CScharf

CScharf is an interpreted, general-purpose programming language that combines aspects of dynamic and strict typing. Primitive types are subject to strict typing while complex types such as classes are afforded some flexibilities that allow for more dynamic typing. Although a primarily object-oriented language, CScharf provides the ability to apply the procedural and functional programming paradigms.

/\* To expand on this, the typing of primitive types is similar to what is seen in C# (e.g. int val = 10, long x = 1000L;) but more complex types are typed based on a generic form of their type (e.g. instance square = new Square(), array values = new int[10]). \*/

Data types and variables

CScharf supports the following data types: integers, floating point numbers, doubles, Booleans, strings, anonymous types, higher-order functions, arrays, class instances (or objects), Java classes (through reflection), and allows for void to be used by functions as a return type. Values stored in anonymous types are immutable,

Variable names are limited to containing letters, numbers, and underscores and must start with with a letter or underscore. Due to its strict typing, CScharf requires that variables must be declared/defined with a type against which values are checked prior to assignment to ensure type safety.

Syntactically, CScharf shares many aspects with C# with minor exceptions regarding typing, functions, and reflection syntaxes.









Identifiers

## Semantics: Types, Variables, and Values

Types

In CScharf there are many primitive types with native support in the language, these types are: integer (int), floating point (float), boolean (bool), string (string), anonymous type (anon), function (func), array (array), class instances (instance), java classes through reflection (reflection).

Additionally, functions can return any of these types, as well as having no return type by using void in place of a type.

Lastly, there are two modifiers that can be applied to variables: const and readonly. Constant variables must be supplied a value, and readonly variables can only be modified in class constructors. More information will be provided in the Variables section.

Variables

Variable declarations and definitions have the following structure:

[modifier] <type> <identifier> = <value>

Constant and readonly variables are supported in CScharf with the difference being that constant variables cannot be declared so must be given a value (defined), and readonly variables can only be modified in the constructor of a class.