tools like Microsoft Visio. Most design work for data models use it.

But the best advice is to:

Practice, practice, practice!

Come up with a project and practice using your new SQL skills. If an employer asks for a portfolio, you have some good examples to share.

**R Spatial**

Statisticians and data miners use R for open statistical software development and analysis.

What you may not have known is that it’s already being adopted in GIS.



This is because R can visualize and analyze spatial data. It places emphasis on statistics, but it can do both.

It’s not a bad idea to learn some programming languages/packages that are not strictly tied to ArcGIS. And for this reason, R is often recommended:

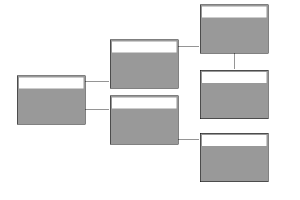
Robin Lovelace and James Chesire’s [**Introduction to Visualizing Spatial Data in R**](https://cran.r-project.org/doc/contrib/intro-spatial-rl.pdf) give you hands-on experience with R’s popular graphics package ggplot2

[**R Spatial Cheatsheet**](http://www.maths.lancs.ac.uk/~rowlings/Teaching/UseR2012/cheatsheet.html) is for Geospatial Data in R and Beyond. The key functions and manipulations of spatial vector and raster data.

**Object-Oriented Programming (OOP)**

Object-oriented programming (OOP) is a must-have skill for a GIS programmer. OOP is about maintaining code as objects and reusing code through instantiation.

You can apply OOP in conjunction with your programming language of choice.



Most GIS development is leaning toward the OOP paradigm. And you should too! Mastering the concepts of OOP will propel your career forward in GIS programming.

Udacity’s [**Object-Oriented JavaScript**](https://www.udacity.com/course/object-oriented-javascript--ud015) helps you build your own blocks of code as libraries and avoid copying and pasting lines of code.

**GitHub and CodeAcademy Coding Community**

Finally, you may be wondering why so many people use GitHub.

GitHub is like a Facebook for developers. It’s an open community that helps developers see what their peers are working on.



Developers can inspect the entire history of a project by version. They can also study a projects’ code and modify it on their own.

**How to use Git and GitHub (Udacity)** – Use version control over the life of a project and optimize collaboration through GitHub. While you’re doing it this may seem pointless. Don’t skip it! ([**How to use Git and GitHub**](https://www.udacity.com/course/how-to-use-git-and-github--ud775))

Over 25 million people use CodeAcademy because it’s great for learning syntax. It’s not only for syntax, but you can interactively take lessons in Python programming.

But CodeAcademy comes at a cost for the PRO version courses. Test it out, and it might be a good investment to start in Python.

**Learn to Code in Python Interactively (CodeAcademy)** – If you want to just start coding somewhere, CodeAcademy has the interface for you to do so. ([**Learn to Code Interactively, For Free**](https://www.codecademy.com/learn/python))

**From Zero to Somewhat of a Coding Hero**

Code all day.

Debug all night

Knowledge in GIS programming is a nice showpiece for your CV. Learn multiple GIS programming languages and you’ll be a grand slam.

Agree or disagree with the article? Let us know with a comment below.