

# Writing an Extension in Java

## Introduction

QDL has its own system for writing module, but some times you need to add functionality that is not in the base system. You may write your own custom Java code and import it. It will function like any other QDL module. It is really quite simple. There are two interfaces the you need to implement, one for functions and one for variables and over-ride a single method to load it all.

## Modules

A module the basic “encapsulated unit of execution” in QDL. That means that it has its own state and when you execute functions from a module, they execute there. All modules have a URI that identifies them uniquely and a human-readable alias that may be changed to suit the current user needs. Modules may not be nested in the sense that you cannot create a new one inside of one, but you can import them and use them.

## Functions

Functions must implement the `edu.uiuc.ncsa.qdl.extensions.QDLFunction` interface. This has a few methods (refer to the [Java documentation](#)). The basic way it works is that you tell how many arguments this may accept and when called, you will be given an array of objects which you must use. Note that in order to keep them straight you should specify in the function documentation what you expect the user to supply.

## Variables

Variables implement the `edu.uiuc.ncsa.qdl.extensions.QDLVariable` interface. This has two methods, one for the name and one for the value (as an Object).