



Encryption

Juka can encrypt and decrypt files. It can also run encrypted Juka script files.

To encrypt a file "HelloWorld.juk", run the following in Console:

```
./juka --encrypt HelloWorld.juk
```

This will create two files:

- HelloWorld.juk.encrypt
- HelloWorld.juk.key

They HelloWorld.juk.encrypt contains the encrypted file. The HelloWorld.juk.key contains AES key for use to decrypt the file.

To decrypt the file back to "HelloWorld.juk" run the following command:

```
./juka --decrypt HelloWorld.juk
```

This will take "HelloWorld.juk.encrypt" and "HelloWorld.juk.key" and decrypt the file. NOTE: You can encrypt/decrypt any file (not just Juka scripts)

To run the encrypted files without having to decrypt, run the following:

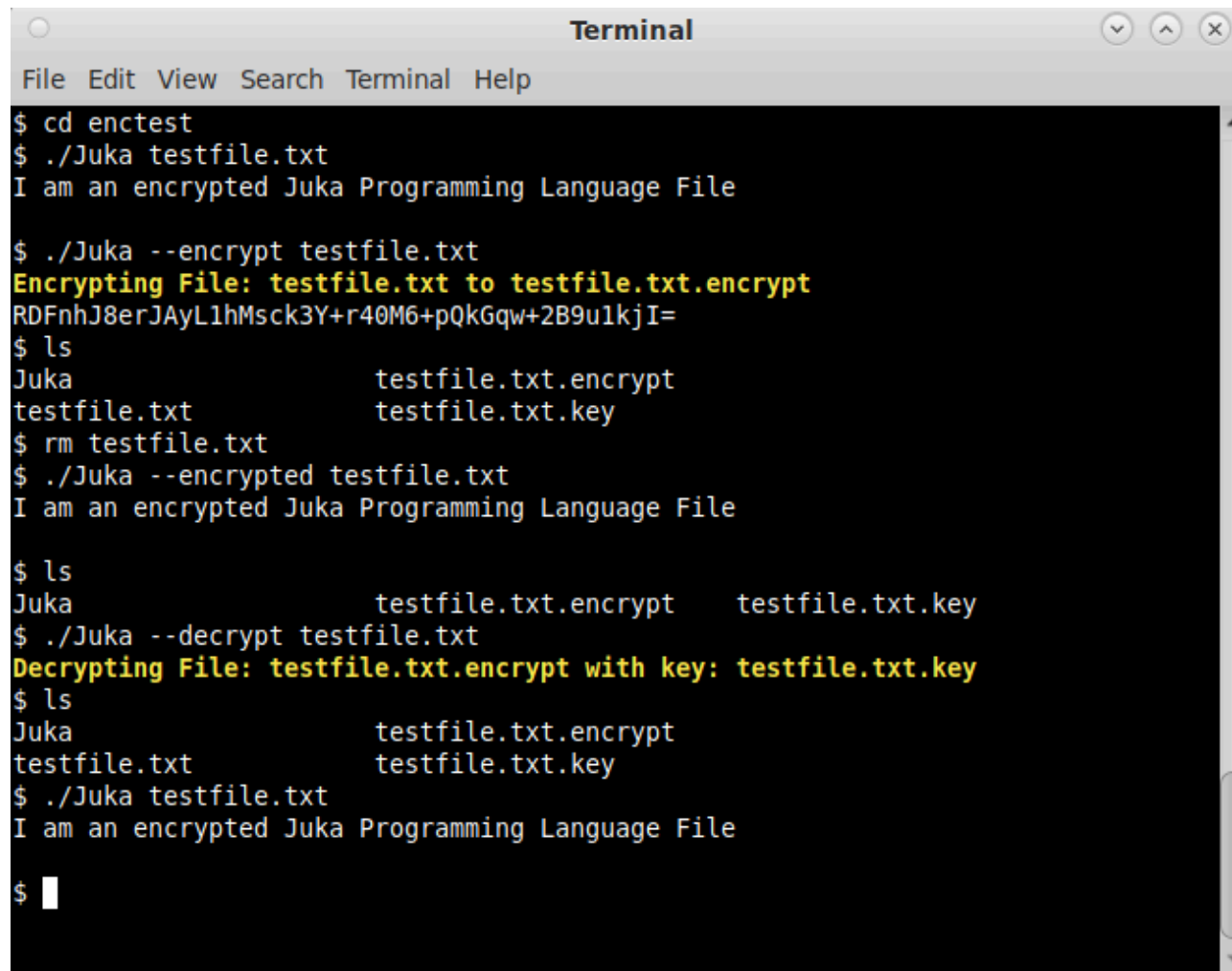
```
./juka --encrypted HelloWorld.juk
```

This will use the following files:

- HelloWorld.juk.encrypt

- HelloWorld.juk.key

decrypt them in memory and execute them in Juka.



```
Terminal
File Edit View Search Terminal Help
$ cd enctest
$ ./Juka testfile.txt
I am an encrypted Juka Programming Language File

$ ./Juka --encrypt testfile.txt
Encrypting File: testfile.txt to testfile.txt.encrypt
RDFnhJ8erJAYLlhMsck3Y+r40M6+pQkGqw+2B9u1kjI=
$ ls
Juka                testfile.txt.encrypt
testfile.txt        testfile.txt.key
$ rm testfile.txt
$ ./Juka --encrypted testfile.txt
I am an encrypted Juka Programming Language File

$ ls
Juka                testfile.txt.encrypt  testfile.txt.key
$ ./Juka --decrypt testfile.txt
Decrypting File: testfile.txt.encrypt with key: testfile.txt.key
$ ls
Juka                testfile.txt.encrypt
testfile.txt        testfile.txt.key
$ ./Juka testfile.txt
I am an encrypted Juka Programming Language File
$
```

Fundamentals of Juka Programming Language

Juka is a general-purpose programming language that is designed to be simple, efficient, and expressive. It is a statically typed language, which means that the types of variables and expressions are checked at compile time. Juka is also a compiled language, which means that it is converted to machine code before it is executed.

Syntax

The syntax of Juka is similar to that of other programming languages, such as Java and C#. The basic building blocks of Juka code are expressions, statements, and functions.

- **Expressions** are used to evaluate values. They can be composed of variables, literals, operators, and functions.
- **Statements** are used to control the flow of execution. They can be used to assign values to variables, call functions, and make decisions.
- **Functions** are reusable blocks of code. They can be called from other functions or from the main program.

Offline Documentation

There are two ways to access Juka documentation offline:

- **PDF:** You can download the Juka documentation in PDF format by going to the **Downloads** page.
- **Cached Juka website (Juka PWA):** You can also create a cached version of the Juka website that can be accessed offline. To do this, follow these steps:
 - i. On a mobile device, click on the "add this to homepage" button. This will automatically create a PWA app that can be accessed even without the internet.
 - ii. On a desktop computer, find the browser URL field and look at the right section. Click on the "Install Juka Programming Language Website" button. This will install a browser app that can be ran offline (without internet).

Once you have created a cached version of the Juka website, you can access it offline by opening the PWA app or the browser app.

Downloading Juka

There are two ways to download Juka:

- **From the official website:** You can download the latest version of Juka from the **official website**.
- **From GitHub:** You can also download the latest binaries from the **GitHub repository**.

Which version should you download?

There are typically three versions of Juka available for each desktop operating system:

- **JukaApp:** An app version of Juka with a graphical user interface (GUI). It provides the simplest way to run Juka.
- **Juka:** This is the main console application. If you are a more experienced user, it is recommended that you download this version.
- **JukaAPI:** A server to serve the Juka REST API. This is typically used for server-side development.

Are the files self-contained?

Yes, the files are self-contained. This means that you do not need to download any other files or install any other programs to run Juka.

Writing your first application

Let's start by creating a "Hello World" application.

1. Create an empty file and save it as `HelloWorld.juk`.
2. In that file, copy and paste the following code:

```
func main() = {  
    printLine("Hello World")  
}
```

3. Once done, save the file and run the following command:

```
./juka HelloWorld.juk
```

This will compile and run your application. If everything went well, you should see the following output:

```
Hello World
```

Variables

What is a variable?

A variable is a named location in memory that can be used to store a value. Variables can be used to store any type of data, including numbers, strings, objects, and arrays.

Defining Variables

Variables are defined using the `var` keyword. The syntax for defining a variable is as follows:

```
<name> = <value>
```

For example, the following code defines a variable named `x` and sets it to the value 3:

```
x = 3
```

Reassigning Variables

The value of a variable can be changed by reassigning it. To reassign a variable, simply use the `var` keyword again, followed by the new value. For example, the following code reassigns the value of the variable `x` to 4:

```
x = 3  
x = 4
```

Deleting Variables

Variables can be deleted using the delete keyword. The syntax for deleting a variable is as follows:

```
<name> = null
```

For example, the following code deletes the variable `x`:

```
x = null
```

Common Errors

There are a few common errors that can be made when working with variables. One common error is assigning a value to a variable that is not the correct type. For example, the following code will generate an error because the value "3" is a string, but the variable `x` is an integer:

```
x = "3"
```


Strings

What is a string?

A string is a sequence of characters. In programming, strings are often used to represent text.

Defining Strings

Strings are defined using quotation marks. For example, the following code defines a string named `animal` and sets it to the value "cat":

```
animal = "cat"
```

Combining strings with a number

Whenever you add a string to a number (or vice versa), the end result will be a string. The numbers are implicitly converted into strings. For example, the following code will print the string "123abc":

```
"123" + "abc"
```

Conditions

What is a condition?

A condition is a statement that can be evaluated to be true or false. Conditions are used to control the flow of execution of a program.

Syntax

The syntax for a condition is as follows:

```
if <condition>
{
<statements>
}
else
{
<statements>
}
```

For example, the following code will print "Correct!" if the value of the variable sum is equal to 5, and "Incorrect!" otherwise:

```
sum = 4+1
if sum == 5{
    "Correct! 4+1 = 5"
}
else
{
    "Incorrect!"
}
```

Classes

What is a class?

A class is a blueprint for creating objects. It defines the properties and methods that objects of that class will have.

Coming Soon

Installing and Running Juka (Console)

Juka can be run on many platforms, including:

- Windows
- macOS
- Linux
- FreeBSD
- Web browsers

Installing Juka

To install Juka, follow these steps:

1. Go to the **Juka website**.
2. Click the **Download** button.
3. Select the platform you want to install Juka on.
4. Follow the instructions on the screen to install Juka.

Running Juka

Once Juka is installed, you can run it by following these steps:

1. Open a terminal window.
2. Navigate to the directory where Juka is installed.
3. Type the following command and press Enter:

Installing and Running Juka (Console)

Juka can be run on many platforms, including:

- Windows
- macOS
- Linux

Installing Juka

To install Juka, follow these steps:

1. Go to the **Juka website**.
2. Click the **Download** button.
3. Select the platform you want to install Juka on.
4. Follow the instructions on the screen to install Juka.

Running Juka

Once Juka is installed, you can run it by following these steps:

1. Open a terminal window.
2. Navigate to the directory where Juka is installed.
3. Type the following command and press Enter:

```
./juka
```

This will start the Juka interpreter. You can then type Juka code and press Enter to execute it.

To run Juka code from a file, open a terminal window and navigate to the directory where the file is located. Then, type the following command and press Enter:

```
./juka <filename>
```

For example, to run the code in the file `hello_world.juk`, you would type the following command:

```
./juka hello_world.juk
```

This will execute the Juka code in the file, and print the output to the console.

FreeBSD

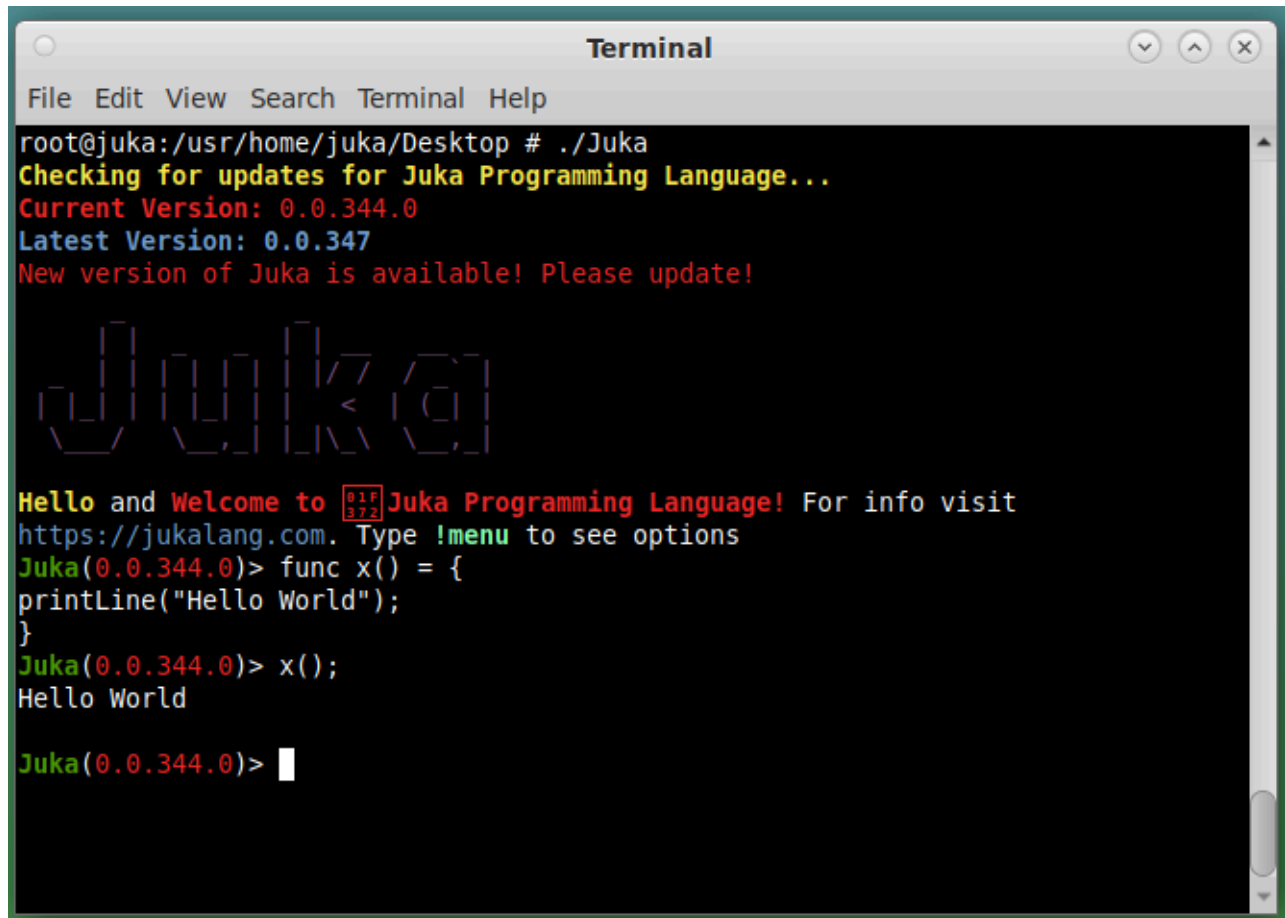
To install Juka on FreeBSD, run the following command:

```
pkg install juka
```

or download it directly from <https://jukalang.com/download>

Once Juka is installed, you can start the editor by running the following command:

```
./juka
```



```
Terminal
File Edit View Search Terminal Help
root@juka:/usr/home/juka/Desktop # ./Juka
Checking for updates for Juka Programming Language...
Current Version: 0.0.344.0
Latest Version: 0.0.347
New version of Juka is available! Please update!

Juka

Hello and Welcome to 0.1.13.7.2 Juka Programming Language! For info visit
https://jukalang.com. Type !menu to see options
Juka(0.0.344.0)> func x() = {
printLine("Hello World");
}
Juka(0.0.344.0)> x();
Hello World

Juka(0.0.344.0)> █
```

To run Juka code from a file, run the following command, substituting the name of your file for `my_file.juk`:

```
./juka my_file.juk
```

For example, to run the code in the file `hello_world.juk`, you would run the following command:

```
./juka hello_world.juk
```

This will compile and execute the Juka code in the file, and print the output to the console.

Juka in Jupyter

Juka is a new programming language that is designed to be used for data science and analytics. It is similar to Python, but it is more powerful and efficient.

Jupyter is a popular interactive environment for developing and running code. It can be used with many different programming languages, including Python, R, and Julia.

The Juka team has created a Juka kernel for Jupyter. This kernel allows you to run Juka code in Jupyter notebooks.

To use the Juka kernel, you will need to install Juka and Jupyter. You can find instructions for installing Juka on the [Juka website](#). You can find instructions for installing Jupyter on the [Jupyter website](#).

Once, downloaded add Juka to path.

On Windows, it is recommended that you follow the steps, outlined at <https://stackoverflow.com/questions/44272416/how-to-add-a-folder-to-path-environment-variable-in-windows-10-with-screensho>

On Linux/Unix/MacOS (bash): Assuming that we stored Juka in /usr/bin/share/Juka, we run the following command:

```
$ export path = "/usr/bin/share/Juka:$PATH"
```

This will add Juka to PATH

Once you have installed Juka and Jupyter, you need to install Juka kernel:

Using PIP

```
pip install juka_kernel
```

Using Github

or install it directly from the source by downloading from:
https://github.com/jukaLang/juka_kernel and running:

```
python -m juka_kernel.install
```

in the kernel folder.

Next Steps

You can start using the Juka kernel in Jupyter notebooks. To do this, open a Jupyter notebook and select "New" from the menu. In the "Notebook" dropdown menu, select "Juka".

You can now start writing Juka code in your Jupyter notebook. To run your code, select "Run" from the menu.

The Juka kernel will compile and run your code. The output of your code will be displayed in the notebook.

The Juka kernel is a powerful tool that allows you to use Juka in Jupyter notebooks. This makes it easy to develop and debug Juka code.

Technical Details

The Juka kernel is written in Python and uses the **Jupyter protocol**. The kernel does not compile or run the Juka code. It just serves as a messaging protocol between Jupyter and Juka.

The Juka kernel is open source and is available on **GitHub**.

Contributing

If you would like to contribute to the Juka kernel, please see the **contribution guidelines**.

Unity 3D

Juka can be used in Unity 3D by following these steps:

1. Install the JukaCompiler NuGet package.
2. Create a multi-line string in the engine (using ```) and then run the following:

```
new JukaCompiler.Compiler().Go(sourceAsString, false);
```

Example

The following code will print "Hello World" to the console:

```
string juka = `
"Hello World"
`
var output new JukaCompiler.Compiler().Go(juka, false);
```

Azure Function

Running Juka Locally

Juka can be run locally without an Azure Web server. This is useful for testing Juka before uploading it to a server.

To run Juka locally, follow these steps:

1. Open the Juka.sln project in Visual Studio 2022.
2. Click the **Start AzureJukaFunction** button. This will start an Azure emulator locally.
3. Use Postman to send functions to the Azure emulator. To do this, create a new request and set the request method to **POST**. In the **Body** tab, select **Raw** and paste the following code:

```
{  
  "code": "\"Hello World\""  
}
```

4. Click **Send**. The Juka emulator will run the function and print the message "Hello World" to the console.

Running Juka on Azure

To run Juka on Azure, follow these steps:

1. Upload the Juka package to Azure.
2. Use web deploy to publish Juka to Azure.

3. To call Juka from a client, send a raw request message to the Juka endpoint. The request message should have the following format:

```
{  
  "code": "\"Hello World\""  
}
```

Note: You must escape quotations, otherwise, Juka will return an error.

Juka in a C# Project

Juka can be added to any C# project as a NuGet package. To do this, open the Package Manager Console and run the following command:

```
Install-Package JukaCompiler
```

Once the package is installed, you can use it to compile Juka code from within your C# project. To do this, add the following code to your project:

```
//C# File
using JukaCompiler;

public class Program {
    public static void Main(string[] args) {
        string jukaCode = "func main() { println(\"Hello World\")
}";
        // Compile the Juka code.
        var compiledCode = new
JukaCompiler.Compiler().Go(jukaCode, false);

        // Execute the compiled code.
        var result = compiledCode.Execute();

        // Print the result.
        Console.WriteLine(result);
    }
}
```

This code will compile the Juka code and print the message "Hello World" to the console

Using a file

If you need to compile the code from a file, you can do so by passing the path to the file to the Go method. For example, the following code will compile the Juka code in the file my_code.juka:

```
//C# File
using JukaCompiler;

public class Program {
    public static void Main(string[] args) {
        string jukaCodePath = @"my_code.juka";
        // Compile the Juka code.
        var compiledCode = new
JukaCompiler.Compiler().Go(jukaCodePath, false);

        // Execute the compiled code.
        var result = compiledCode.Execute();

        // Print the result.
        Console.WriteLine(result);
    }
}
```

Output

The output of the Go method is a string. This string contains the compiled Juka code. You can use this string to execute the compiled code or to save it to a file.

For more information visit, <https://www.nuget.org/packages/JukaCompiler>

Installing Juka App

The Juka app is a powerful tool for learning and using the Juka programming language. With the Juka app, you can:

- Write, compile, and run Juka code on your mobile device.
- Learn Juka with interactive tutorials and exercises.
- Get help with Juka syntax and semantics.
- Share your Juka code with others.

The Juka app is available for free on the following mobile platforms:

- Android
- iOS
- Amazon Fire
- Windows

Juka Android App

The Juka Android App is a powerful tool for learning and using the Juka programming language. With the Juka Android App, you can:

- Write, compile, and run Juka code on your Android device.
- Learn Juka with interactive tutorials and exercises.
- Get help with Juka syntax and semantics.
- Share your Juka code with others.

The Juka Android App is available for free on the Google Play Store. To download the app, simply open the Google Play Store app on your Android device and search for "Juka".

More Information Coming Soon

We are currently working on adding new features to the Juka Android App, such as:

- Support for more Juka features, such as functions, classes, and objects.
- The ability to debug Juka code.
- The ability to run Juka code on a remote server.

We will be sharing more information about these features soon. In the meantime, you can learn more about the Juka Android App on our discord channel.

JukaApp for Windows

The Juka Windows app is a powerful tool for learning and using the Juka programming language. With the Juka app, you can:

- Write, compile, and run Juka code on your Windows device.
- Learn Juka with interactive tutorials and exercises.
- Get help with Juka syntax and semantics.
- Share your Juka code with others.

The Juka Windows app is available for free on the [Juka website](#). To download the app, simply visit the website and click the **Download** button.

More Information Coming Soon

We are currently working on adding new features to the Juka Windows app, such as:

- Support for more Juka features, such as functions, classes, and objects.
- The ability to debug Juka code.
- The ability to run Juka code on a remote server.

We will be sharing more information about these features soon. In the meantime, you can learn more about the Juka Windows app on our discord channel.

Running Online

There are three ways to run Juka without installing anything (and using a 3rd party system):

- **IDE**

The fully featured web-based online IDE. You can either run it **online** through a third party system (or host your own version by installing the app)

- **Lite IDE**

The Juka Try Online is a web-based IDE that allows you to write, compile, and run Juka code without installing anything. To use the Juka Try Online, simply visit the **Juka website** and start coding.

- **API**

The Juka API is a RESTful API that allows you to run Juka code on a remote server. To use the Juka API, you will need to create an account and obtain an API key. Once you have an API key, you can use it to run Juka code by making HTTP requests to the Juka API endpoints.

References

- **Juka IDE**
- **Juka LiteIDE**
- **Juka API**

Juka API

The Juka API is a powerful tool for developers who want to use the Juka programming language in their applications. With the Juka API, you can:

- Run Juka code on a remote server.
- Integrate Juka code with your existing applications.
- Build custom Juka applications.

The Juka API is still under development, but it is already available for use. To learn more about the Juka API, visit the [Juka website](#).

Coming Soon

We are currently working on adding new features to the Juka API, such as:

- Support for more Juka features, such as functions, classes, and objects.
- The ability to debug Juka code.
- The ability to use the Juka API in a variety of programming languages.

We will be sharing more information about these features soon. In the meantime, you can learn more about the Juka API on our discord channel.

Introduction to Development

Thank you for taking the time to contribute to the Juka project. We are always looking for help, and we appreciate any assistance that you can provide.

Contributing

There are a few ways that you can contribute to the Juka project:

- **Report bugs:** If you find a bug in the Juka code, please report it on the [Juka issue tracker](#).
- **Contribute code:** If you would like to contribute code to the Juka project, please fork the repository on GitHub and submit a pull request.
- **Write documentation:** The Juka documentation is always under development, and we would love to have your help. Please feel free to edit the documentation on GitHub.
- **Spread the word:** The best way to help the Juka project is to spread the word. Tell your friends and colleagues about Juka, and help us get the word out.

Coding style

Juka is built on top of .NET, so please follow the coding conventions outlined in the [.NET coding conventions](#).

Testing

Testing is an important part of the Juka development process. We encourage you to write tests for your code. Please focus on edge cases when you write tests.

Submitting to GitHub

When submitting code to GitHub, please keep the code clean and easy to read. Follow the KISS principle (Keep It Simple, Stupid). If you want to make a major change, please fork the project, update it, and once you've fully implemented the fix, submit a pull request to the main project.

Thank you!

Thank you for your interest in the Juka project. We appreciate your help in making Juka the best it can be.

Architecture

Folder Structure

The Juka codebase is organized into the following folders:

- **./examples** - This folder contains examples of Juka code.
- **./src/JukaCompiler** - This folder contains the core of the Juka compiler.
- **./src/JukaAzureFunction** - This folder contains the code for running Juka Azure functions on Microsoft's Azure Cloud Server.
- **./src/JukaNetwork** - This folder contains the code for the Juka network, which is a blockchain network for storing and managing packages.
- **./src/JukaUnitTest** - This folder contains unit tests for the Juka compiler.
- **./src/Juka** - This folder contains the code for the Juka GUI (coming soon).

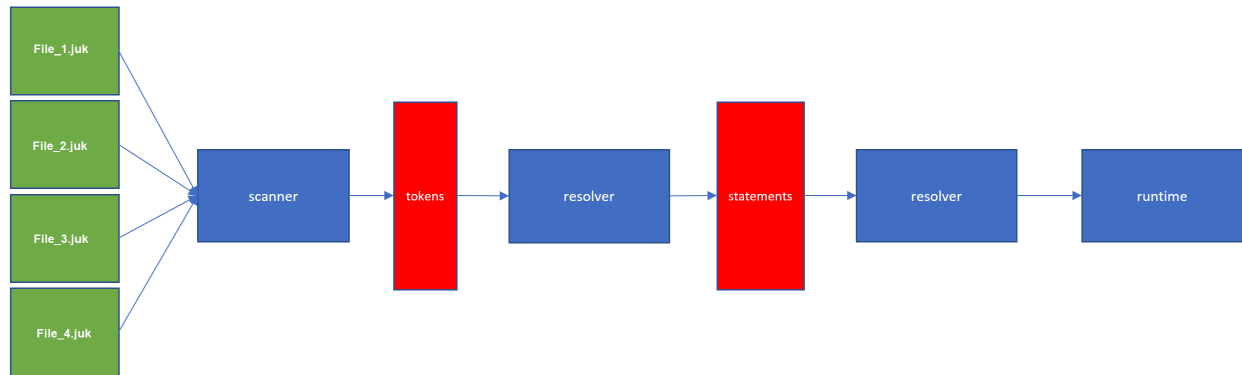
Visual Studio/Development Requirements

To develop and run Juka, you will need the following:

- The latest version of Visual Studio.
- The following Visual Studio packages:
 - Azure development
 - .NET desktop development

Runtime

Juka uses the following scheme to run the code:



The Juka compiler compiles Juka code into a .NET .dll library. This library can then be used in any C# project, including Xamarin/MAUI for building iOS/Android Apps, and it is mainly used to build cross-platform apps for Mac/OS, Windows Apps, and Windows desktop applications.

The Juka Azure Function runtime code is used to run Juka Azure functions on Microsoft's Azure Cloud Server.

The Juka network is a blockchain network for storing and managing packages. It is used as a decentralized package manager.

The Juka unit tests are used to test the Juka compiler.

The Juka GUI is a graphical user interface for Juka that is currently under development.

Notes

- The Juka project is still under development, so some features may not be available yet.
- If you have any questions or feedback, please feel free to contact us.

Visitor Pattern

The visitor pattern is a software design pattern that allows a single operation to be applied to the elements of an object structure without changing the structure itself. It is a way of separating an algorithm from an object structure on which it operates.

Juka is built using the visitor pattern. This means that Juka code can be written in a way that is independent of the specific object structure that it is being applied to. This makes Juka code more reusable and easier to maintain.

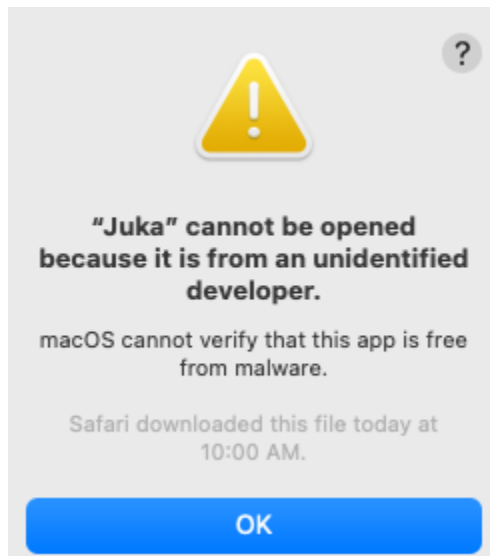
For more information on the visitor pattern, please see the Wikipedia article:

- [**Visitor Pattern**](#)

Troubleshooting Macintosh

Error Message

When you first run Juka on Macintosh, you might get the following error message:



Cannot be opened because it is from an unidentified developer.

Workaround

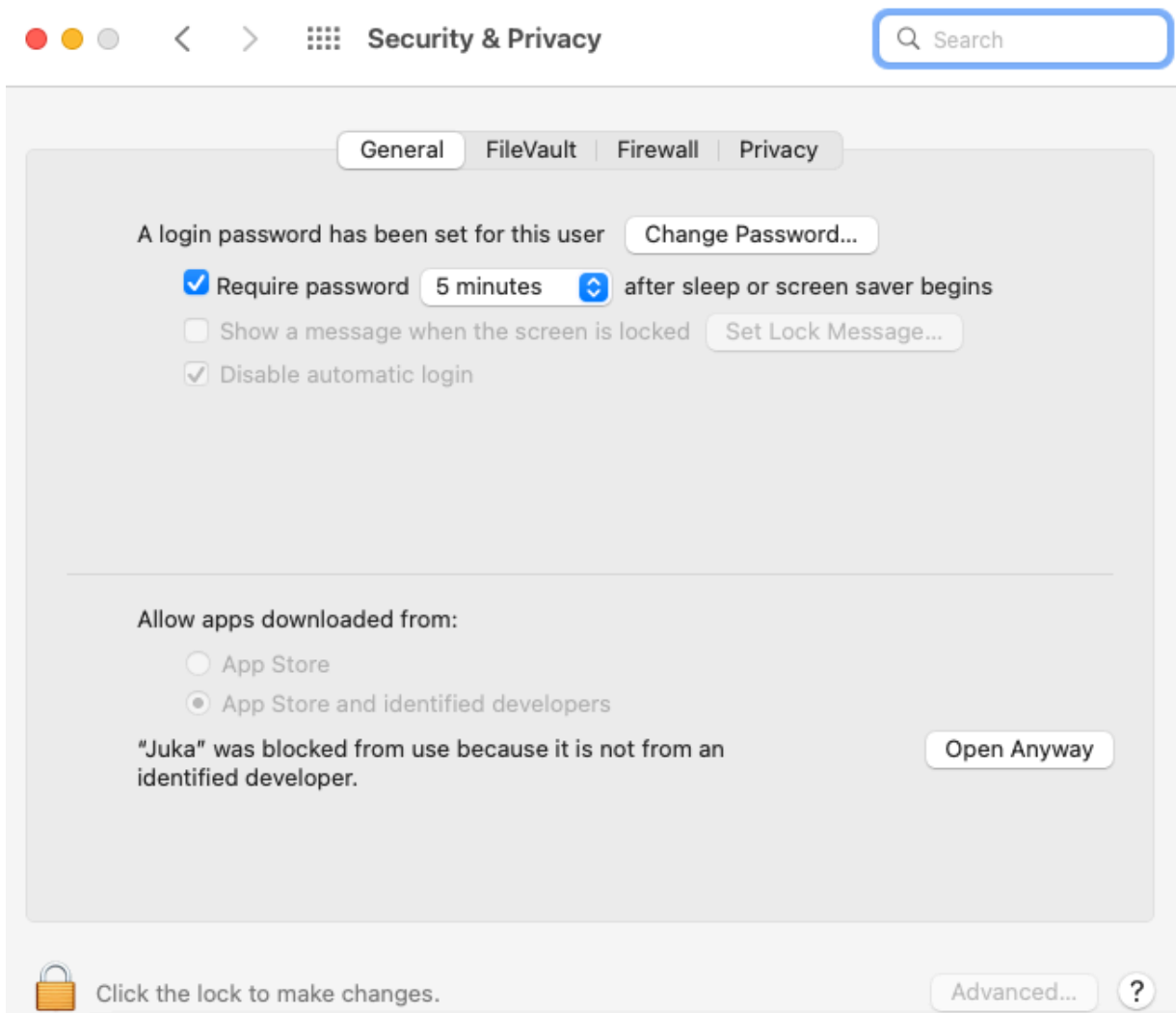
To resolve this issue, follow these steps:

Double-click on the Juka app.

1. If it tries to open the application in the text editor, open up a terminal in

the same folder and run `./Juka`.

2. Click on the Apple icon in the top left corner of your screen and go to System Preferences.
3. Click on Security & Privacy.
4. Click on the General tab.
5. Click on the Open Anyway button next to the Juka app.



6. You should now be able to run Juka.

Alternative Workaround

You can use the following workaround:

Assuming that Juka is stored on your Desktop, open a Terminal window and navigate to the Juka directory. Run the following commands:

```
chmod +x ./Juka  
sudo spctl --master-disable ./Juka
```

3. This will disable Gatekeeper for the current session, allowing you to run Juka.
4. Once you are done using Juka, run the following command to re-enable Gatekeeper:

```
sudo spctl --master-enable
```

This will re-enable the "gatekeeper".

Notes

- Juka is currently not signed by Apple, so you may encounter this error message when you first run it.
- We are working on getting Juka signed by Apple, so this error message should not be an issue in the future.
- If you have any further questions, please feel free to contact us.

Site Mirrors

If the official Juka website, <https://jukalang.com>, is broken or is slow, try using one of our other mirrors:

- **Mirror 1:** <https://jukalang.onrender.com/>
- **Mirror 2:** <https://juka.netlify.app>

We are constantly working to improve the performance of our website, but sometimes there may be issues that are beyond our control. If you experience any problems, please try using one of our mirrors.

FAQs

Q: Does Juka work on Linux?

A: Yes, Juka is compatible with Windows, Linux, FreeBSD, RaspberryPi, and Macintosh. It can also run on the cloud via Juka Azure function. We also plan to support mobile platforms such as Android and iOS/iPadOS.

Q: Can I try Juka without downloading?

A: Yes, you can try Juka online at <https://ide.jukalang.com>.

Q: What is Juka used for?

A: Juka is a general-purpose programming language that can be used to create a variety of applications, including web applications, desktop applications, and mobile applications. It is also a good choice for rapid prototyping and scripting.

Q: Could this language help me in _?

A: It might. The language is still in early stages, but we are looking for active members. If you have a specific project in mind, please let us know and we can see if Juka is a good fit.

Q: Is Juka Copyrighted?

A: Juka is provided under modified AGPL GNU license. See <https://jukalang.com/license> for more information.

Q: How do I start using Juka?

A: There are a few different ways to start using Juka. You can download the Juka compiler and run it on your own computer, or you can use one of the online Juka interpreters. Once you have a Juka compiler or interpreter, you can start writing Juka code.

Q: Do you have an IDE or a text editor where I can write the code?

A: You can write the code using any text editor. If you like Jupyter, you can install Juka kernel, see [run Juka in Jupyter](#).

Q: I want to host Juka compiler online... How can I do it?

A: There are many ways to host Juka Compiler online. One of the ways is using Microsoft Azure Server (See the [Azure Function documentation](#)). If you do not want to use Azure, you can download JukaApi.