"Hey, IRIs"
Alfred Workflows

Simple To-Do List

1. This object sets the keyword you use to start adding a new task to your to-do list. Type "todo" followed by your task (e.g. "todo bake a cake").

2. The Arg/Var utility moves your task (the query) to a variable called "todo" until it's needed later in the green path.

3. The List Filter input lets you select (from the options you set in the object) which to-do list to add the task to. (Following the green path)

4. Using "todo" keyword and selecting a list opens the specified to-do list instead. (Following the blue path and there is no "task" variable)

5. This filter is set so that if there is no to-do text, it should follow this blue path & open the appropriate list.

6. The Append To File output appends the to-do text (moved into a variable called "todo") in the orange object to the appropriate list.

7. The Post Notification output lets you know that the task has been added through the Notification Center.

8. This filter is set so that if there is no to-do text, it should follow this blue path & open the appropriate list.

9. The Open File action opens the file specified in the yellow List Filter object.
Team Roles

Project Environment
SYNTAX
<table>
<thead>
<tr>
<th>TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
</tr>
<tr>
<td>float</td>
</tr>
<tr>
<td>string</td>
</tr>
<tr>
<td>bool</td>
</tr>
</tbody>
</table>
variable1

|      |

variable2
tuple

| function
int plus2(int a)
    return a + 2
Siri

int plus1(int a)
    return a + 1
Siri

int main()

0
|
plus1
|
plus2
|
printi
Siri
Pipe: "Overload"
Semant + LLVM
Semant: Sassign/SCall
List: C-based Implementation
String: Heap-based Implementation
Tuple: LLVM-based Implementation
Future Works
TESTING
Testing

- Identifiers
- Keywords
- Control flow: if, while
- Types: int, float, string, list, tuple
- Built-in functions: inputint, inputfloat, inputstring, sendmail
- Comments
- Operators
- Variables and function declarations

<table>
<thead>
<tr>
<th>Pipe</th>
<th>1 2 3</th>
<th>a</th>
<th>1</th>
<th>printi(1)</th>
<th>1,2,3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>a, b, c</td>
<td></td>
<td>printi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Test Suites

- Two test suites, one intended to pass, and one intended to fail.
- `test_*`, `fail_*` -> `.out`, `.err`
- Over 80 tests in total
Demo

• Calculate Tips

• Binary Search

• Send Email