Team Cookpy:

Jacob Beneski
Mark Bernhardt
Andrew Thomson

Cookpy is a loosely cooking themed esolang developed in Python, and is pronounced like 'Cookie'

Cookpy is useful for chaining math functions in an interesting way, with variable support.

Usage for Cookpy:

Cookpy is stack based, so items are added to the stack first, then acted upon.

For example to add two values:

```
1 2 mix
```

will push 3 onto the stack.

To take it one step more:

```
1 2 mix serve
```

will push 3, then pop and print it.

You can chain commands together to do interesting math functions.

```
5 10 mix 500 cure 3 skim serve
```

will print 7497.

Additional math functions are listed below.

You can also store values as variables. Only global variables exist for simplicity.
Variables also start with a '$' to differentiate them from other words, or comments

To store a variable use the pack keyword:

$variable-name value pack

$var1 5 pack

will put 5 into the variable $var1.

To recall a variable use the 'shop' keyword:

$var1 shop

will put 5 onto the stack.

You can reassign a variable the same way as assigning a new variable

You can also use the trash keyword to delete a variable

$var1 trash

will delete the variable.

Finally order can be used to read in a value from console:

$input order pack

will save the value from console as $input

To run cookpy code, execute:

python cookpy.py *sourcefile*

Valid Words are:

x and y are two valid numbers inline, or on the stack

'mix': add two values from the stack. Usage: x y mix
x+y is put on the stack
'skim': subtract two values from the stack. Usage: x y skim
x-y is put on the stack
'serve': pop and print the top of the stack. Usage: x y mix serve
x+y is printed
'popcorn': pop and print all items from the stack. Useful for debugging.
Usage: popcorn **The stack is printed**
'clarify': peek the top of the stack. Usage: clarify
Peeks the top of the stack
'measure': print the length of the stack. Usage: measure
Prints the length of the stack
'dust': divide two values from the stack. Usage: x y dust
x/y is added to the stack
'grill': floor divide two values from the stack Usage: x y grill
x/y is added to the stack using int division
'dissolve': get remainder of two values from the stack. Usage: x y dissolve
The remainder of x/y is added to the stack
'cure': multiply two values from the stack. Usage: x y cure
x * y is added to the stack
'grind': deletes the top value from the stack. Usage: grind
**deletes the top value from the stack

'pack': store the value of a variable Usage: $variable x pack
Stores x in $variable
'shop': get the value of a variable onto the stack Usage: $variable shop
Pushes the value of $variable onto the stack
'trash': delete the variable Usage: $variable trash
Deletes the variable
'order': read input from the console Usage: x order
The value from order will be pushed on the stack as x

Variable names must start with a '$' and can contain A-Z, a-z, . or -

At the moment only numerical operations are supported, but order can take in and serve can print strings.

Boolean, and loops are not supported at the moment.

Syntax errors raise an assertion error if not enough items are on the stack, or will raise a standard usage error if an invalid word is tried.
Helpful sources:

http://scratch-lang.notimetoplay.org/scratch-lang.html

https://regex101.com/

https://www.d.umn.edu/~alphanu/cookery/glossary_cooking.html

https://stackoverflow.com/questions/2052390/manually-raising-throwing-an-exception-in-python

https://stackoverflow.com/questions/308122/simple-regular-expression-for-a-decimal-with-a-precision-of-2

https://stackoverflow.com/questions/6416131/python-add-new-item-to-dictionary

https://www.pydanny.com/why-doesnt-python-have-switch-case.html

and lots of other stack overflow questions that got lost in the process. They helped with various regex errors, python syntax and other issues.