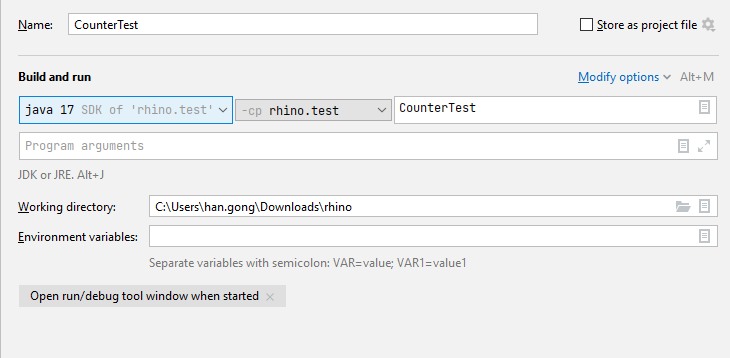
# Rhino Guide

Han Gong

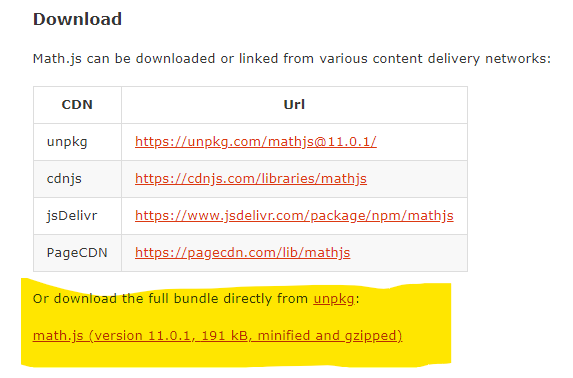
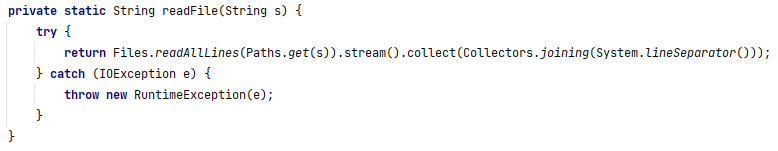
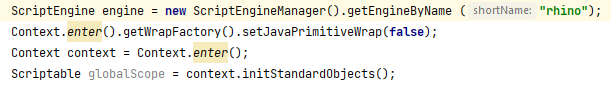
Rhino is an implementation of JavaScript in Java ([Github](https://github.com/mozilla/rhino)). This is a guide to install and implement a simple example to use Javascripts libraries within Java.

## Step 1 – Install Rhino

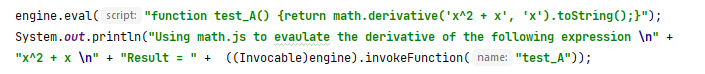
1. First clone the [Rhino repository](https://github.com/mozilla/rhino) to your local system. Open the project with your IDE.
2. Build the Rhino following the build instruction within the Github page using terminal (./gradlew test, testBenchmark might take a long time).
3. After the build is finished, open the \rhino\examples folder. Your IDE will prompt you set up the run environment. Run the CounterTest.java and check the output result with the comment within the code. The run configuration should look like the following

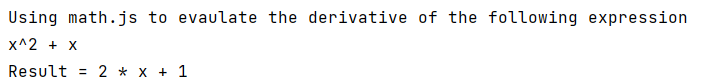


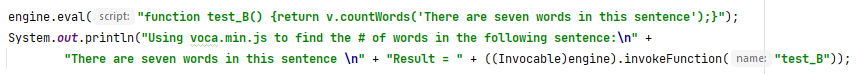
## Step 2 – Using Javascripts Libraries

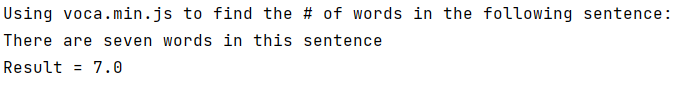
1. After we have correct output from the CounterTest.java, we can move on to implement Javascripts libraries using Rhino. The two examples of Javascripts library are [math.js](https://mathjs.org/) and [Voca](https://vocajs.com/).
2. In order to use Javascripts libraries, the library must be in UMD ([Universal Module Definition](https://github.com/umdjs/umd)). You can download the math.js [here](https://mathjs.org/download.html) and Voca [here](https://vocajs.com/) and move them to the project root folder.   
    
3. To use these libraries, we need to create a java class [JSLibrary](https://github.com/gonghanvip/rhino/blob/master/examples/JSLibrary.java).. The general roadmap to run the Javascripts:
   1. Read the library using: engine.eval(readFile(“MyJSLibrary.js”)); With the following readFile method:  
      
   2. Initialize the script engine with the following code segment:  
      
   3. Define your own scripts by callng: engine.eval(“function myFunction(){return MyJSLibrary.myLibraryFn(); }”);
   4. Execute your own script by calling: ((Invocable)engine).invokeFunction(“myFunction”);

## Step 3 – Sample outputs

1. For math.js, you can evaluate derivative by using the [derivative() function](https://mathjs.org/docs/reference/functions/derivative.html). For example, . 

The output for this code segment should be the following:  


1. For voca.min.js, we can count the number of words within a given string using the [countWords() function.](https://vocajs.com/#countWords) For example, there are seven words within the following sentence: “There are seven words in this sentence”.   
     
   The output of this code segment should be the following:



## Notes:

My personal fork for this Rhino project can be found [here](https://github.com/gonghanvip/rhino).