

Introduction

Our language is called Wordy. This language has a **functional paradigm**.

This language is created to perform manipulations on strings.

Design

Core Features:

We determined core features by thinking of the most basic operations that can be performed on a string which then combined can be used to create more complex operations.

- **Sentence:** *Basic data types and operations.*
A basic string.
- **Num:** *Basic data types and operations.*
A basic number.
- **Count:** *Basic data types and operations.*
A feature that counts the number of words within a sentence.
- **Reverse:** *Basic data types and operations.*
A feature that reverses the order of words within a sentence.
- **Insert:** *Basic data types and operations.*
A feature that inserts a string into a sentence at a given location.
- **Remove:** *Basic data types and operations.*
A feature that splits removes a given string from a given sentence. (under development)
- **Equ:** *Conditionals*
A feature that compares either two integers, bools, or strings.
- **IfElse:** *Conditionals*
A conditional used to branch in our language.
- **While:** *Recursion/loops*
A looping feature that allows repetition while a condition is true. (under development)
- **Bind:** *Variables/local names*
Assign variable to a String (under development)
- **Cap:** *Basic data types and operations.*
Capitalizes a string (under development)
- **Function:** *Procedures/functions with arguments*
A feature can be defined and named to limit repeated code. They will use a call by value parameter. This will satisfy the Procedures/functions with arguments requirement.

Static type system (2)

Our language uses a static type system this will allow our programs to be checked for type errors upon execution.

Library Features: (under development)

We determined library features by thinking of useful and frequently used operations which can be implemented with the use of our core features.

- Capitalize:

A feature that capitalizes each letter of a given sentence.

- Lowercase:

A feature that makes each letter of a given sentence lowercase.

Sugar Features:

We determined sugar features by thinking of frequently used operations that can be shortened into keywords.

True: two string values are the same

False: two string values are different

And: If both strings are true

Or: If either string is true

Safety Features:

Using static type system, each of our *cmd* functions has a case condition with an Error result. If the input type into the command is not match, the output result is an **Error** string.

Implementation

So far, we have only one semantic domain is as follows:

Expr -> Value

Plan to add:

Stmt -> Value

This language is used to write programs that manipulate strings.