



New Item

Enter an item name

Select an item type



Freestyle project

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



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as 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 on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Copy from

Jenkins / All / addparameters / Configuration

Configure This project is parameterized ?

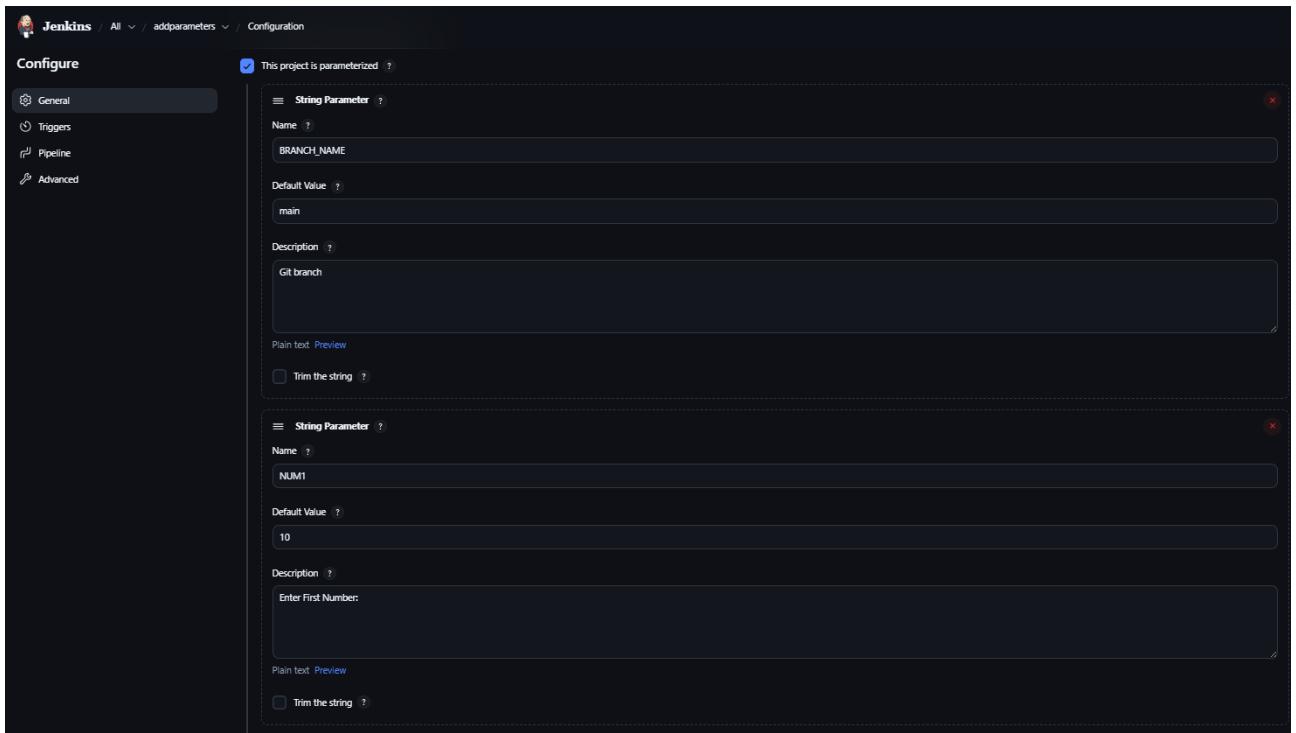
General Triggers Pipeline Advanced

String Parameter

Name : BRANCH_NAME
Default Value : main
Description : Git branch

String Parameter

Name : NUM1
Default Value : 10
Description : Enter First Number



Jenkins / All / addparameters / Configuration

Configure Trim the string ?

General Triggers Pipeline Advanced

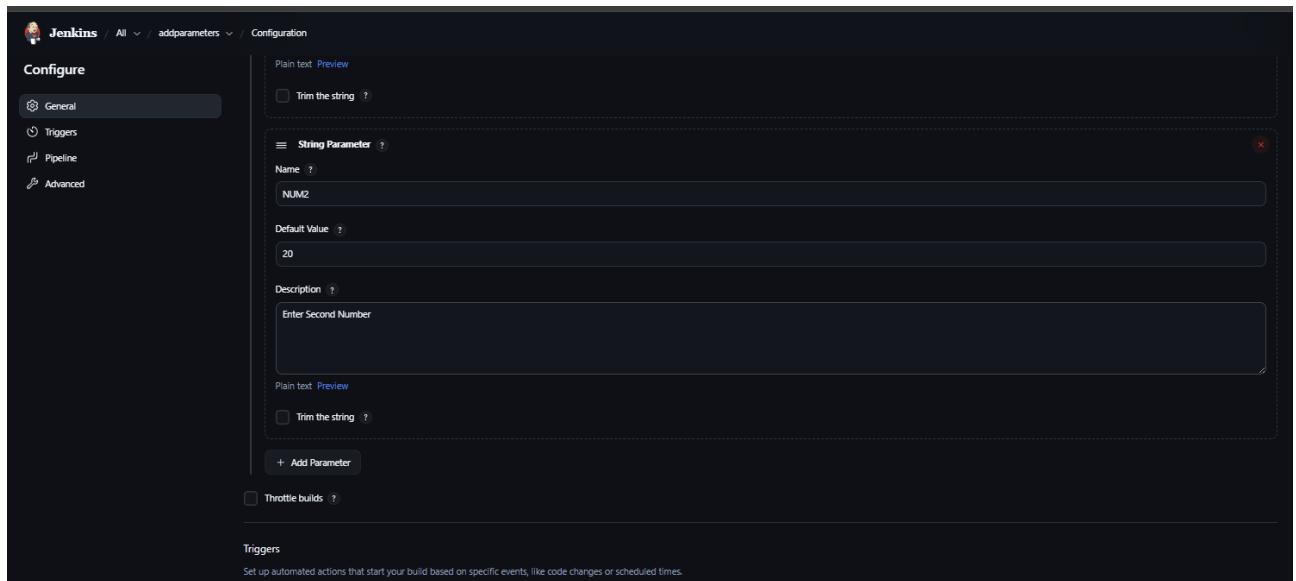
String Parameter

Name : NUM2
Default Value : 20
Description : Enter Second Number

+ Add Parameter

Throttle builds ?

Triggers
Set up automated actions that start your build based on specific events, like code changes or scheduled times.



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

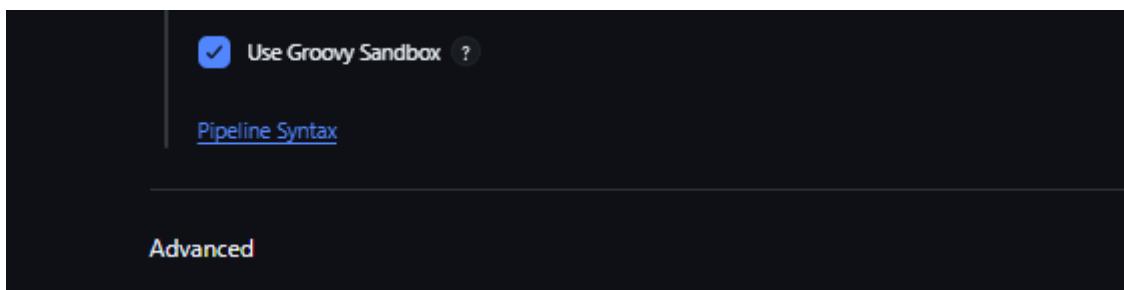
Definition

Pipeline script

```
Script ?  
1 pipeline {  
2   agent any  
3   parameters {  
4     string(name: 'BRANCH_NAME',  
5        defaultValue: 'main',  
6        description: 'Git branch')  
7     string(name: 'NUM1',  
8        defaultValue: '10',  
9        description: 'Enter First Number')  
10    string(name: 'NUM2',  
11       defaultValue: '20',  
12       description: 'Enter Second Number')  
13  }  
14  stages {  
15    stage('Checkout') {  
16      checkout([$class: 'GitSCM',  
17        branches: [[name: 'refs/heads/main']],  
18        doGenerateSubmoduleConfigurations: false,  
19        extensions: [[[$class: 'PruneStaleBranch'],  
20          branches: 'main',  
21          daysToKeep: 10,  
22          numToKeep: 10],  
23          strategyName: 'Kotlin',  
24          type: 'prune'],  
25          type: 'git'],  
26        submoduleCfgs: [],  
27        userRemoteConfigs: [[branch: 'main',  
28          credentialsId: '433f3a2c-1a2d-4a2e-8a2b-1a2d2a2e2a2f',  
29          url: 'https://github.com/swetha1507-s/mygitrepol.git']]  
30      ]  
31    }  
32    stage('Build') {  
33      steps {  
34        bat 'echo Hello World'  
35      }  
36    }  
37    stage('Test') {  
38      steps {  
39        echo 'Test Stage'  
40      }  
41    }  
42  }  
43}
```

Use Groovy Sandbox ?

[Pipeline Syntax](#)



Jenkins / All / addparameters / Pipeline Syntax

Overview

This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step

archiveArtifacts Archive the artifacts

- archiveArtifacts: Archive the artifacts
- bat: Windows Batch Script
- build: Build a job
- catchError: Catch error and set build result to failure
- checkout: Check out from version control
- cleanWs: Delete workspace when build is done
- deleteDir: Recursively delete the current directory from the workspace
- dir: Change current directory
- echo: Print Message
- emailExt: Extended Email
- emailRecipients: Extended Email Recipients

Go to file Add file Code

Local Codespaces

Clone

HTTPS SSH GitHub CLI

<https://github.com/swetha1507-s/mygitrepol.git>

Clone using the web URL.

Open with GitHub Desktop Download ZIP

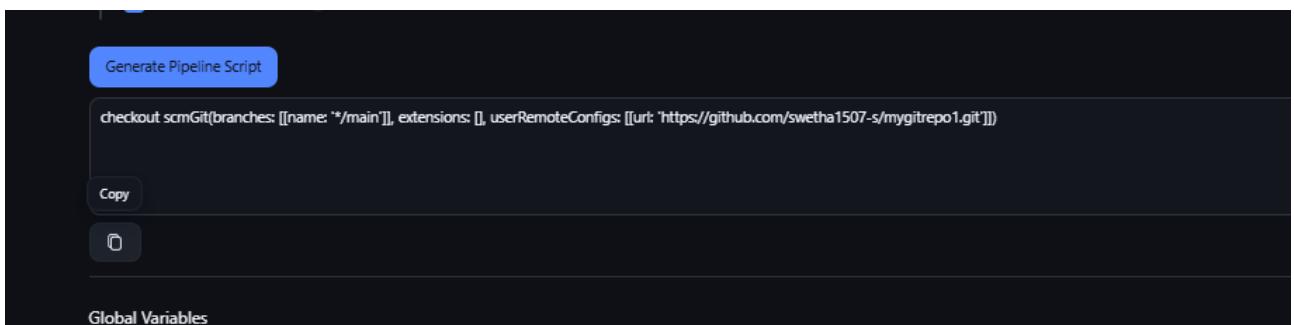
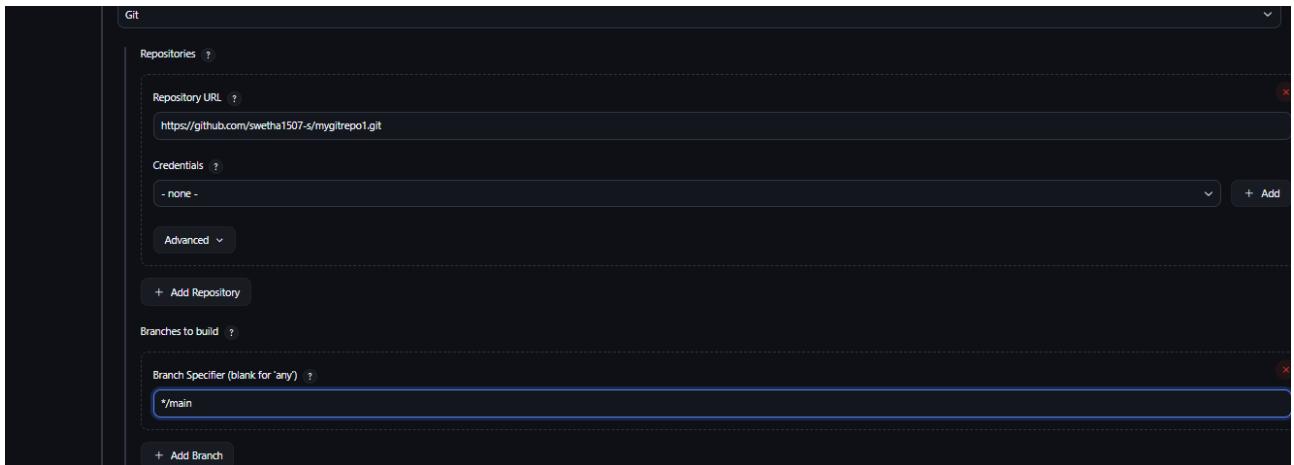
About

learning git

Activity 0 stars 0 watching 0 forks

Releases

1 tags Create a new release



No need ;

directly give repo url in url

pipeline script

```
pipeline {  
    agent any  
    parameters {  
        string(name: 'BRANCH_NAME',  
              defaultValue: 'main',  
              description: 'Git branch')  
        string(name: 'NUM1',  
              defaultValue: '10',  
              description: 'Enter First Number:')  
        string(name: 'NUM2',  
              defaultValue: '20',  
              description: 'Enter Second Number')  
    }  
    stages {  
        stage('Checkout') {  
            steps {
```

```

git branch: "${params.BRANCH_NAME}",
url: 'https://github.com/swetha1507-s/mygitrepo1.git'
}
}
stage('Addition Build') {
steps {
bat "python add.py ${params.NUM1} ${params.NUM2}"
}
}
}
}

```

Definition

Pipeline script

Script ?

```

14v stages {
15v   stage('Checkout') {
16v     steps {
17v       git branch: "${params.BRANCH_NAME}",
18v       url: 'https://github.com/swetha1507-s/mygitrepo1.git'
19v     }
20v   }
21v   stage('Addition Build') {
22v     steps {
23v       bat "python add.py ${params.NUM1} ${params.NUM2}"
24v     }
25v   }
26v }
27v }
28v

```

Use Groovy Sandbox ?

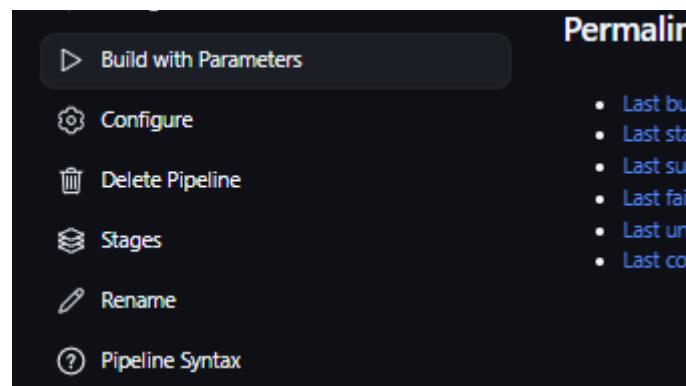
Pipeline Syntax

Advanced

Advanced ▾

Save Apply

click save



 Jenkins / All / addparameters

Pipeline addparameters

Status Changes Build with Parameters Configure Delete Pipeline Stages Rename Pipeline Syntax

This build requires parameters:

BRANCH_NAME
Git branch
main

NUM1
Enter First Number:
11

NUM2
Enter Second Number:
20

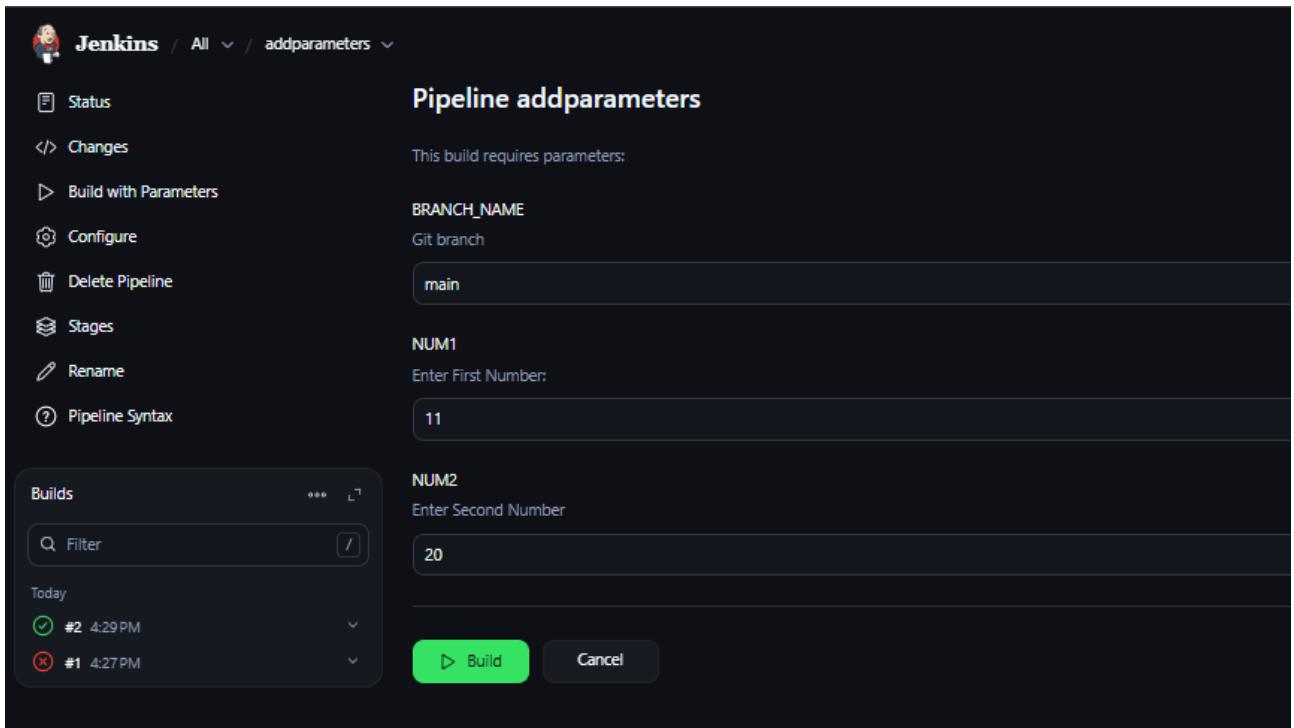
Builds

Filter

Today

#2 4:29PM
#1 4:27PM

> Build Cancel



Builds

Filter

Today

#3 4:38PM
#2 4:29PM
#1 4:27PM

</> Changes

Console Output

Edit Build Information

Delete build '#3'

Parameters

Timings

Git Build Data

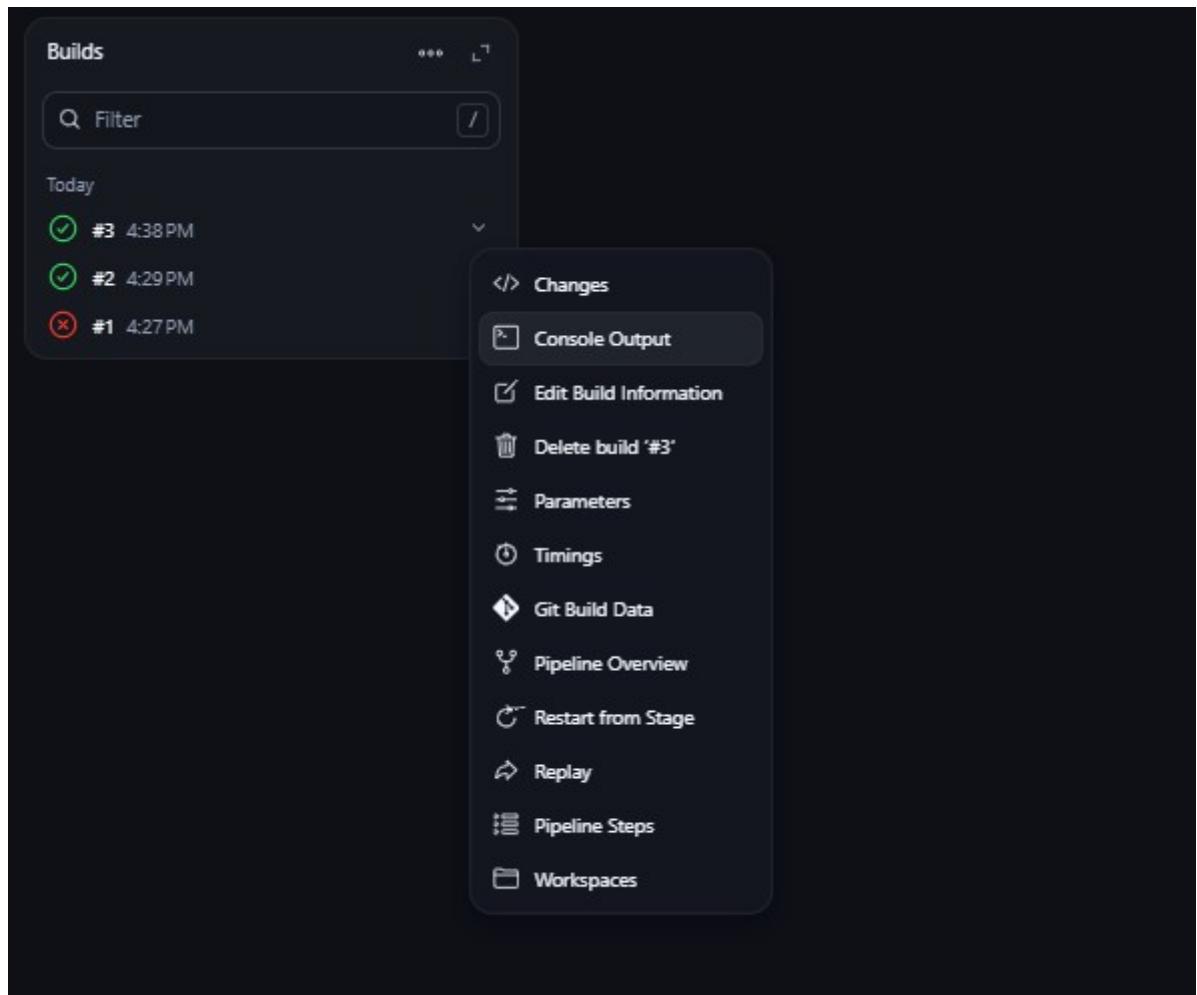
Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces



Jenkins / All / addparameters / #3 / Console Output

Status Changes Console Output Edit Build Information Delete build #3 Parameters Timings Git Build Data Pipeline Overview Restart from Stage Replay Pipeline Steps Workspaces Previous Build

Console Output

```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\addparameters
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\addparameters\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/swetha1507-s/mygitrepo1.git # timeout=10
Fetching upstream changes from https://github.com/swetha1507-s/mygitrepo1.git
> git.exe --version # timeout=10
> git.exe --version # 'git version 2.52.0.windows.1'
> git.exe fetch --tags --progress -- https://github.com/swetha1507-s/mygitrepo1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out revision d0ef7927e1ef25684d60925693bc3492ad5b97b (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f d0ef7927e1ef25684d60925693bc3492ad5b97b # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D main # timeout=10
> git.exe checkout -b main d0ef7927e1ef25684d60925693bc3492ad5b97b # timeout=10
Commit message: "Update add.py"
> git.exe rev-list --no-walk d0ef7927e1ef25684d60925693bc3492ad5b97b # timeout=10
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Addition Build)
[Pipeline] bat
C:\ProgramData\Jenkins\.jenkins\workspace\addparameters>python add.py 11 20
=====
Addition Result
=====
First Number : 11
Second Number: 20
Sum : 31
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```