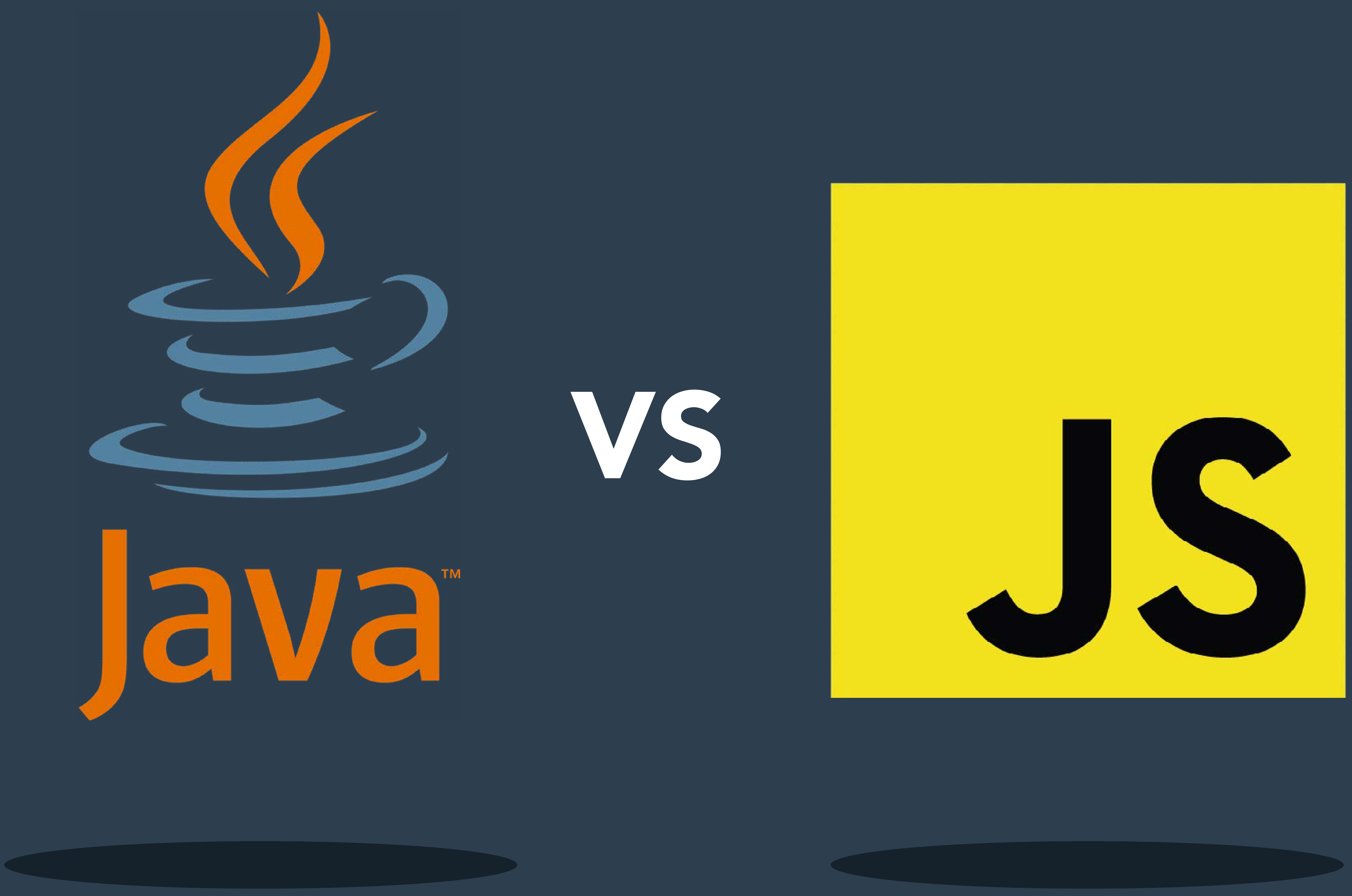
HubSpot | GENERAL ASSEMBLY

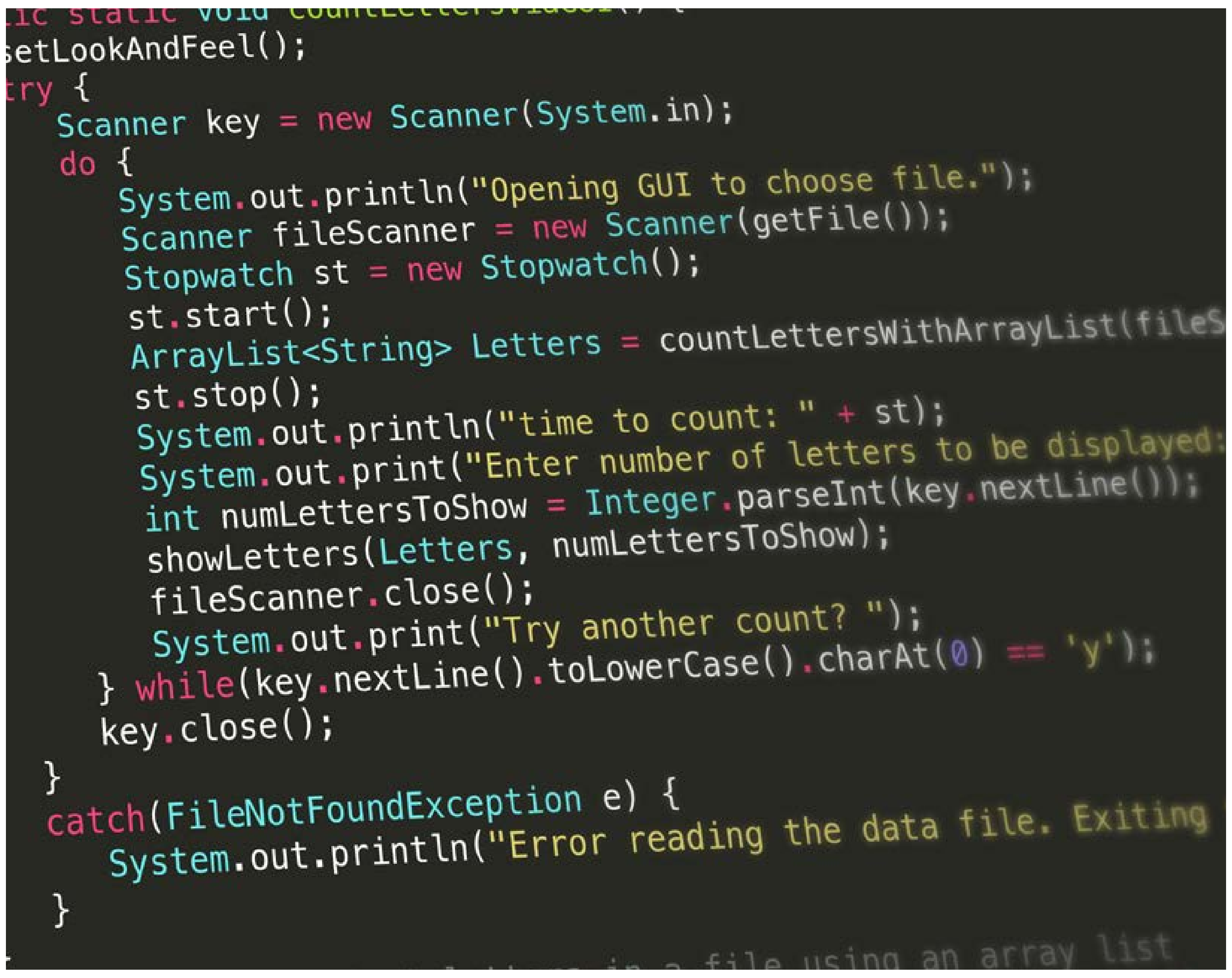
The Ultimate Guide to Java & Javascript for Beginners



Introduction

If you've ever played Minecraft, used most Android apps, listened to music on Spotify, updated your job status on LinkedIn, or booked a ride on Uber, you are using a product built with Java.

If you've ever played Candy Crush, used Facebook, Instagram, or really used any application that has an interface that you interact with on the internet, then you've experienced JavaScript.



```
ic static void countLetters
setLookAndFeel();
try {
   Scanner key = new Scanner(System.in);
   do {
      System.out.println("Opening GUI to choose file.");
      Scanner fileScanner = new Scanner(getFile());
      Stopwatch st = new Stopwatch();
      st.start();
      ArrayList<String> Letters = countLettersWithArrayList(fileS
      st.stop();
      System.out.println("time to count: " + st);
      System.out.print("Enter number of letters to be displayed:
      int numLettersToShow = Integer.parseInt(key.nextLine());
      showLetters(Letters, numLettersToShow);
      fileScanner.close();
      System.out.print("Try another count? ");
   } while(key.nextLine().toLowerCase().charAt(0) == 'y');
   key.close();
}
catch(FileNotFoundException e) {
   System.out.println("Error reading the data file. Exiting
}
                     file using an array list
```

JavaScript has the word "Java" in it. Often, this leads many to assume that Java and JavaScript are somehow related. Even worse, the two languages are sometimes mistaken for each other. Part of the reason their names are so close stem from a marketing approach that took place in the mid-1990s when JavaScript was introduced to the world. But before we jump into that story, let's make one thing clear: **Java and JavaScript are two different programming languages with two distinct purposes.**

Java and JavaScript are both among the most used languages of all time. At a high level, Java is a general purpose programming language, while JavaScript was originally a web technology that helped you create interactive web applications. Since the introduction of Node.js in 2010, JavaScript is capable of doing just about anything Java or any other programming language can do.

Java is used to build complex software like web applications, mobile applications, big data technologies, and more. Java is usually the first choice for developing enterprise programs because of its powerful features delivering high performance. In addition to its high performance, it also makes applications secure and easily scalable. Some examples of applications built with Java include Google Android Applications, several mobile apps like CashApp, Spotify, Signal, and even Twitter.

Note:
All the above applications also use JavaScript to create the user experience we all enjoy.

JavaScript, on the other hand, began its reign during the Web 2.0 revolution. In the beginning, JavaScript focused on client-side development to add functionality to static web pages and JavaScript, on the other hand, began its reign during the Web 2.0 revolution. In the beginning, JavaScript focused on client-side development to add functionality to static web pages and applications. This added functionality provided a better interactive user experience. It is currently the most popular language for building web-based applications because anything that requires smooth user interaction, uses JavaScript. Any application that has interactive experience on the web browser, like Google Maps, Facebook (Meta), Twitter, Instagram, Uber, Airbnb, Netflix, and other smartwatch apps use JavaScript to various degrees.

This guide will show you what learning Java and JavaScript means for your career, why these languages are so popular, what are some common uses of each, how they differ from each other, and which one is right for you. We will review the basic syntax of each language, the role each one plays in the industry, and provide resources, tools, and additional information to enrich your understanding of them.

HubSpot

Build your optimized website on the HubSpot CMS.

Why do the work when HubSpot can do it for you? Use HubSpot's CMS to build your website without being a programming expert.

HubSpot's customizable, marketer and developer-friendly CMS offers: Get started

- Flexible website themes
- Built-in integrations with the HubSpot CRM and 1,000+ other tools
- SEO recommendations
- Adaptive testing
- Drag-and-drop editor
- Serverless functions
- And more!

Get Started

General Assembly

General Assembly equips you with the skills needed for a career in today's high-growth roles: Coding, Data or UX Design.

Launched in 2011, General Assembly has graduated more than 40,000 global alumni from full-time & part-time courses.

Sign up for a free intro to class or attend an info session!

Why choose General Assembly?

- Learn at your own pace from industry experts.
- Work with a career coach to achieve the career you love.
- Find the payment schedule that fits your needs.

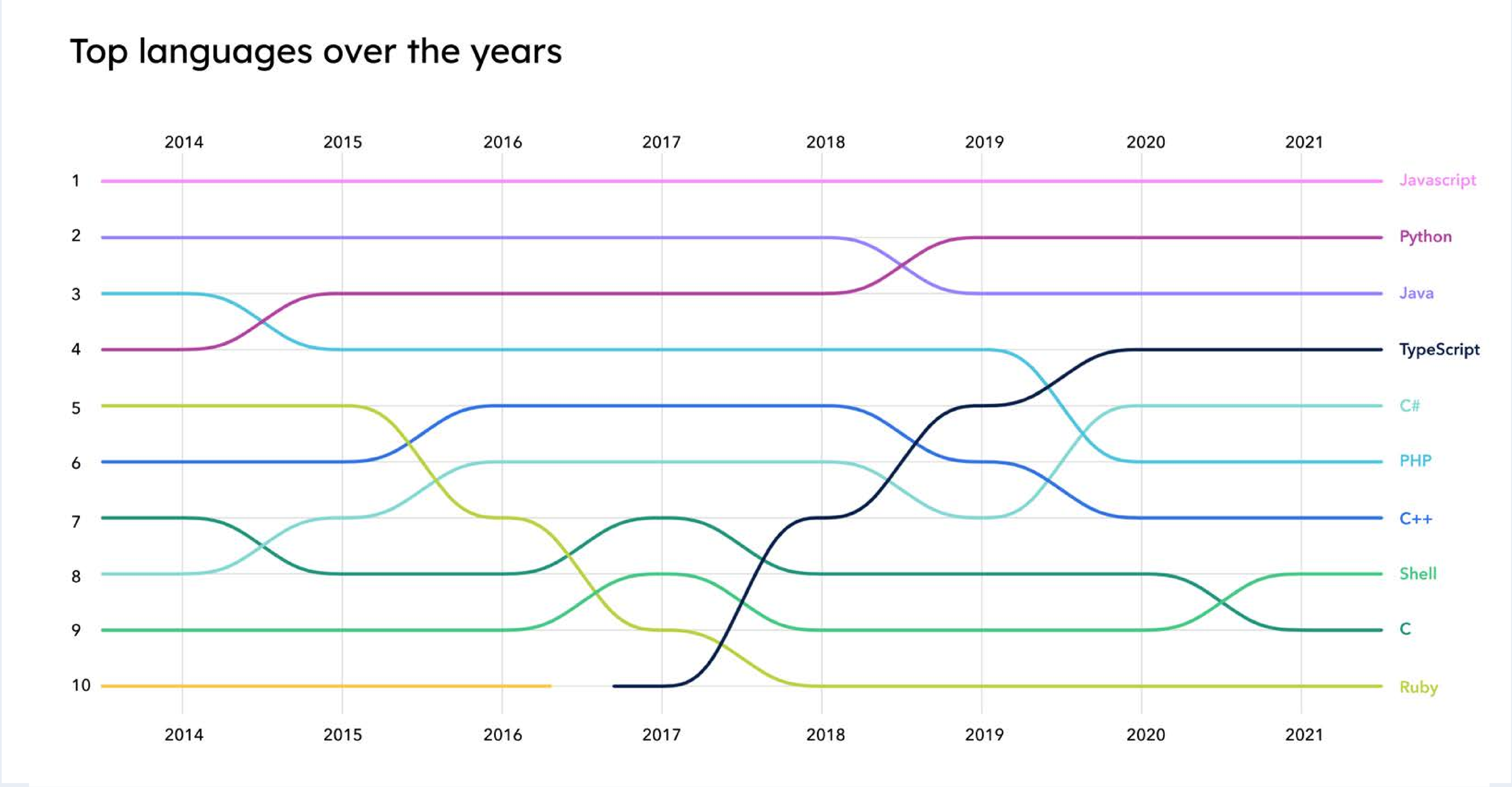
Sign Up

Table of Contents

Why So Many Programming Languages?

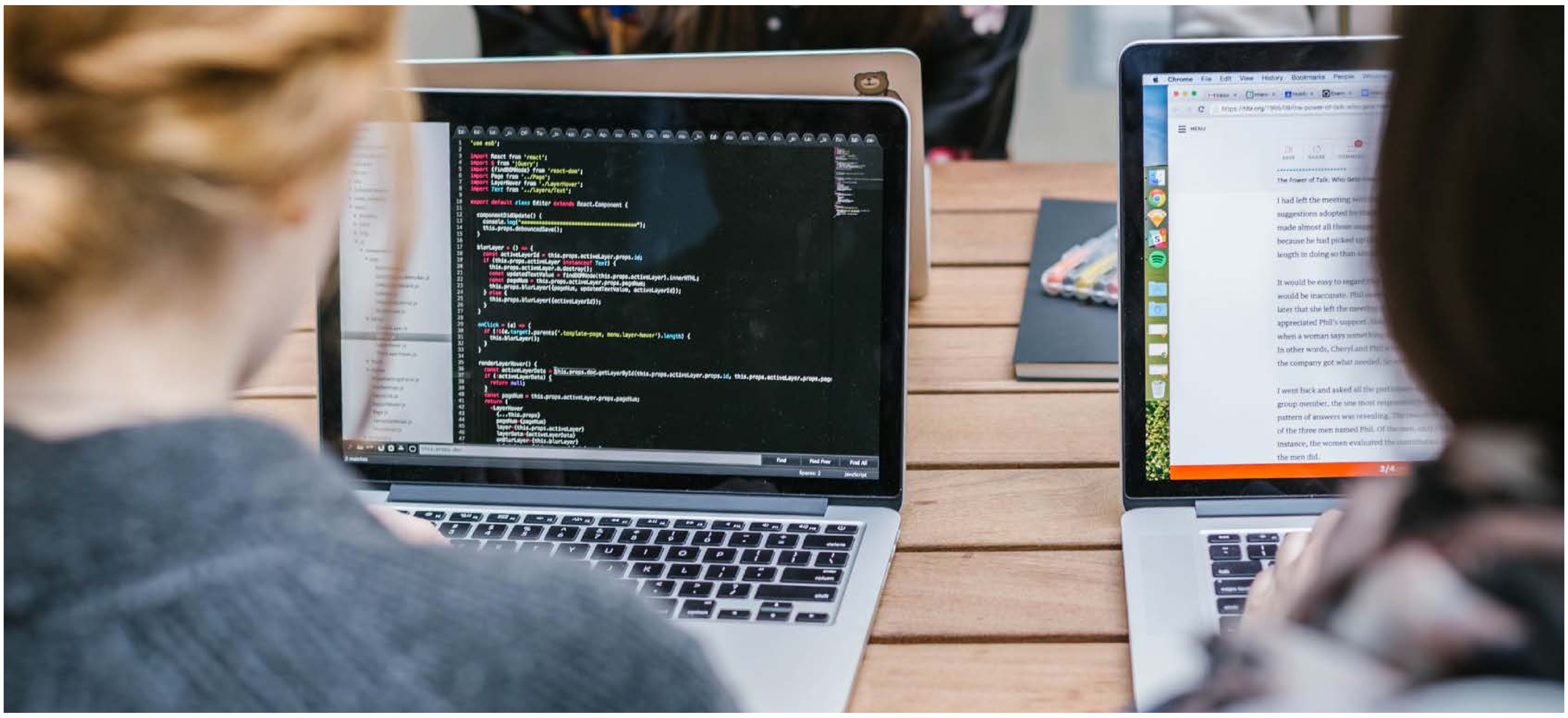
There are so many programming languages out there, and more are developed every few years. To name a few, we have Python, JavaScript, Java, PHP, C++, Ruby, C#, and so many more.

According to Github, the top three most popular programming languages are JavaScript, Python, and Java. So why are there so many of them?



Source

Perhaps the main reason we have so many languages is that technology is always evolving. As new technologies are invented and evolve, we often need new tools that can make software for these technologies. It may be overwhelming to learn that there are so many programming languages. The good news is that if you learn one — say for instance you learn JavaScript — others become easier to learn. Every language like Java or Python that you learn, makes the next language easier. The primary reason for this is that outside of syntax and best practices, many of the essential programming concepts are universal.



The programming languages we currently use might have the capability to deal with a new and specific problem that arises. Each language is usually introduced to solve a particular problem that the industry faces.

For example, the programming language C was created by Dennis Ritchie and Bell Labs back in 1972. It was created to overcome some of the issues of older languages. C++, for instance, was created a decade later in 1985 by Bjarne Stroustrup in order to upgrade certain features in C and to make it more accessible for people to use across multiple industries.

Put simply, different kinds of development require different languages to meet the needs. Some development tasks require focus on speed, others can sacrifice speed over scalability. While others prioritize performance over simplicity. Furthermore, some will want to focus on either mobile development or desktop applications. Different problems often require different sets of tools to solve them.

Each software team requires a different set of expertise. For example, Netflix uses Java to build the backend that does the heavy lifting tasks. They use Python for automation, AI, and machine learning — this is what gives users suggestions to watch. They also use JavaScript to render the videos, browsing interface, search interface, and user interactions. **When deciding to learn a programming language, remember that you don't need to learn all the languages** — you just need to learn one of them and you can analogically figure out what code bases built with other languages are doing. If you are coding with a team, you will likely work with one that is diverse in their programming knowledge.

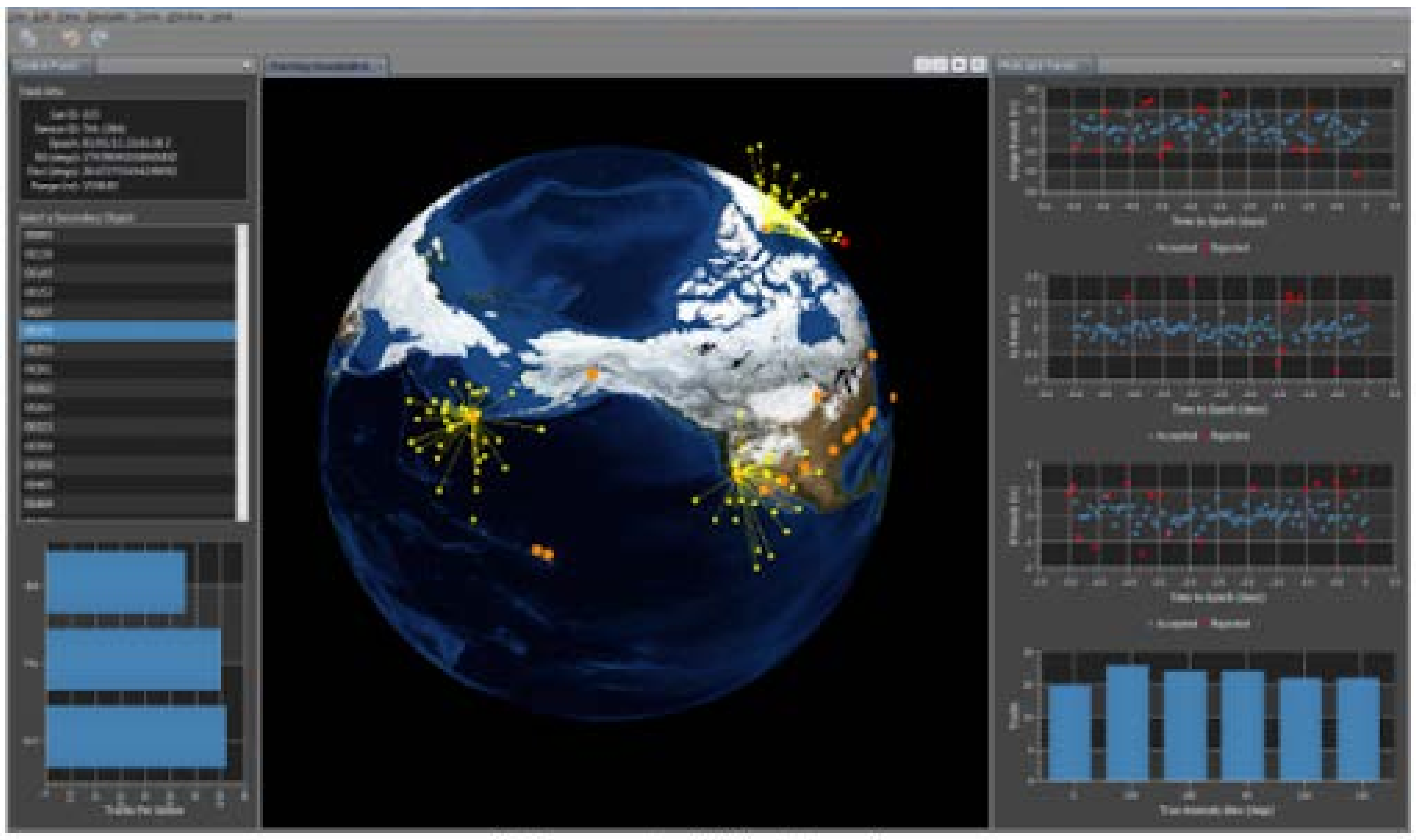
What is Java?

Java was created at Sun Microsystems, Inc. by James Gosling and a team of researchers in an effort to create a new language that would allow electronic devices to communicate with each other. It was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The work on the language began in 1991 and about four years later, the focus then changed to the World Wide Web, and in 1995, the first Java was released.

What made Java so revolutionary at the time was its ability to be multi-platform. This means that the Java code can be written the same way for many platforms, which helped lead to its popularity for use on the Internet, where many different types of computers may retrieve the same web page.

By the late 1990s, Java went from bringing multimedia to the Internet to growing beyond the web where it was powering consumer devices like cell phones, financial computing, and even NASA's Mars exploration rover computing (Also see — NASA's open-source code software projects)

Java became extremely popular in many sectors of technology. In 2010, Oracle Corporation acquired Sun Microsystems and took over the management of Java.



NetBeans running NASA mission software

Source

"Java is the glue that ties our legacy software capabilities together"
- NASA Software Engineer

The difference between the way Java and other programming languages work is that the code in other languages is first translated by a compiler into instructions for a specific type of computer. The Java compiler is unique because it turns code into something called Bytecode, which is interpreted by a software called Java Runtime Environment (JRE), and the Java Virtual Machine(JVM). The JRE acts as the middleman between the code and any host computer. This is how Java can run the same way on many platforms.

Java also follows the Object Oriented Programming (OOP) paradigm. OOP is a fundamental paradigm and it is very popular — nearly every developer at some point in their career has used it. The basic idea behind OOP is that it relies on the concept of classes and objects. It is used to structure a software program into simple, reusable pieces of code blueprints, which are used to create individual instances of objects. This makes development efficient and less vulnerable to errors.

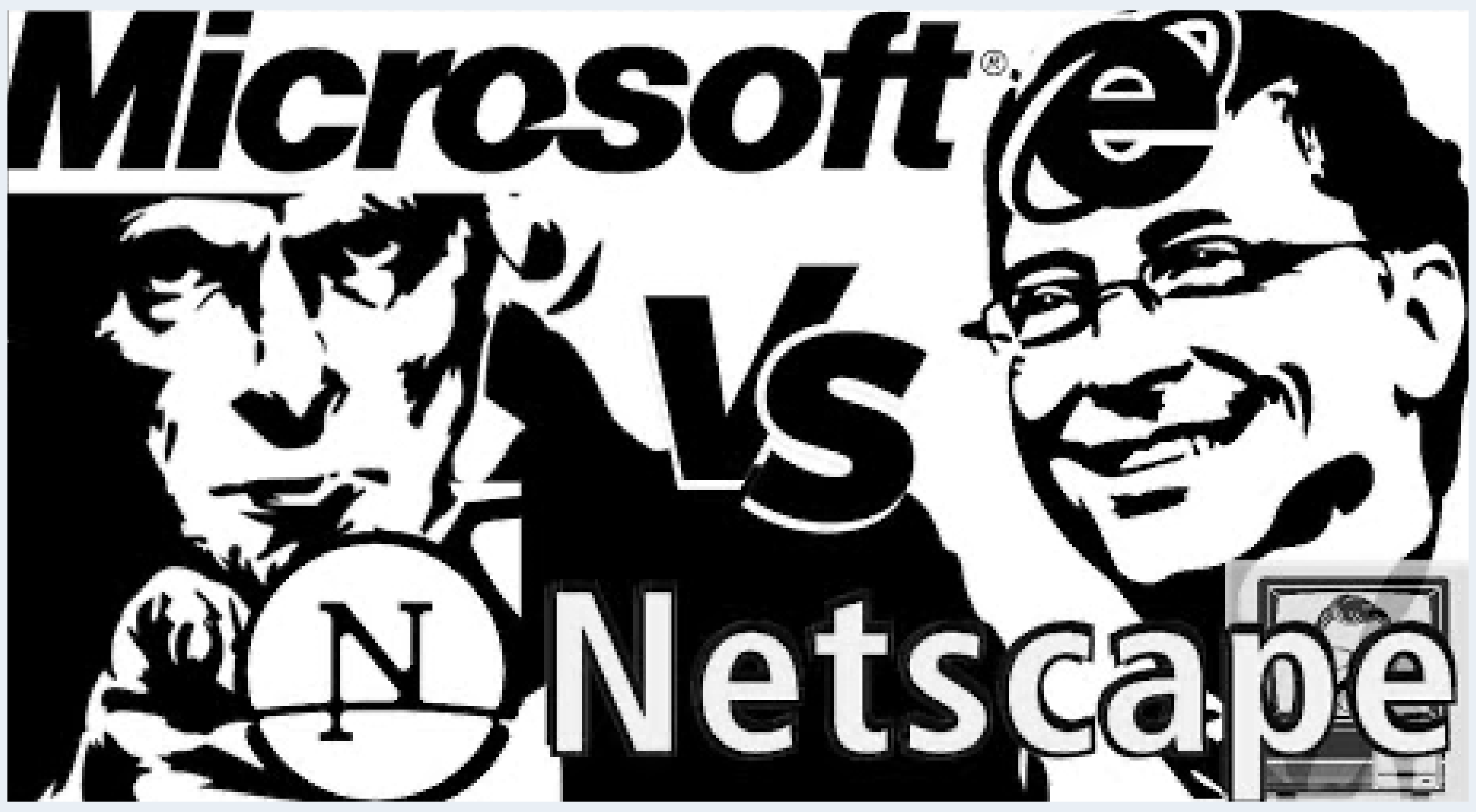
Fun fact: *During development, the team had a hard time naming Java. The initial suggested words were, "dynamic," "revolutionary," "Silk," "DNA," etc. They wanted something that reflected the essence of technology and its revolutionary nature.*

One day James Gosling was enjoying a cup of coffee nearby his office and decided to name the language Java, after the Indonesian island where the first coffee was produced.

What is JavaScript?

JavaScript is quickly becoming the world's most popular programming language. Virtually every personal computer in the world has JavaScript interpreters installed on it and in active use. In the year 1995, Brendan Eich was tasked with writing a new language in ten days!

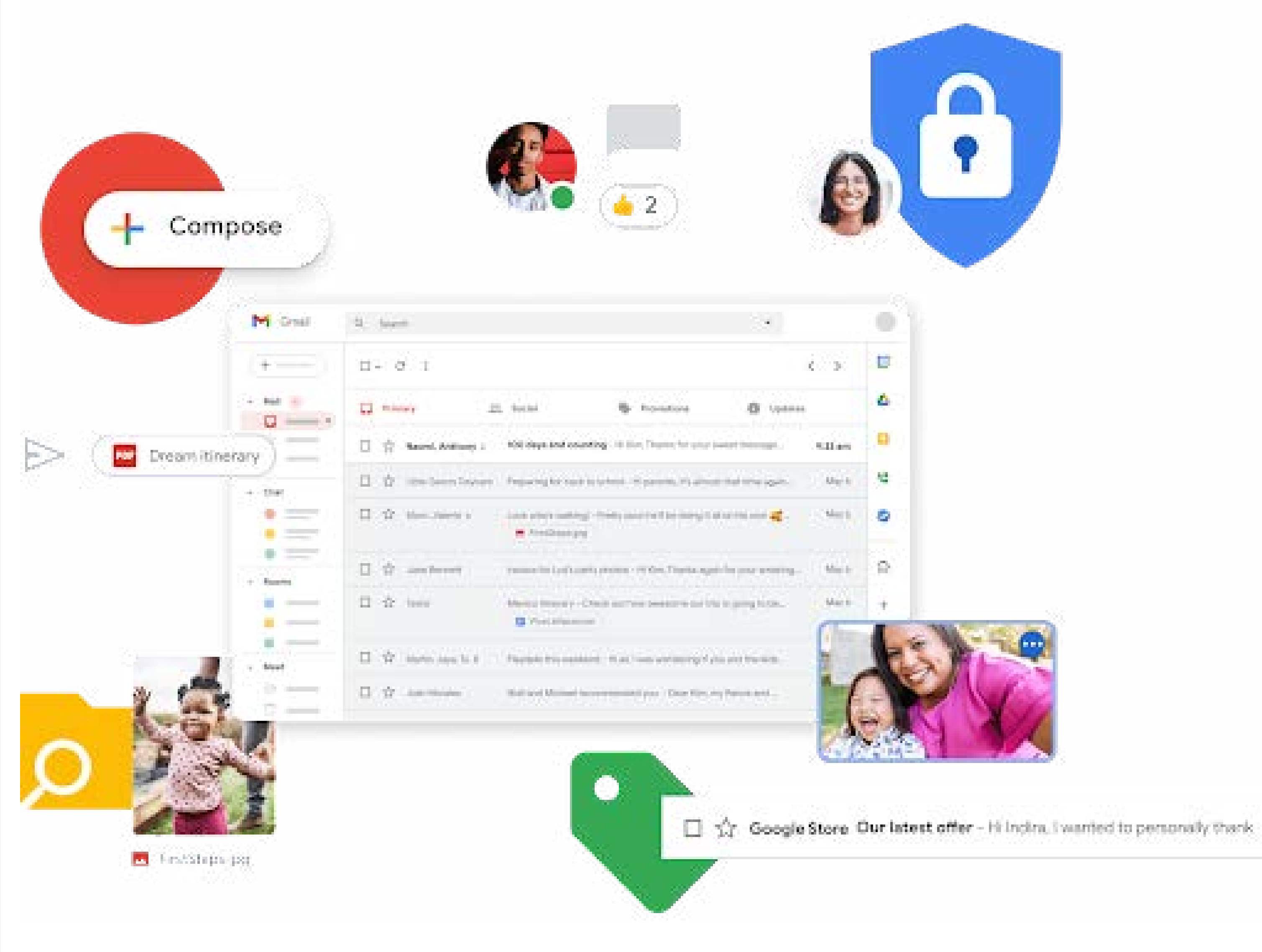
During the browser wars of the 1990s, Netscape Communications Corp., originally known as Mosaic Communications Corp., planned on capitalizing on the emerging World Wide Web. Their first web browser was the Mosaic Netscape 0.9, followed by the Netscape Navigator. In trying to compete with Microsoft's Internet Explorer, Netscape wanted to make something unique and that could be the gateway through which people use the internet.



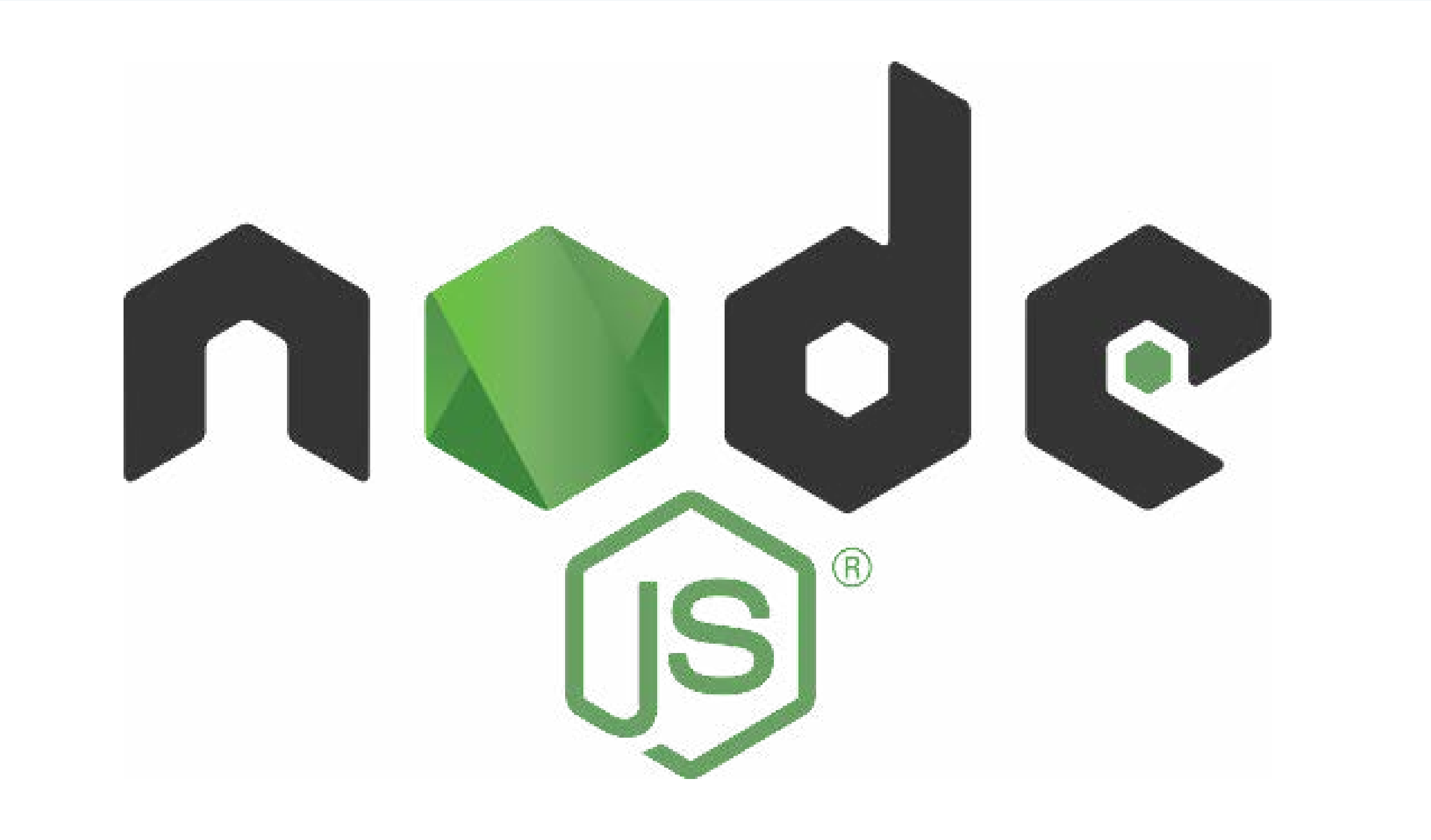
The browser wars initiated a need for something that could drive web-based development and open the flood gates of creativity on the internet. Although at this time, Java was the most popular programming language, developers needed and wanted something lightweight and something that can be interpreted instead of being compiled. This way it would be appealing to even the non-professional programmers.

Netscape realized that they might be losing the browser wars because Microsoft was bigger and had more of a market share. So right before the release of the Netscape Navigator 2.0, Netscape tasked Brendan Eich to write up a language in ten days as a last resort to compete with Microsoft. On November 30th, 1995, JavaScript was released along with the Netscape Navigator 2.0. Companies like America Online, AT&T, and twenty-eight other companies agreed to use it as an open standard language.

As JavaScrip's popularity increased, anyone who wanted to build a browser application and wanted interactive experiences, had to use or learn JavaScript. By the 2000s, JavaScript permeated every aspect of people's lives on the internet.


Compose
Dream itinerary
2
Google Store Our latest offer - Hi Indira, I wanted to personally thank

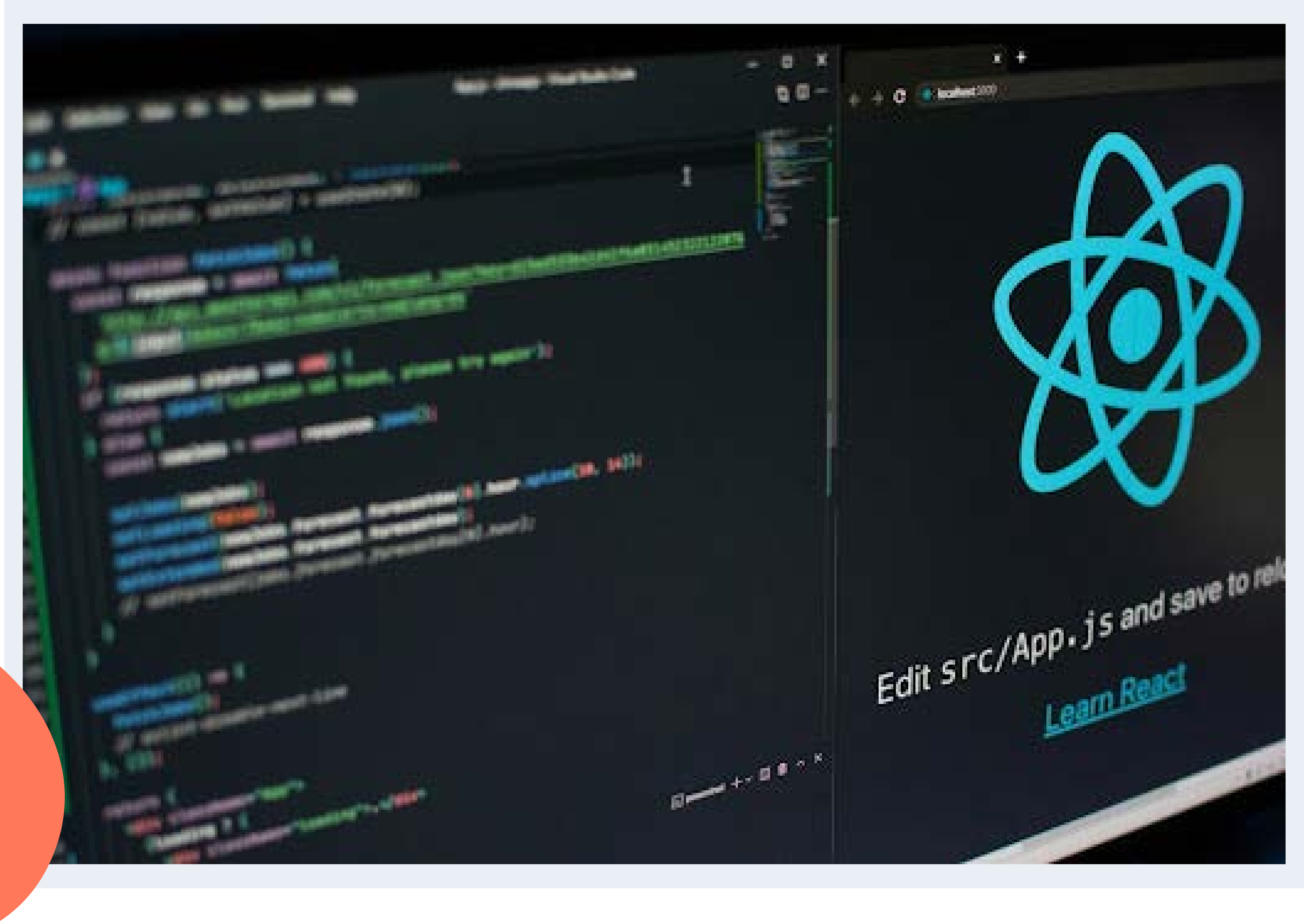
Source



In the early 2000s, Google Mail (Gmail) was released. Gmail was the first application that took JavaScript to the next level. It used it to dynamically change content that you were looking at in the browser. So instead of waiting for a whole new page to load, when you click on an email in your inbox, Google uses JavaScript in the background to fetch and display the content you want to view. No more waiting on a brand new page to load. This changed everything. It saved users time and energy and ushered in the Web 2.0 revolution.

One of the best features of JavaScript is that not only is it faster to learn and implement, but anyone with a web browser suddenly has a development environment right in front of them. Then in 2010, Ryan Dahl introduced Node.js — which is a runtime environment that executes JavaScript code outside the browser. Since then, developers have been able to use JS for writing servers, talking to databases, and creating large, scalable applications.

Fun Fact: *JavaScript had several names through the years. We had JavaScript, Mocha, LiveScript, JScript, and then later as the language grew in recognition, several companies got together and created [ECMAScript](). The European Computer Manufacturers Association decided to standardize and maintain JavaScript. The word "Java" in JavaScript was a marketing ploy to take advantage of the rising popularity of Java at the time.*



Another fantastic feature of JavaScript is that it has the ability to be used in multiple types of programming paradigms. In programming, there are two main categories for programming paradigms: imperative (also called procedural) and declarative (also called functional). Object oriented languages are usually very procedural — procedural is a top down programming approach that gives attention to the details and the flow of the code. Functional programming languages are usually more declarative — that is they focus more on logic than the control flow. JavaScript supports both procedural object-oriented programming and functional programming. This is what makes it a multi-paradigm language.

Java vs JavaScript

Features of Java

- Write code once and run it on almost any computing platform.
- It is designed for building object oriented applications.
- It is a multi-threaded language with automatic memory management which means it can run many parallel programs.
- Facilitates-distributed computing like when companies link different computer systems over the same network as it is network-centric. For instance, your Netflix account on the browser is linked to the Netflix app on your smartphone.

  Note: This feature is not unique to Java

Features of JavaScript

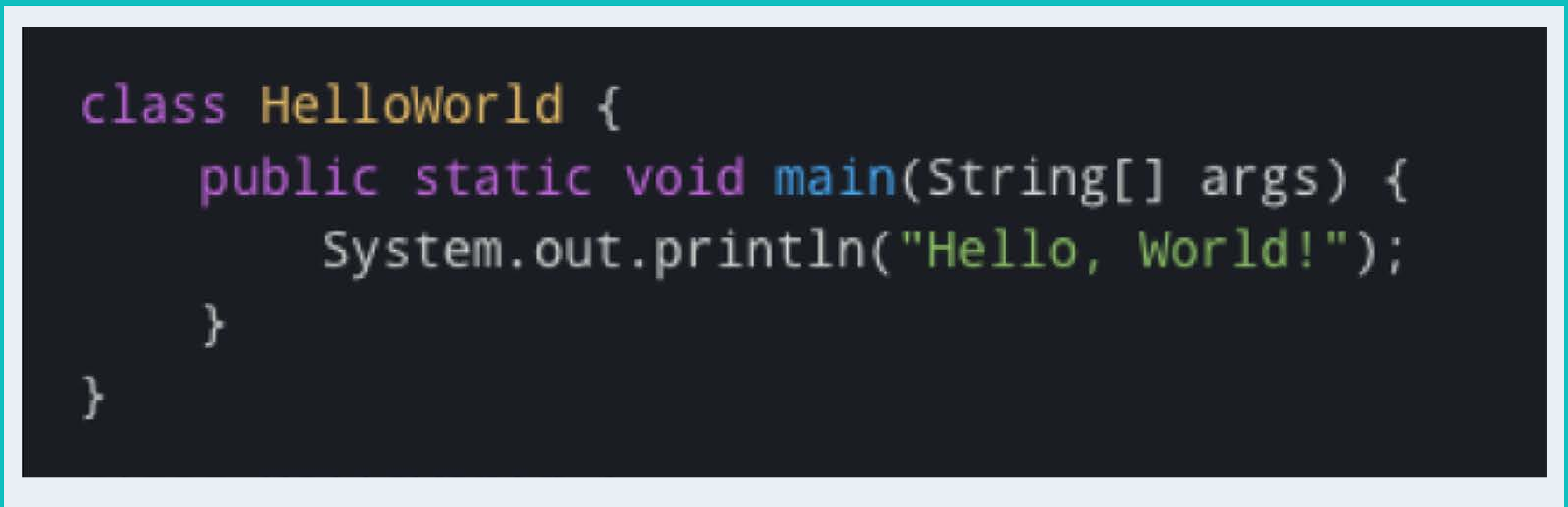
- It is a cross-platform language so there's efficiency not only in writing code but also using it across many different browsers or smartphones.
- It is widely used for client-side (front-end) and or server-side(back-end) as well as (Full Stack). This means you can build an entire application with just JavaScript and not have to worry about learning other languages.
- Easy to learn and start coding as a first language.

Syntax

Let's take a look at the differences in syntax. Note that both languages are heavily influenced by the C programming language's syntax.

In order to run Java on your computer, you can simply download it and follow the Get Started tutorial provided by Oracle. But if you want to explore how Java runs without downloading it, you can use an online Java compiler to experiment with coding in Java. Click "RUN" to see your output.

This is a program that prints "Hello, World!" In Java.

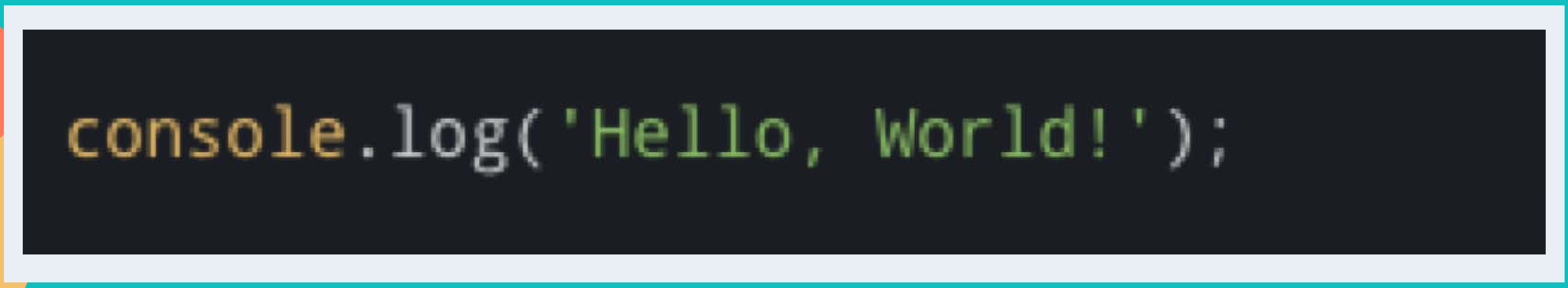


```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

In order to run JavaScript, you can simply use your Google Chrome browser to write code and execute it. You can also use the same online compiler you used for Java, and pick JavaScript as the preferred language to compile JavaScript code.

If you want to execute JavaScript code on your computer, you can download Node.js and follow the Get Started tutorial to run Node.js scripts in your command line interface.

This is a program that prints “Hello, World!” In JavaScript



```
console.log('Hello, World!');
```

Notice the differences in syntax: In Java, you have to build a class which then is tasked with the function of printing out the statement. In JavaScript, you have a built-in function that handles the printing for you.

Perhaps the main difference between the syntax is that — with Java — you have to be very specific about the type of variable you are using. For instance, “Hello, World!” is a type of string, and you must specify that in Java. On the other hand, JavaScript is dynamic, meaning you don’t need to be that specific. At run time, the interpreter will handle any nuances for the program. The advantage of dynamic type-checking is programmer productivity — you write less code and are free to focus on other tasks. The advantage of static type-checking is that errors are caught early in the development process and it executes the code much faster while using less memory.

Which One is Right For You?

As we talked about in the earlier section, there are so many programming languages, and they all solve different problems. It all boils down to what you are trying to build and what resources you have at your disposal.

Both Java and JavaScript can be used to build a variety of programs and products. However, since it's so popular, JavaScript has many resources available, alongside a vast community of developers that can help you along your journey.

You should consider Java if you are interested in:

- **Android apps.**
- **Enterprise software.**
- **Scientific computing.**
- **Big data analytics.**
- **General purpose hardware programming.**
- **Server-side technologies** like GlassFish, JBoss, etc.
- **Desktop applications.**

You should consider JavaScript if you are interested in:

- **A language that is easy to learn and understand**
- **Dynamic Single Page Applications (SPAs)** like Facebook, Instagram, Reddit, Airbnb, Google Maps, etc.
- **Front-end technologies** like ReactJS, AngularJS, etc.
- **Server-side technologies** like Node.js, Express.js, etc.
- **Mobile app development** like React Native
- **Games.**
- **Large online community that supports beginners**

Both Java and JavaScript are high in demand and worth learning. If you have never programmed before, you might find it easier to start with JavaScript. This is because JavaScript has a simpler syntax than Java for a beginner and it is more forgiving than Java while you are learning.

Additional Resources Learning

Want to take your learning to the next level? Whether you want to make a career change into tech, or advance in your current career, the ability to understand and use JavaScript is relevant in many tech careers.

To explore more about coding, programming, and tech skills, sign up for a free workshop to have a first-hand experience.

If you know that coding is your next career move and still have some questions, sign up for a free info-session and talk to the General Assembly team about the course goals, content, instructors, coaches, student resources, job titles, and more.

For a full description of courses on coding see the full-time Software Engineering Immersive from General Assembly, or gear up with a part-time course in Front-End Web Development or JavaScript Development. Schedule a call with the General Assembly Admissions team to get answers to your questions.

People like Lyn Muldrow know the value of leveraging JavaScript skills to break into a tech career. Previously a stay-at-home mom, Lyn became a Hack the Hood technical programs manager then a software engineer at LinkedIn through GA's Software Engineering Immersive program.

> *"GA gave me the foundational knowledge and confidence to pursue my career goals. With caring teachers, a supportive community, and up-to-date, challenging curriculum, I felt prepared and motivated to build and improve tech for the next generation!"*

Glossary

Object Oriented Programming: OOP is a programming paradigm that relies on the concept of classes and objects. It is used to structure a software program into simple, reusable code blueprints (usually called classes), which are used to create individual instances of objects. OOP languages include JavaScript, C++, Java, and Python.

Multi-paradigm Programming Language: In programming, a multi-paradigm programming language is one that supports more than one paradigm. The design goal of such languages is to allow programmers to use the best tool for a job, admitting that no one paradigm solves all problems in the easiest, or most efficient way.

Multi-platform Language: As technology evolves, we need languages that allow us to build the same applications on multiple platforms. For example, we need an app that can be downloaded on the Android platform, IOS, Windows, and the Wed. This makes development much faster and easier, this also reduces development costs.

Web 2.0: Refers to a website that allows users to interact and collaborate with each other through social media dialogue as creators of user-generated content in a virtual community. It was substantially different from previous versions of the web because it created a culture where end users are participating in the web instead of just consuming it.

Client-side Development: is a type of development that involves programs that run on the client or user's device. The developers focus on creating the part of the website that the user can interact with. This is also sometimes called front-end development.

Server-side Development: is a type of development that involves programs that run on a server. The developers focus on behind-the-scenes development that has to do with databases and routing. This is also sometimes called back-end development.

Full Stack Development: refers to the development of both client-side and server-side portions of web applications and websites. Full stack developers have the ability to design complete web applications as they work on the front-end, back-end, database, deployment, and debugging of applications.

Runtime environment (RTE): is an environment in which a program or application is executed. In this environment, the program can send instructions to the computer's processor, access the computer's memory (RAM), and other system resources.

Conclusion

Now you can tell the difference between Java and JavaScript and identify the use cases for each language and how to use them in your day-to-day.

When you're deciding which language to learn, or which one to learn first, use this guide in parallel to other learning experiences. Courses and workshops can give you hands-on practice in your journey.

These coding languages have helped change our way of life, and mastering them can also help you shape and change your career in the world of software engineering.

Shahzad Khan is a Software Engineering Instructor, member of GA's Product Advisory Board, serves on the Board of Directors at Sympara and owner of Frame of Mind, a technical product consulting and engineering services company based in Austin, Texas. He is passionate about philosophy, technology education, accessibility, data-driven solutions, and product management. Shahzad has been a part of GA's ecosystem for over three years and has led instruction for both consumer and enterprise cohorts.

Read more about Shahzad Khan's experience here