

# Jenkins

Step 1: To run a simple JAVA code

Create a New Jenkins Job

Go to the Jenkins dashboard and click **New Item**.

Choose **Freestyle Project** and give it a name, for example “Java”

Click **OK** to create the job.

The screenshot shows the 'New Item' form in Jenkins. At the top, it says 'New Item'. Below that is a text input field labeled 'Enter an item name' with the value 'Java'. Underneath is a section 'Select an item type' with four options: 'Freestyle project' (selected), 'Pipeline', 'Multi-configuration project', and 'Folder'. Each option has a description. At the bottom, there is a blue 'OK' button.

**New Item**

Enter an item name

Java

Select an item type

- Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps like archiving artifacts and sending email notifications.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for workflows and/or organizing complex activities that do not easily fit in free-style.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as platform-specific builds, etc.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together! Folder creates a separate namespace, so you can have multiple things of the same folders.

OK

Step 2:

Go to build environment section click on add build step click on “execute windows bash command”

The screenshot shows the 'Build Environment' section of the Jenkins job configuration. On the left, there is a sidebar with navigation links: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment' (selected), 'Build Steps', and 'Post-build Actions'. The main area shows a list of build steps with a search filter. The 'Execute Windows batch command' step is highlighted. Below the list is an 'Add build step' button. At the bottom, there is a 'Post-build Actions' section with an 'Add post-build action' button and 'Save' and 'Apply' buttons.

General

Source Code Management

Build Triggers

**Build Environment**

Build Steps

Post-build Actions

Filter

- Custom Python Builder
- Execute Python script
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

Add build step ^

Post-build Actions

Add post-build action v

Save Apply

## Build Steps

≡ Execute Windows batch command ?

Command

See [the list of available environment variables](#)

Advanced ▾

Add build step ▾

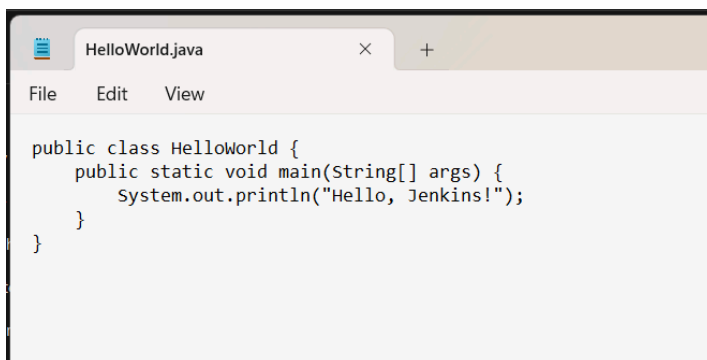
### Step 3: Write a Simple Java Program

Create a file in ur device

Write the below code in it

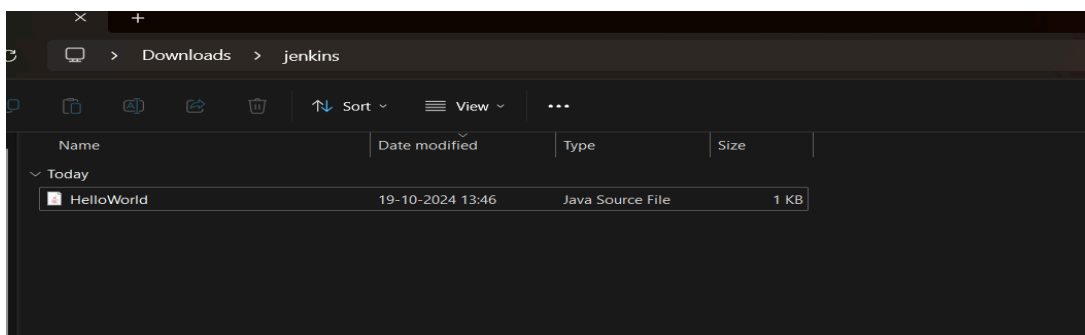
```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, Jenkins!");  
    }  
}
```

Save this as **HelloWorld.java** on your local machine.



A screenshot of a code editor window titled 'HelloWorld.java'. The editor shows the following Java code:

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



## Step 4: To run the code

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
cd C:\Users\Nihar\Downloads\jenkins
javac HelloWorld.java
java HelloWorld
```

Advanced ▾

Add build step ▾

Give the location of your file and run the java file  
And Save the project

## Step 5: Build the project Click on build now

Status

</> Changes

Workspace

▶ Build Now

⚙️ Configure

🗑️ Delete Project

★ Favorite

🌊 Open Blue Ocean

✎ Rename

Click on the build program #1  
And then click on the console output for the final output of ur java program

Status

</> Changes

📄 Console Output

📝 Edit Build Information

🗑️ Delete build '#1'

🕒 Timings

🌊 Open Blue Ocean

✓ Console Output

Download

Started by user [Nihar Morye](#)

Running as SYSTEM

Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Java

[Java] \$ cmd /c call C:\WINDOWS\TEMP\jenkins16735938022438897265.bat

C:\ProgramData\Jenkins\jenkins\workspace\Java>cd C:\Users\Nihar\Downloads\jenkins

C:\Users\Nihar\Downloads\jenkins>javac HelloWorld.java

C:\Users\Nihar\Downloads\jenkins>java HelloWorld

Hello, Jenkins!


C:\Users\Nihar\Downloads\jenkins>exit 0

Finished: SUCCESS


For executing Python program follow the same steps  
Create a project name it keep it freestyle project

Enter an item name


Select an item type




**Freestyle project**  
Classic, general-purpose job type that checks out from up to or  
steps like archiving artifacts and sending email notifications.



**Pipeline**  
Orchestrates long-running activities that can span multiple build  
workflows) and/or organizing complex activities that do not eas




**Multi-configuration project**  
Suitable for projects that need a large number of different conf  
platform-specific builds, etc.



**Folder**  
Creates a container that stores nested items in it. Useful for gro  
folder creates a separate namespace, so you can have multiple  
folders.

OK

Create a file in ur local device write a python code in it save it with the python extension

Name	Date modified	Type	Size
▼ Today			
 python	19-10-2024 13:58	Python File	1 KB

≡

Execute Windows batch command ?

×

Command

See [the list of available environment variables](#)


```
cd C:\Users\Wihar\Downloads\jenkins
python python.py
```

Advanced ▼

Apply and save it  
Click on build now

 **Status**


 Changes


 Workspace

 Build Now


 Configure


 Delete Project

 Favorite


 Open Blue Ocean


 Rename

 **Build History** trend ▼

 **#1**

Oct 19, 2024, 1:59 PM

 Atom feed for all

 Atom feed for failures

Open the #1 build  
Click on console output

 Status

 Changes

 **Console Output**

 Edit Build Information

 Delete build '#1'

 Timings

 Open Blue Ocean

 **Console Output**

Download

Copy

View as plain text

```
Started by user Nihar Morye
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Python
[Python] $ cmd /c call C:\WINDOWS\TEMP\jenkins11494737073072233894.bat

C:\ProgramData\Jenkins\jenkins\workspace\Python>cd C:\Users\Nihar\Downloads\jenkins

C:\Users\Nihar\Downloads\jenkins>python python.py
Hello World!

C:\Users\Nihar\Downloads\jenkins>exit 0
Finished: SUCCESS
```

For running a parameterized java code on jenkins  
Step 1: Create a new project name it keep it as a free style project

## NEW ITEM

Enter an item name

Java parameter

Select an item type



### Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps like archiving artifacts and sending email notifications.



### Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for build workflows and/or organizing complex activities that do not easily fit in free-style job type.



### Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing platform-specific builds, etc.



### Folder

Creates a container that stores nested items in it. Useful for grouping things together. Each folder creates a separate namespace, so you can have multiple things of the same name in different folders.

OK

Step 2:

Under the **General** tab, you can add a description.

Check **This project is parameterized**.

## General

Description

Plain text [Preview](#)

☐ Discard old builds [?](#)

☐ GitHub project

☒ This project is parameterized [?](#)

Add Parameter [v](#)

☐ Throttle builds [?](#)

☐ Execute concurrent builds if necessary [?](#)

Advanced [v](#)

Click on **Add Parameter**, and choose a parameter type like **String Parameter**

The screenshot displays the Jenkins 'String Parameter' configuration page. It shows two parameters, 'NUMBER1' and 'NUMBER2', each with a name, a default value, a description field, and a 'Trim the string' checkbox. The 'NUMBER1' parameter has a default value of '0'. The 'NUMBER2' parameter has a default value of 'd'. Below the parameters, there is an 'Add Parameter' button.

String Parameter ?

Name ?

NUMBER1

Default Value ?

0

Description ?

Plain text [Preview](#)

☐ Trim the string ?

String Parameter ?

Name ?

NUMBER2

Default Value ?

d

Description ?

Plain text [Preview](#)

☐ Trim the string ?

Add Parameter ▾

Add parameters give it a name and set a default value

Step 3:

Create a Java program that accepts parameters from the command line (Jenkins will pass these parameters). Here's the code for the addition program:

```
// Addition.java
public class Addition {
    public static void main(String[] args) {
        // Ensure there are exactly 2 arguments passed
        if (args.length != 2) {
            System.out.println("Please provide exactly two numbers as arguments.");
            return;
        }

        try {
            // Parse the input arguments as integers
```

```


int num1 = Integer.parseInt(args[0]);
int num2 = Integer.parseInt(args[1]);

// Perform the addition
int result = num1 + num2;

// Output the result
System.out.println("The sum is: " + result);
} catch (NumberFormatException e) {
    System.out.println("Invalid input. Please enter two valid integers.");
}
}
}

```

Save it in ur local device with java extension

✓ Today			
 Addition	19-10-2024 14:13	Java Source File	1 KB

Now execute the code in the windows batch command

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

cd C:\Users\Nihar\Downloads\jenkins

javac Addition.java

java Addition %NUMBER1% %NUMBER2%

Advanced ▾

Add build step ▾

Run the Java program with Jenkins parameters as given above  
Apply and save the project



## Click on build parameter

**Project Java parameter**

This build requires parameters:

NUMBER1  
25

NUMBER2  
64

**Build** **Cancel**

**Build History** trend ▾

No builds

Atom feed for all Atom feed for failures

## Add parameter in the number 1 & number 2 and click on build to execute the program

Dashboard > Java parameter > #1 > Console Output

**Console Output**

```
Started by user Nihar Morye
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Java parameter
[Java parameter] $ cmd /c call C:\WINDOWS\TEMP\jenkins2764960938517340024.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Java parameter>cd C:\Users\Nihar\Downloads\jenkins

C:\Users\Nihar\Downloads\jenkins>javac Addition.java

C:\Users\Nihar\Downloads\jenkins>java Addition 25 64
The sum is: 89

C:\Users\Nihar\Downloads\jenkins>exit 0
Finished: SUCCESS
```

U can do the same in python using parameters