

# Mozilla Rhino

**מגישים:** ארז אביטל, אלי זסלבצ'יק, אלכסנדר אוקונב

Rhino – זה מנוע של JavaScript המפותח כולו ב-Java ומנוהל ע"י Mozilla כתוכנת קוד פתוח.

## היסטוריה

פרויקט Rhino התחיל ב-Netscape ב-1997. בזמן זה Netscape תכננו לייצר דפדפן שכולו כתוב ב-Java ולכן הם היו צריכים יישום של JavaScript שכתוב גם כן ב-Java. כש-Netscape הפסיקו לעבוד על "Javagator", כך דפדפן היה נקרא, פרויקט Rhino היה מוכן כמנוע של JavaScript. מאז כמה מהחברות הגדולות (בהן גם חברת Sun) קיבלו רישיון להשתמש ב-Rhino במוצרים שלהן ושילמו ל-Netscape עבור זה, דבר זה אפשר להמשיך לעבוד על הפרויקט.

בהתחלה Rhino הידר כל JavaScript קוד ל-Java bytecode (סט הוראות של JVM – מכונה וירטואלית של Java) בתוך קבצי מחלקה של Java שנוצרו. זה נתן ביצועים יותר טובים, לעיתים קרובות אפילו יותר טוב ממימוש של JavaScript ב-C++, אבל היו לזה 2 חסרונות:

- (1) זמן הידור היה ארוך, מפני שייצור של bytecode וטעינת מחלקות שנוצרו היה תהליך עתיר משאבים.
  - (2) לא היה יישום של זיכרון זליגה אפקטיבי בגלל שרוב JVM לא אסף מחלקות או מחרוזות שלא בשימוש שנוצרו כתוצאה של טעינת קובץ מחלקה.
- כתוצאה בסתיו 1998 Rhino הוסיפו מצב פרשני. קוד יצירת קובץ מחלקה הועבר לחבילה אופציונלית שנטענת באופן דינאמי. הידור מהיר וכאשר סקריפטים לא בשימוש כבר, הם נאספים כמו כל אובייקט Java אחר.

Rhino שוחרר ל-Mozilla באפריל 1998. בהתחלה יצירת קובץ מחלקה לא הופיע גרסת שחרור, אבל א"כ כל קוד של Rhino שוחרר כקוד פתוח. מאז Rhino הופיע בהרבה שימושים ומספר רב של אנשים תרמו לקוד.

השם של פרויקט נובע מחיה שהופיעה על כריכת ספר של JavaScript מ-O'Reilly.

---

## **Rhino** מוטיבציה לשימוש ב

Rhino ממיר סקריפטים של JavaScript למחלקות. Rhino מיועד להיות בשימוש באפליקציות בצד שרת, לכן אין תמיכה מובנת עבור אובייקטים של גלישה ב-WEB, אשר בדרך כלל מקושרים ל-JavaScript.

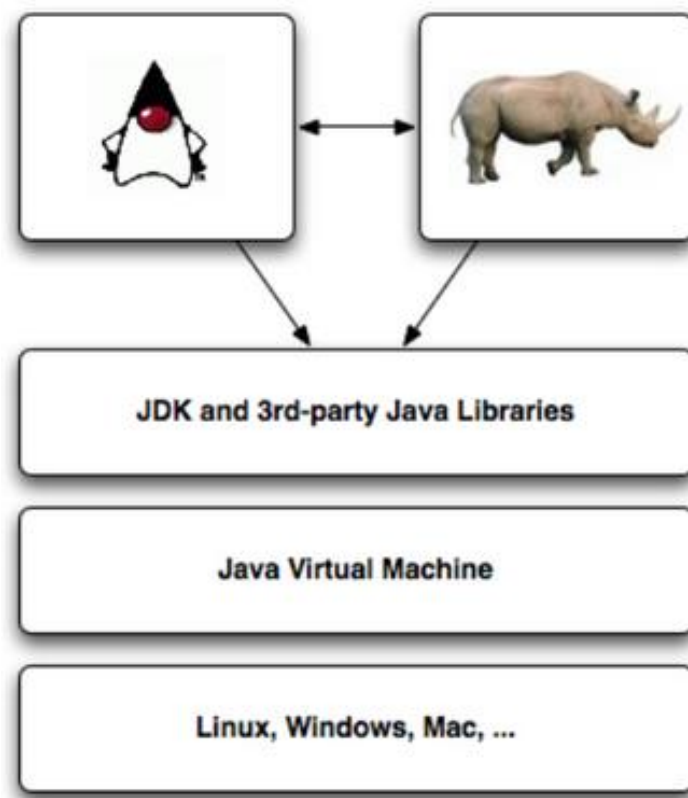
Rhino יכול לשמש כ-debugger ע"י שימוש במעטפת של Rhino. מעטפת של JavaScript מספקת דרך קלה להריץ סקריפטים במצב batch או בסביבה אינטראקטיבית עבור תכנות מחקרי.

## **Rhino – מבט מלמעלה ( איור 1 )**

קוד מקור של Java הופך ל- Java bytecode.

קוד מקור של Rhino JavaScript גם הופך ל- Java bytecode.

תוצאה ברשת: Rhino יכול לעשות מה ש-Java יכול לעשות.



איור 1

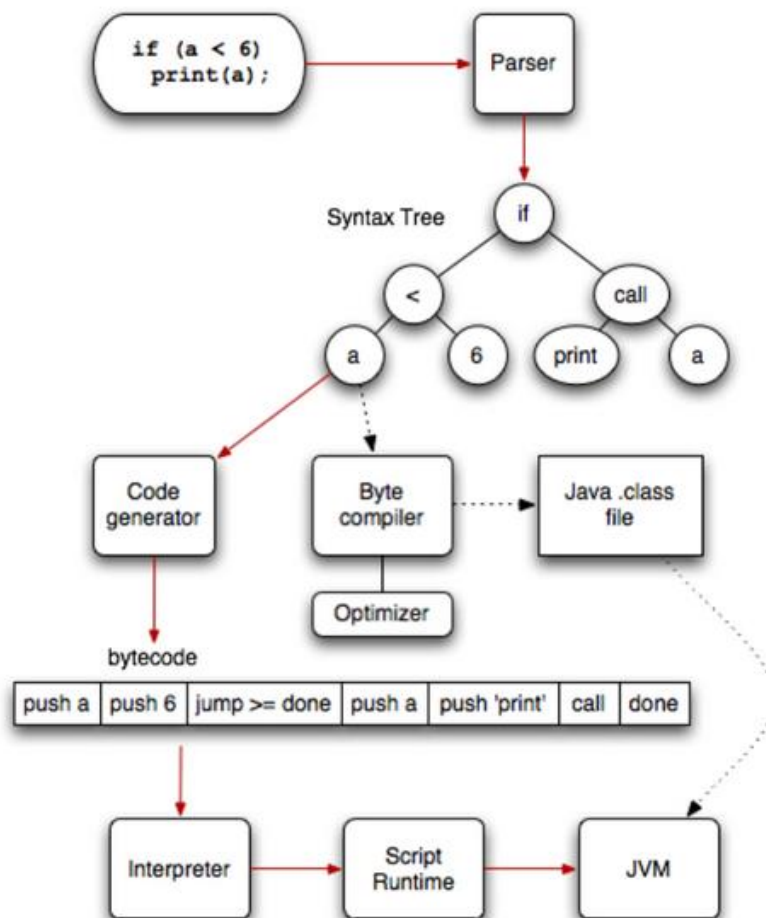
## Rhino – מה קורה בפנים (איור 2)

ל- Rhino יש 2 מסלולים: מתרגם ומקמפל.

- קוד מקומפל יותר מהיר
- לוקח יותר זמן לעשות codegen (כלי עבור יצירת קוד מקור וסוגים אחרים של תוכן, המתבסס על metadata השמור במאגר)

ריצת סקריפט מיישמת סמנטיקה של JavaScript

- אובייקטים מובנים
- אלגוריתמים של ECMA (ECMA – שפת סקריפטים, אשר בשימוש נרחב בסקריפטים צד-לקוח)
- גשר Java/JavaScript
- מנוע E4X
- מנוע RegExp
- מנוע אבטחה
- ממשק לבדיקת באגים
- ממשק עבור מתכנתים



## מימושים אפשריים - JavaScript Map-Reduce Jobs in Hadoop using Mozilla Rhino

כאשר נדרשים לכתוב מנגנון של Map Reduce מגלים כי יש צורך בידע נרחב בכתובת צד שרת (בדוגמא שלנו , שפת התכנות הינה Java כאשר ניתן לממשה בכל שפה אחרת)

בנוסף כאשר קיימים שינויים , נדרש לבצע קימפול של הקוד , ועם זאת החיסרון של דרך זאת.

כדי שנוכל לפשט כתיבת מנגנון שכזה , נדרש לבצע שימוש בשפה הדומה ל SQL, בשפה המתקמפלת וקריאה למנגנון באמצעות שפת סקריפט שבמקרה שלנו מודגמת Mozilla Rhino

על מנת לבצע פעולה זו נדרש להקים סביבה אירוח שנוכל להשתמש ב Rhino בהקשר של Hadoop, ובדרך הטובה לבצע זאת ע"י ביצוע ( delegate ) של מטודות Map, Reduce ב JS.

דוגמת קוד לשימוש בקוד JS Rhino:

Identity Mapper

*map.js*

```
1 function map(key, value, output, reporter){
2   oKey.set(key);
3   oValue.set(value);
4   output.collect(oKey, oValue);
5 }
```

Identity Reducer

*reduce.js*

```
1 function reduce(key, values, output, reporter){
2   oKey.set(key);
3   while(values.hasNext()){
4     output.collect(oKey, values.next());
5   }
6 }
```

מקור:

<http://www.gettingcirrius.com/2011/06/javascript-map-reduce-jobs-in-hadoop.html>