

Default datatypes

String	str
Boolean	bool
Array	list[T]
Location	loc
Map / set	map[T, T]
Integer	int
Set (unique array)	set[T]

T = type. For example: list[str] for a list of strings

Imports

String	import String
Boolean	import Boolean
List	import List
Location	import Location
Map	import Map
Integer	import util::Math
Set	import Set
M3 AST	import analysis::m3::AST
IO	import IO

Read java project

```
createAstsFromDirectory(loc
project, true);
```

This method is located in the M3 AST library
return: set[Declaration]

Debugging

Print to console (one line)	println(value)
Print to console (formatted)	iprintln(value)

This methods are located in the IO library

Location

File	file:///home/rascal/rascal.rsc
Project	project://android-project/
URL	http://www.google.nl
Folder	file:///home/rascal/

Data manipulation 1

String - append	str : "str" + "str"
String - compare	bool : "str" == "str"
String - interpolation	"value: <str>"
List - assignment	list[int] : [3,2,1];
List - append	list[int] : [3,2] + 1;
Set - assignment	set[int] : {3,2,1}
Set - append	set[int] : {3,2} + 1;

All values are immutable

Data manipulation 2

```
// data Game = game(list[Player]
players);
// Game game = getGame();
// Access players:
game.players
// Overwrite players:
game.players = []
// Add player:
game.players = game.players +
player
```

If else

```
if("str" == "str") {
    iprintln("true");
} else {
    iprintln("false");
}
```

For loop (array)

```
list[int] numbers = [1, 2, 3, 4,
5];
for (number <- numbers) {
    iprintln(number);
}
```

For loop (map)

```
map[str, int] numbers = ["one" : 1,
"two" : 2,
"three" : 3, "four" : 4, "five" :
5];
for (numberText <- numbers) {
    int numberValue =
numbers[numberText];
    iprintln(numberValue);
}
```

File manipulation

Read file	readFile(location)
Write file	writeFile(location, value)

Method definition

```
public str getString() {
    return "String";
}
```

The method test() is reserved

Visit pattern - Matches

wildcard	_
Value check	"exact value"
Assignment	method
Block assignment	block:

Visit pattern - Example 1

```
visit (ast) {
    case \import(importName): {
        iprintln(importName);
    }
}
```

Visit pattern - Example 2

```
visit (ast) {
    case \import(_): {
        iprintln("Found an
import");
    }
}
```



By **rubene**
cheatography.com/rubene/

Visit pattern - Example 3

```
visit (ast) {
    case \import("ImportedClass"): {
        iprintln("found ImportedClass");
    }
}
```

Visit pattern - Example 4

```
visit (ast) {
    case
\field(simpleType(simpleName(name)), _): {
    iprintln("found a field: <name>");
}
}
```

Syntax keywords

layout The layout of the syntax. Is the syntax separated by spaces or new lines.

lexical Definition for a block of the input, which is separated by the layout.

start syntax The global syntax definition of the input

syntax The global syntax can be separated into smaller pieces, which can be defined with a syntax.



By **rubene**
cheatography.com/rubene/

Published 28th May, 2014.
Last updated 11th June, 2014.
Page 2 of 2.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>