

DATA MAN



LEFTS TRY TO BFS?

```
public void bfs() {  
    // BFS uses Queue data structure  
    Queue q = new LinkedList();  
    q.add(rootNode);  
    visited[rootNode] = true;  
    printNode(rootNode);  
  
    while( !q.isEmpty() ){  
        int n, child;  
        n = (q.peek()).intValue();  
        child = getUnvisitedChildNode(n); // Returns -1 if no unvisited niode left  
        if ( child != -1 ) {  
            // Found an unvisited node  
            visited[child] = true; // Mark as visited  
            printNode(child);  
            q.add(child); // Add to queue  
        }  
        else{  
            q.remove(); // Process next node  
        }  
    }  
}
```

OR MAY BE NOT

Let Data Man BFS

```
bfs i in G:  
  if (i.name is "Bill"):  
    print "My name is Bill\n"  
  else  
    print "I am not Bill"  
  end  
end
```



WHO IS DATA MAN?

A language that helps manipulate
large Data Sets easily.

READABLE

EASE OF USE

LESS CODE

BUZZZZZZZZ



WORDS

PORTABLE

Dragon-Book-Compatible

ROBUST

SYNTAX

No Semicolons;
Natural Language like
Readable

TASTE OF DATA MAN

```
num main():  
    String my_name = "Jeremy"  
    num my_age = 19  
  
    //System.out.println("NO THANKS");  
    if (my_age < 18):  
        print "Hello " + my_name " is " + my_age + "-year-old!\n"  
        print "A year from now you will be " + (my_age + 1) + "\n"  
    end  
end
```

WHY DATA MAN?

Graphs

Matrices

Types

BUILT IN GRAPHS & NODES

node a = "New York"

node b = "California"

graph CITES = a + b

"." operator CITIES.a = "Texas"

condtions apply: DATA MAN IS STILL TRAINING FOR THIS HE IS STILL
A KID SO RELAX!

GRAPH TRAVERSAL

BFS & DFS

```
dfs node_var in graph_var:  
    // some code here ...  
end
```

THE MATRIX

9	0	8	8	5	4	9	5	6	8	5	8	4	9	6	7	6	5	9	0	4	4	7	9	0	9					
7	2	0	2	3	7	5	0	8	4	2	8	6	4	5	2	3	6	7	3	4	2	2	9	2	0	8	9	8	7	
9	2	5	3	8	8	9	7	3	1	0	5	4	2	7	8	2	6	9	7	0	4	9	5	3	7	3	9	4	9	
0	6	7	9	8	3	7	4	7	2	4	1	5	0	4	8	3	6	7	6	6	2	0	4	1	5	5	4	9	0	
7	9	5	2	2	0	8	4	9	0	2	6	8	8	1	8	9	4	9	4	5	4	2	4	5	4	5	5	1	7	
1	8	6	9	4	0	6	0	0	4	4	1	8	8	5	8	8	2	5	8	3	2	3	5	4	9	6	3	8	1	
2	4	8	8	0	1	1	9	7	4	3	8	8	4	3	3	5	6	5	5	5	3	3	6	4	2	9	4	3	2	
1	5	5	8	0	9	3	4	1	0	4	9	8	9	6	3	7	6	1	6	4	7	4	8	0	0	7	9	9	1	
1	8	7	2	0	3	6	0	2	4	2	7	6	2	5	7	6	3	9	6	8	1	0	9	2	4	0	1	5	1	
4	7	9	1	4	9	7	1	1	1	2	5	3	0	6	8	2	9	8	2	5	3	1	9	4	0	2	5	5	4	
8	8	5	6	0	7	4	5	1	7	0	3	3	3	0	2	4	3	7	3	7	4	2	6	8	9	7	7	1	6	8
2	2	2	3	4	4	1	9	4	3	0	4	0	4	7	5	6	3	2	6	9	0	1	9	4	9	6	8	1	2	
7	0	8	8	2	7	6	3	8	9	0	5	4	6	6	9	1	4	3	7	5	8	6	4	0	4	1	1	4	7	
9	5	2	9	2	8	5	5	2	7	9	8	4	2	5	5	7	9	7	2	2	6	7	2	0	8	6	5	0	9	
7	0	0	5	8	9	8	6	7	6	1	5	3	8	9	4	5	9	5	0	9	6	0	5	4	2	2	3	0	7	
9	1	7	0	6	9	8	5	9	7	6	6	9	7	2	0	6	6	9	6	3	7	7	1	8	8	8	6	9	9	
1	8	3	1	4	8	9	5	7	8	1	1	5	1	6	3	8	5	4	7	8	1	5	0	9	5	8	6	5	1	
5	5	2	2	7	2	3	1	9	2	0	5	6	5	2	3	6	1	5	3	1	8	4	6	9	9	5	9	5	5	
8	2	5	0	6	1	0	1	1	7	3	2	2	0	8	8	5	0	2	9	8	0	9	9	0	8	1	1	7	6	

MATRIX DECLARATION & OPERATIONS

matrix A= {1, 2, 3; 4, 5, 6; 7, 8, 9}

matrix B= {1, 2, 3; 4, 5, 6; 7, 8, 9}

$A + B$	Matrix addition
$A - B$	Matrix subtraction
$A .* B$	Dot product (of 2- dimensional matrices)
$A * B$	Cross product
$ A $	Determinant
$\sim A$	Inverse
$\ A\ $	Norm

COMPOUND TYPES

Swag!

THE MOST COMMONLY USED TYPES

```
type my_birthday = {num year, num month, num day}
```

```
type fb_contacts = {string name, num age, string  
hometown}
```

```
fb_contacts roshen = {"Roshen", 25, "Townsville"}
```

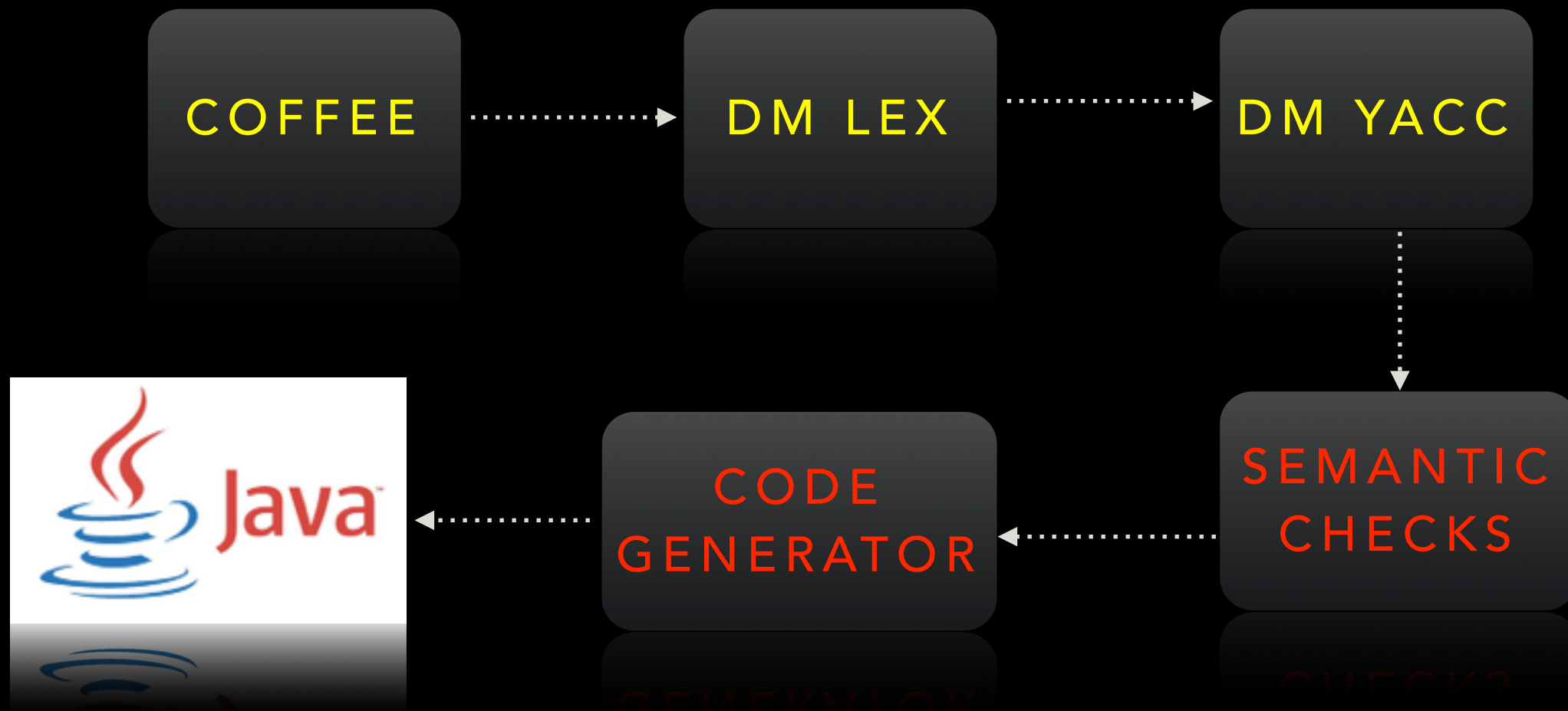
GIVE ME A BREAK!



DEMO TIME

Fingers Crossed!

DATAMAN'S SKELETON



COMPILER

PLY : Python Lex-Yacc

```
hz2317@paris:~/PLT/newproj/rehe/DataMan$ python DMyacc.py testing/graph_operations.mar
```

```
\translation_unit
  \declaration
    \{'line': 1, 'name': 'int', 'pos': 28}
  \init_decl_list
    \init_decl
      \func_decl
        \{'line': 1, 'name': 'main', 'pos': 32}
      \param_list
      \stmt_list
        \declaration
          \{'line': 66, 'name': 'node', 'pos': 104}
          \init_decl_list
            \init_decl
              \{'line': 66, 'name': 'new_york', 'pos': 110}
              \{'type': 'STRING', 'val': 'Nw York'}
          \declaration
            \{'line': 68, 'name': 'node', 'pos': 132}
            \init_decl_list
              \init_decl
                \{'line': 68, 'name': 'new_jersey', 'pos': 137}
                \{'type': 'STRING', 'val': 'New Jesey'}
          \declaration
            \{'line': 70, 'name': 'node', 'pos': 163}
            \init_decl_list
              \init_decl
                \{'line': 70, 'name': 'california', 'pos': 168}
                \{'type': 'STRING', 'val': 'California'}
          \declaration
            \{'line': 72, 'name': 'node', 'pos': 195}
            \init_decl_list
              \init_decl
                \{'line': 72, 'name': 'chicago', 'pos': 200}
                \{'type': 'STRING', 'val': 'Chicago'}
          \declaration
            \{'line': 74, 'name': 'node', 'pos': 221}
            \init_decl_list
              \init_decl
                \{'line': 74, 'name': 'texas', 'pos': 226}
                \{'type': 'STRING', 'val': 'Texas'}
          \declaration
            \{'line': 76, 'name': 'node', 'pos': 244}
            \init_decl_list
              \init_decl
                \{'line': 76, 'name': 'virginia', 'pos': 249}
                \{'type': 'STRING', 'val': 'Virginia'}
        \print
```

TESTING

PROJECT MISMANAGEMENT

Google Docs

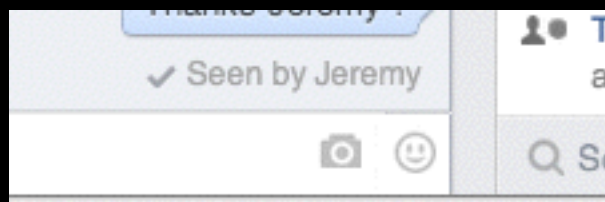


Git Hub



Facebook Chat

Evidence



PRO-CRASTINATION



Commits

LEARNED THE HARDWAY

- Not only Columbia has large Libraries - Apache does too
- Compiler is not this magic box
- A happy team is a productive team
- Version Control is important
- Learn to speak Java, C, Python, Chinese
- The name of the game is Agile!

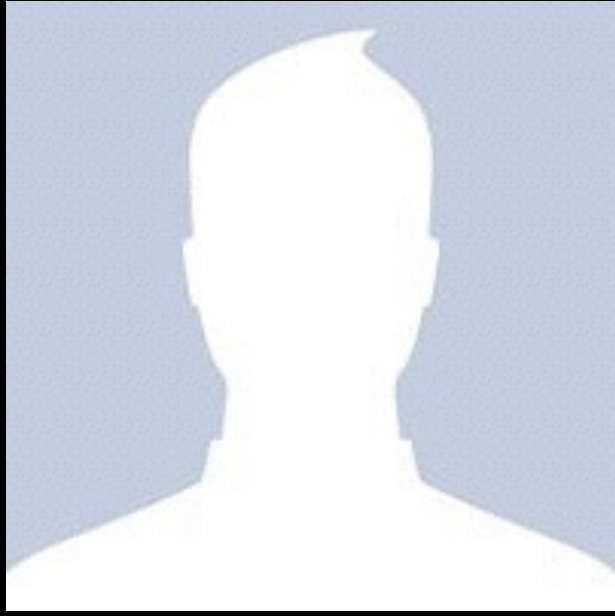
May be hello word deadline too

WE ARE THE PROUD PARENTS

Huayu



Zefeng



Jeremy



Roshen



DADDY!



Boxuan

FACED WITH A LARGE DATA SET



HERE HE COMES TO YOUR RESCUE

DATA MAN