Trifle

- Trifle v0.3.0

1. Introduction

Warning

This specification is incomplete and will likely be changed in the future. Use this specification at your own risk!

Trifle is a multi-paradigm general purpose programming language that's (usually) dynamically typed and also compiled. During compilation it scans through the code looking for any potentetial bugs and **warns** (it doesn't throw an error) the user about them.

This specification is simple, but to understand this, I would recommend becoming familiar with a different programming language first to get the hang of programming.

2. Data Types

2.1. Null

The `null` keyword is normally referenced as an arbitrary value where a value is undefined.

2.2. Boolean

A Boolean is a logical token referencing to either `true` or `false`.

2.3. String

A String is a sequence of zero or more characters, normally wrapped in single (`'`) or double (`"`) quotes, like so: `"This is a string."`. Strings usually represent text.

2.4. Number

A Number represents either a single number (`0`, `1`, `1234567890`, ...) or the sum of a mathematical expression (`1 * 2`).

2.4.1. Mathematical Operators

Mathematical operators include:

- `N + x`: Adds x to N.
- `N - x`: Subtracts x from N.
- `N * x`: Multiplies N by x.
- `N / x`: Divides N by x.
- `N % x`: Represents the remainer of N divided by x. (`N / x`)
- `N++`: Adds `1` to `N`.
- `N--`: Subtracts `1` from `N`.
- `N += x`: Adds x to N.
- `N -= x`: Subtracts x from N.

2.5. Value Operators

Value operators tell the interpreter that a value is being defined. They will always start with the `@` symbol, such as `@func` or `@class`.

2.6. Variables

Variables reference a variable identifier (E.G. `$NAME`) to a value:

```
$<IDENTIFIER> = <VALUE>;
```

Where `<VALUE>` is an object and $NAME is a variable identifier. It requires a `;` at the end.

```
$myVar = 1;
```

Which can later be referenced as `$NAME`,

```
someFunction($myVar);
```

2.6. Function

Functions are macros that can be repeated and modified using parameters.

2.6.1. Function Definitions

Functions can be defined using Trifle's `func` value operator, followed by a function identifier, an argument list and a code block to be executed when the function runs.

```
@func <IDENTIFIER>(...) {
    // ...
}
```

For example:

```
@func myFunctionName($value) {
    // ...
}
```

2.6.2. Function Calls

Functions are called like they would be defined, minus the `func` value operator and code block.

```
<IDENTIFIER>(...);
```

Example:

```
@func myFunctionName($value) {
    // ...
}
myFunctionName("a string");
```

3.