

OSEMN Assignment - Java v/s JavaScript

April 20, 2015

Overview

GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. Unlike Git, which is strictly a command-line tool, GitHub provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

GitHub offers both paid plans for private repositories and free accounts, which are usually used to host open-source software projects. As of 2015, GitHub reports having over 9 million users and over 21.1 million repositories, making it the largest code hoster in the world.

Projects on GitHub can be accessed and manipulated using the standard git command-line interface and all of the standard git commands work with it. GitHub also allows registered and non-registered users to browse public repositories on the site. Multiple desktop clients and git plugins have also been created by GitHub and other third parties which integrate with the platform.

The site provides social networking-like functions such as feeds, followers, wikis (using wiki software called Gollum) and a social network graph to display how developers work on their versions ("forks") of a repository and which fork (and branch within that fork) is newest. A user must create an account in order to contribute content to the site, but public repositories can be browsed and downloaded by anyone. With a registered user account, users are able to discuss, manage, create repositories, submit contributions to others' repositories, and review changes to code.

GitHub also operates other services: a pastebin-style site called Gist which is for hosting code snippets (GitHub proper is for hosting larger projects), and a slide hosting service called Speaker Deck. The software that runs GitHub was written using Ruby on Rails and Erlang by GitHub, Inc. developers Chris Wanstrath PJ Hyett, and Tom Preston-Werner.

GitHub launched a new program to give students free access to their tools on 7 October 2014. It is called the GitHub Student Developer Pack. GitHub will partner with Bitnami, Crowdfunder, DigitalOcean, DNSimple, HackHands, Namecheap, Orchestra, Screenhero, SendGrid, Stripe, Travis CI and Unreal Engine to launch the program.

GitHub's Terms of Service do not require public software projects hosted on GitHub to meet the Open Source Definition. Therefore (and in any case) it is advisable for users and developers intending to use a piece of software found on GitHub to read the software license in the repository (usually found in a

top-level file called "LICENSE", "LICENSE.txt", or similar) to determine if it meets their needs.

Scope

GitHub is mostly used for code.

In addition to source code, GitHub supports the following formats and features:

- Documentation, including automatically-rendered README files in a variety of Markdown-like file formats (see README files on GitHub)
- Issue tracking (including feature requests)
- Wikis
- Small websites can be hosted from public repositories on GitHub. The URL format is <http://projectname.github.io>.
- Small websites can be hosted from public repositories on GitHub. The URL format is <http://projectname.github.io>.
- Nested task-lists within files
- Visualization of geospatial data
- Gantt charts
- 3D render files which can be previewed using a new integrated STL file viewer which displays the files on a 3D canvas. The viewer is powered by WebGL and Three.js.
- Photoshop's native PSD format can be previewed and compared to previous versions of the same file.

[Source: <http://en.wikipedia.org/wiki/GitHub>]

Data

To gather data from github, I used the Github API <https://api.github.com>. Parameters can be passed to this as queries. The following code describes how I derived my data:

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:utils':
##
##      View

library(plyr)
library(ggplot2)

url <- "https://api.github.com/" # the GitHub API
path <- "search/repositories" # Repositories path
searchJava <- "?q=language:Java" # And language = Java
searchJavaScript <- "?q=language:JavaScript" # Language = JavaScript

URLjava <- paste0(url, path, searchJava)
URLjavascript <- paste0(url, path, searchJavaScript)

data_json_Java <- jsonlite::fromJSON(URLjava)
data_json_Javascript <- jsonlite::fromJSON(URLjavascript)

data_dataframe_Java <- as.data.frame(data_json_Java)
data_dataframe_Javascript <- as.data.frame(data_json_Javascript)

njava <- data_dataframe_Java[1, 1]
njavascript <- data_dataframe_Javascript[1, 1]

# Creating a table with these values:

language <- c("Java", "JavaScript")
repositories <- c(get("njava"), get("njavascript"))

graph <- cbind(language, repositories)
graph_data_frame <- as.data.frame(graph)
```

Results

So, we have tables for repositories created in Java and in JavaScript. Seeing the values in the tables, we can verify the language in which the repository was created.

The number of repositories for each, Java and JavaScript can be calculated as follows:

```

class(graph_data_frame)

## [1] "data.frame"

str(graph_data_frame)

## 'data.frame': 2 obs. of  2 variables:
## $ language      : Factor w/ 2 levels "Java","JavaScript": 1 2
## $ repositories: Factor w/ 2 levels "1039741","1292428": 1 2

summary(graph_data_frame)

##           language repositories
## Java           :1    1039741:1
## JavaScript:1    1292428:1

graph_data_frame

##           language repositories
## 1           Java           1039741
## 2 JavaScript           1292428

write.csv(graph_data_frame,file="OSEMNassignment.csv")

```

Graphing

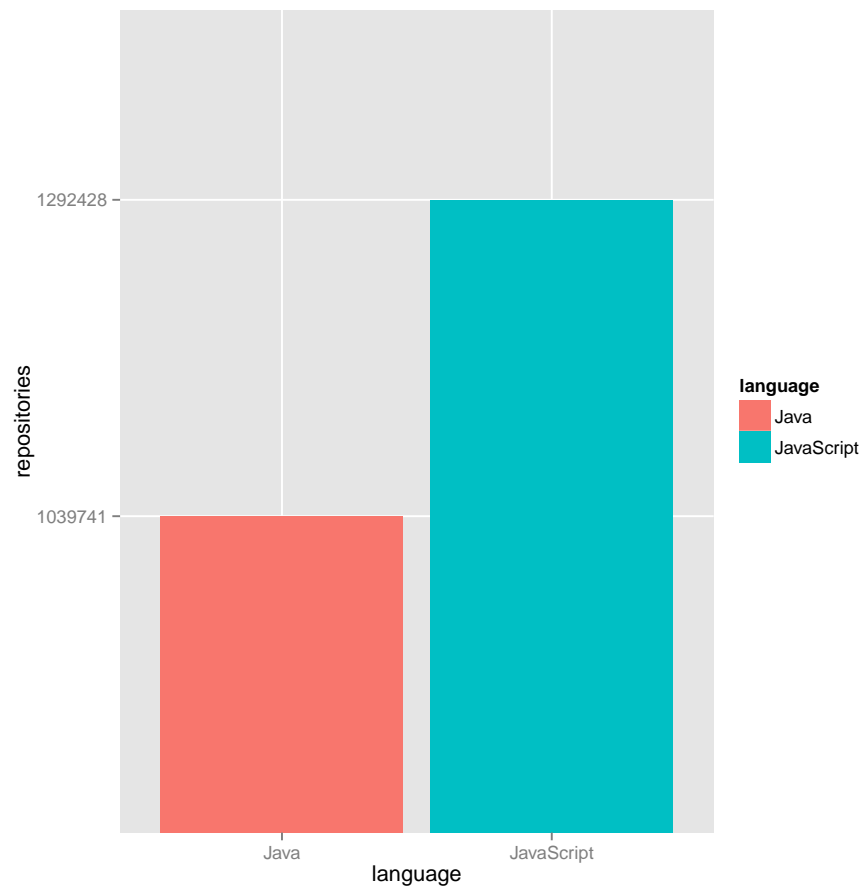
Having a table with language on X-axis and number of repositories for each language on Y-axis, we can easily plot a graph for that table. Once we have that graph, we can easily analyze our results.

```

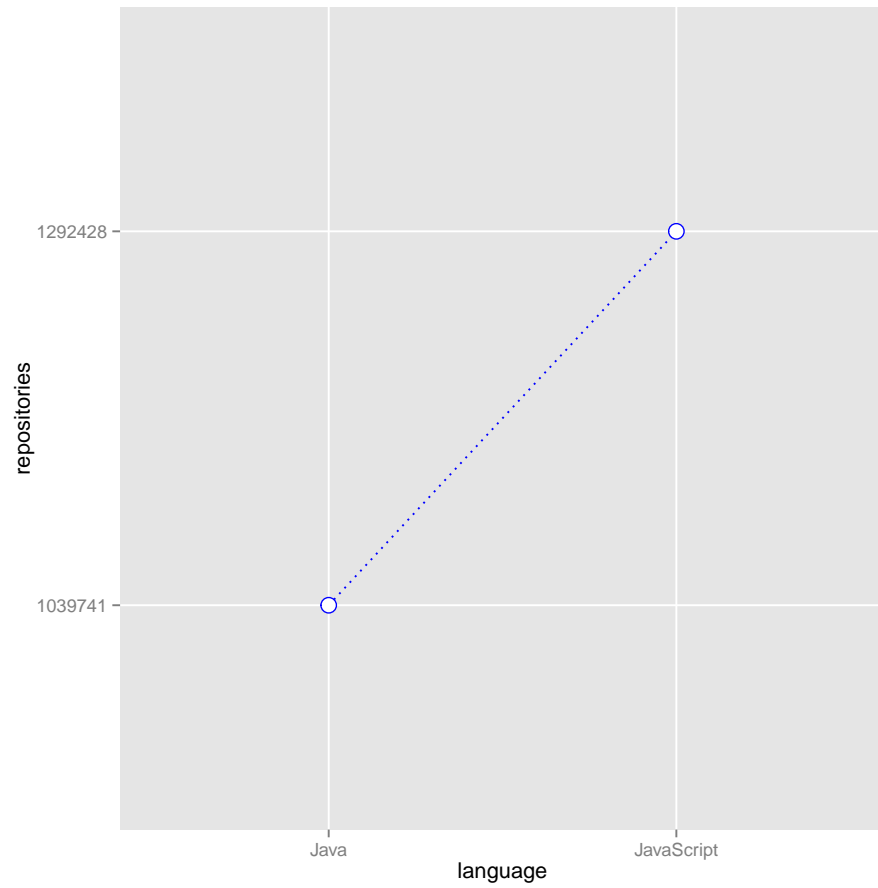
library(ggplot2)
library(scales)

ggplot2::ggplot(graph_data_frame,aes(x=language,y=repositories,fill=language))+
  geom_bar(stat="identity")

```



```
line<-ggplot2::ggplot(graph_data_frame,aes(x=language,y=repositories,group=1))+
  geom_line(color="blue", linetype="dotted")+
  geom_point(fill="white", color="blue", shape=21, size=4)
line
```



We can clearly see, that the number of repositories created in JavaScript is more than those created in Java.