

DiceySpicyScript Language Specification

Helen Qian

Maia Wang

May 17, 2024

Introduction

This programming language would help the user write recipes. This program would provide an efficient way to turn jots down on a notepad into an easy-to-read recipe. Writing recipes on a notepad or paper is easy for anyone to do. However, pencil can get smudged and ink can wash away when it gets wet. Papers can get lost, dirty, or torn. Thus, digitalizing recipes is a smarter way to preserve recipes, make them easily replicable, and nice to look at.

Sometimes it can be a struggle to work with Google Docs or excel to make a nice recipe and it may take more time than the average Joe may have. Furthermore, most people don't even know what Latex is or how to use it. Through this language, we hope people can simply write out their quick notes about ingredients and instructions, and we will transform their writing into nicely formatted recipes. No need to waste time trying to order your ingredients separate from your instructions. Simply as you are creating your Michelin star worthy dish just make a note of the ingredient or instruction as you go. Our language will take lay man terms and format them into a file of latex code and then the user obtains a nicely downloaded pdf of their recipe, ingredients and instructions all neatly organized.

Design Principles

Aesthetically we made choices on how the pdf recipe template that is outputted. The ingredient list is bulleted and the instruction list is enumerated. We specifically choose the font Latin Modern Dunhill to give a cozy home baked feel to the recipe. The recipe pdf also includes design elements like the ingredients and instruction lists being inboxed. We kept the recipe design simple and easy to read but still aesthetically pleasing.

Examples

Example 1: (Example1.txt)

```
tit[Strawberry Shortcake]
ing[1 cup flour, 1 egg, 2 strawberries]
ins[add 1 egg to a bowl and mix, add 1 cup flour to bowl and whisk, chop strawberries]
```

Example 2: (Example2.txt)

```
ins[boil water, cook pasta] ins[add asparagus] tit[Creamy Asparagus Pasta] ing[1 lb short pasta] ing[1 lb asparagus,
two tbsp unsalted butter]
```

Example 3: (Example3.txt)

```
ing[1 bunch lettuce]
ins[rinse and chop lettuce in a bowl]
ing[10 dates]
ins[chop dates in half and add to bowl]
ing[1 slice blue cheese, 1 cucumber, 2 firm tomatoes]
ing[crumble and add blue cheese to bowl, slice up cucumber and add to bowl, slice tomatoes into fourths or even
smaller pieces and add to bowl]
```

```
ing[Italian dressing]
ins[mix italian dressing evenly into salad, enjoy!]
tit[Helen's easy salad]
```

In all of these examples you can see either the title, ingredient(s) or instruction(s) are specified then followed by opening and closing square brackets with content inside. When specifying, the title will be labeled with abbreviation tit, ingredient with ing, instruction with ins. To add a program that contains more than one ingredient or instruction under one label (ing [], ins []), add a comma after each content item. Also note this means the instruction can not have a comma in it.

Place your programs in a text file and run by inputting:

```
dotnet run <example>.txt
```

You can run each of these examples by replacing the corresponding .txt file.

Language Concepts

The user needs to understand the basic format of title, ingredients and instructions. They need to know how to format the code so it reads in—when to use commas versus new lines, when to use parentheses, etc. In terms of primitives, we have the title, ingredients, and instructions. Our language combines these primitives into a full recipe. For specifying titles, only the title specified latest will be taken in as the recipe name.

Formal Syntax

```
<expr> ::= <Title> | <Ingredients> | <Instructions> | <expr><expr>
<Instructions> ::= <Instruction>, <Instruction> | <Instruction>
<Ingredients> ::= <Ingredient>, <Ingredient> | <Ingredient>
<Title> ::= "name of recipe"
<Ingredient> ::= "Ingredient"
<Instruction> ::= "Instruction"
<Recipe> ::= <expr> | List of <expr>
```

Semantics

Syntax	Type	Meaning
Title	string	a primitive that represents the title of a recipe
Ingredient	string	a primitive that represents an ingredient or set of ingredients
Instruction	string	a primitive that represents an instruction or set of instructions in a recipe
Ingredients	string list	an ingredient list
Instructions	string list	an instruction list
Recipe	string, string list, string list	a recipe containing a title, list of ingredients, and list of instructions

0.1 Remaining Work

To further enhance this language we could make the latex recipe template more personalizable. Like say the user would like to have a certain font or color. Maybe even going further and allowing them to choose from different templates. We played with the idea of letting them create a cookbook. In prior drafts we said this could be done by adding another label of a title, but we changed our language so users could correct the title if they misspecified it by adding another title. We mentioned providing an overall time for the recipe and ultimately decided against this though because that would mean requiring users to enter in time frames for the instructions. This kind of goes against our thought of writing an easy fast program as users would have to calculate the amount of time something took them. Not fun for when you are experimenting. Lastly, something that could make our language better was if users were able to include commas and square brackets in their instructions or ingredients.