

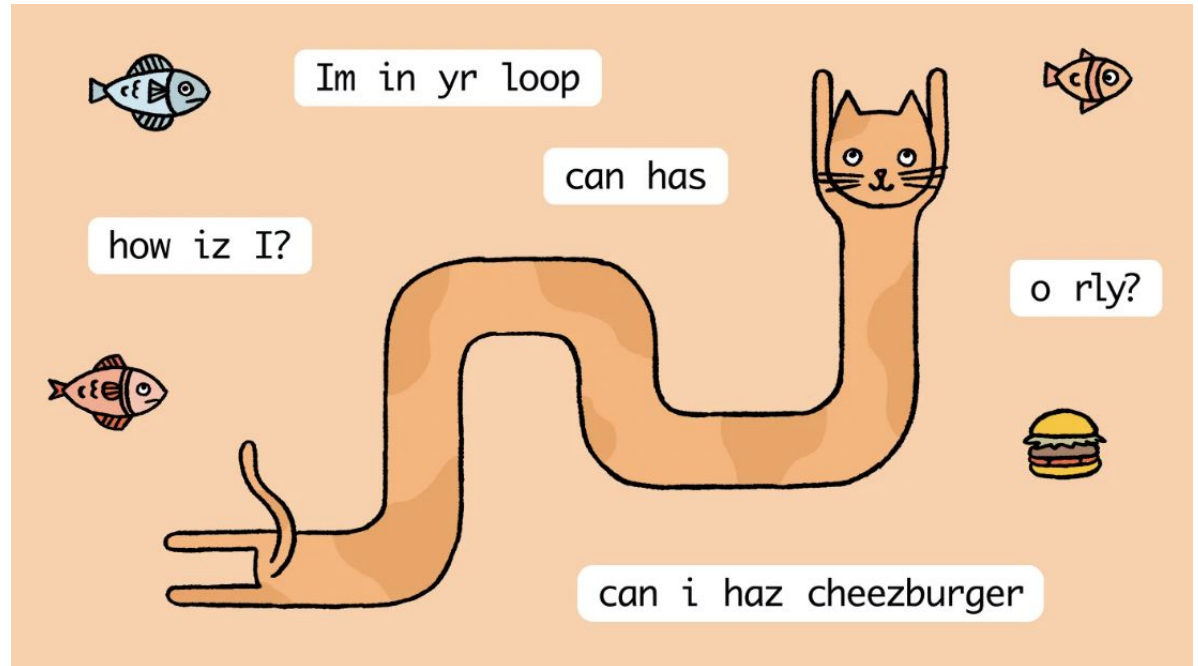
LOLCODE → PYTHON TRANSLATOR

Adrian Turkedjiev & Helen Mancini

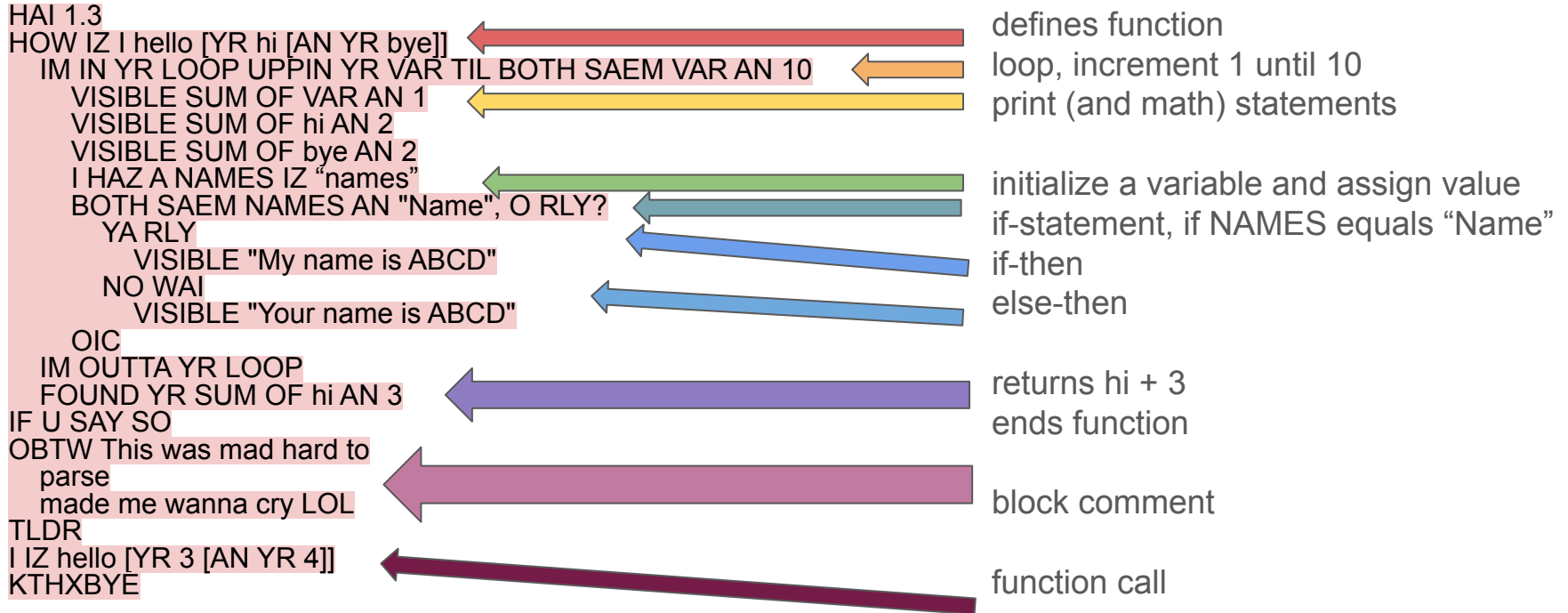
What is LOLCODE?

“A programming language for cat lovers”

Based off internet memes



Example LOLCODE

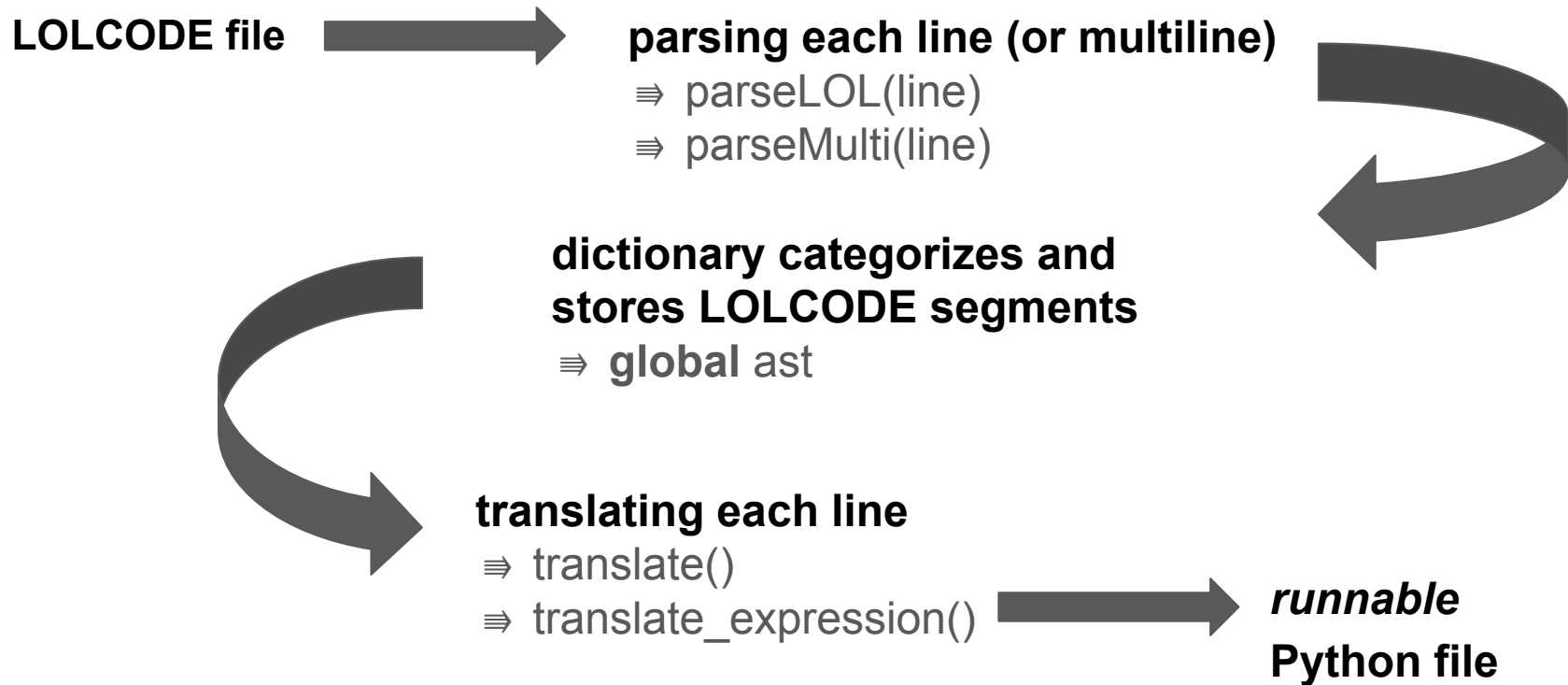


Why LOLCODE?



Translating to Python!

How it works



Parsing

Each type has its own dictionary format

```
TYPES:
declare
  "arguments" : ["variable_name"]
deallocate
  "arguments" : ["variable_name"]
instantiate
  "arguments" : ["variable_name", "assigned_type"]
assign
  "arguments" : ["variable_name", "assigned_value"]
math
  "arguments" : ["operator", "argument_1", "argument_2"]
print
  "arguments" : ["value"]
boolean
  "arguments" : ["operator", "argument_1", "argument_2"]
```

```
comment
  "arguments" : ["content"]
conditional
  "arguments" : ["operator", "argument_1", "argument_2", "content"]
if-else
  "arguments" : ["if_or_else", "content"]
loop
  "arguments" : ["operation", "variable", "limit", "content"]
function
  "arguments" : ["def_or_call", "function_name", "argument_1", "argument_2", "content"]
return
  "arguments" : ["value"]
literal
  "arguments" : ["value"]
```

VISIBLE SUM OF Y AN X

```
elif "VISIBLE" in bare_content:
    tokens = line.split()
    value = parseLOL(" ".join(tokens[1:]))
    latest_id = calc_id()
    ret = {
        "id" : latest_id,
        "original" : line,
        "type" : "print",
        "arguments" : [value],
    }
```

Parses and classifies SUM OF statement through recursion.

```
elif "SUM OF" in bare_content:
    tokens = line.split()
    operator = "+"
    arg1 = tokens[2]
    arg2 = tokens[4]
    latest_id = calc_id()
    ret = {
        "id" : latest_id,
        "original" : l,
        "type" : "math",
        "arguments" : [operator, arg1, arg2],
    }
```


Parsing Multiple Lines *ei functions, loops, if-statements*

- Keeping track of indentation levels through `calc_id()`
 - *`new_id = round(latest_id + (1 * 10** -latest_scope), int(latest_scope))`*

LOLCODE

```
HOW IZ I <function_name> [YR <parameter> [AN YR <parameter>]]  
    <code, could be more nested multilines>  
FOUND YR <result statement>  
IF U SAY SO
```

Python

```
def <function_name>(parameter, parameter)  
    <code, could be more nested multilines>  
    return <result statement>
```

Translating

- Substituting values into defined structure
- Recursive for nested dictionaries
- Checks if you are at start of new line; adds indentation

```
ret += "    " * (len(str(dictionary["id"]).strip(".0")) - 1)
```

Process: Resources & Limitations

- Parsed based on [LOLCODE 1.3](#) specs (by LOLCODE lorde Justin Meza)
- Many online LOLCODE compilers compile older versions, including [this](#) popular one, so we checked our work by writing our own LOLCODE.
- We limited argument amounts based on dictionary methodology (no infinite entries)

*Embarrassed
cat →*



DEMOS

KTHXBYE

