J Language

- J is a mathematical language based on the APL language and invented by Kenneth Iverson and Roger Hui
- J language is **terse** but **powerful**
- J is used by several corporations such as Hewlett Packard and Intel

<table>
<thead>
<tr>
<th>J Term</th>
<th>Other Language Term/Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb</td>
<td>Function or operator</td>
</tr>
<tr>
<td>Noun</td>
<td>Object or variable or constant</td>
</tr>
<tr>
<td>Copula</td>
<td>Assignment</td>
</tr>
<tr>
<td>Punctuation</td>
<td>Separator</td>
</tr>
<tr>
<td>Adverb</td>
<td>n/a</td>
</tr>
<tr>
<td>Conjunction</td>
<td>n/a</td>
</tr>
<tr>
<td>Sentence</td>
<td>Executable unit</td>
</tr>
</tbody>
</table>

Table from "A Casual J Tutorial"
http://jeffzellner.com/miidaj/

Defining Features

- Array based programming of J allows for loopless code.
- **Verbs** are short rules that are applied to an array from right to left.
- **Nouns** are objects such as integers, that verbs operate on.
- There are two kinds of verbs, **monads** and **dyads**. Dyads have arguments before and after the verb while monads are only followed by a noun.
- Monads and Dyads change the meaning of verbs which allow for more ways objects/nouns in arrays can be manipulated.

Example J language:

```j
run=: 2 2 $ 1 2 3 4 <-- '$' creates a 2x2 matrix named 'run' and fills it with the integers 1,2,3,4

^run
2.71828 7.38906

run^2
1 4
9 16
```

<--- monad form of the verb '^'
<--- for every object in 'run' y, e^y is outputed.

<--- dyad form of the verb '^'
use of this verb takes every
object of 'run' to the second power to output 1,4,9,16