Introduction to Quorum

About Quorum

What kind of Programming Language is Quorum?
Quorum is a general purpose tool that we call an "evidence-based" programming language. It started as an interpreted language originally designed to be easier to hear through screen readers for blind or visually impaired users. Eventually, Quorum became a general purpose programming language designed for any user. Current versions compile to Java Bytecode and run on the Java Virtual Machine, similarly to JRuby, Jython, or Scala. Quorum 3.0 also compiles to JavaScript and can be run from the web.

What are the high level technical details of Quorum?
Quorum has many features. It is object-oriented, but this is largely hidden from new users. It has a general purpose type system, with generics for containers (e.g., arrays, hash tables, lists). Quorum also has a standard library, which contains many additions to the language, like math libraries, containers, web components, or other features.

What platforms does Quorum support?
Quorum can be run on most platforms that support the Java Virtual Machine. We have run Quorum on various versions of Windows, Mac, and Linux. There is also have an online IDE for simple programs.

Hour of Code Tutorial
Quorum was featured in the latest Hour of Code in December 2014 and has had over 13,000 participants
The Quorum Tutorial can be found at code.org or through the Quorum web page at:

How is Quorum funded?
Quorum's initial funding started as part of a National Science Foundation project (NSF CNS-0940521).
Current funding for Quorum also comes from the National Science Foundation (NSF CNS-1440878).
Other funders include:
- Washington State School for the Blind
- Southern Illinois University Edwardsville
- University of Nevada, Las Vegas
- Do it program at the University of Washington

Quorum is open source
This means you can download the source code for the compiler or libraries or tools and modify them or contribute yourself. Quorum is under the BSD license.
It also means that Quorum is free!

What does Evidence-Based mean?
The people behind Quorum are researchers and scientists who believe in bringing randomized controlled trials to human factors decisions in programming language design.
Quorum started out as a project to simplify syntax to be accessible, and we continue to innovate on that front, but the project has expanded to improve programming language design overall. Quorum is now a full featured language.
The Quorum web site has a more detailed discussion of this important issue and our evidence standards.
Learn more at: www.quorumlanguage.com/evidence.php

Selected Research
Quorum Syntax Overview

Libraries
Quorum has a number of libraries already available and more are being added all the time. You can learn about them at: www.quorumlanguage.com/libraries.php
Libraries include:
• Accessibility
• Compute
• Containers
• Data
• Language
• Robots
• Sound
• System
• Web

Coming soon:
• Video Games
• Lego Robotics
• Scientific Computing

For Advanced Users
Quorum also supports most common programming language features like:
• Inheritance
• Generics
• Exceptions
• Plugins

For more information: www.quorumlanguage.com/syntax.php

Quorum is a full strength JVM language and the new Quorum 3 compiler (currently in beta) was written in the Quorum language itself.

Basics

Primitive Types

integer – a positive or negative number or zero with no decimal point
integer i = 5

number – any real number that can also have a decimal point
number n = 5.3

boolean – a variable with two possible values: true or false
boolean b = false

text – any string of characters symbols or numbers enclosed in ""
text t = “Hello World!”

Type Conversion*

Types can be converted in Quorum using the cast instructions as follows:
text t = "5.3"
number n = cast(number, t)

*not available in online mode

Variable names

Variable names must start with a letter, but can be followed by any combination of letters, numbers or underscores.

Legal:
number n5 = 5.3
number n_5 = 5.3

Illegal:
number 5n = 5.3
number n&$5 = 5.3

Operators

Math
Addition +
Subtraction –
Multiplication *
Division /
Modulus mod

Boolean
Logical equal =
Logical not equal not=

Text
Concatenation +

Statements

There are no semicolons or end of line markers needed in Quorum.

Conditional Statements

Simple:
if a = 1
end

If Else:
if a > 5
else
end

If Elseif:
if a > 10
elseif a < 10
else
end

Loop Statements

Repeat <expr> times:
repeat 10 times
end

Repeat while <expr>:
integer i = 0
repeat while i < 10
i = i + 1
end

Repeat until <expr>:
integer i = 0
repeat until i = 10
i = i + 1
end

Arrays*

Arrays can be accessed by using the array library:
Libraries.Containers.Array

Fill an array:
Array<integer> a
integer i = 0
repeat 5 times
a:add(i)
i = i + 1
end

Print the 4th element:
output a:Get(3)

Remove the first element:
i = a:RemoveFromFront()

Sort the array:
a:Sort()

*not available in online mode

Classes* and Actions

Classes*

Classes are defined in a code block between the class name and an end.
class Dog
end

*not available in online mode

Actions:

Actions are like methods in other languages. The Main method must be present for a class to be runnable and execution starts there.

action Main
Bark(3)
end

action Bark (integer i)
repeat i times
output "Bark"
end
end