

Making Esoteric Beauty Accessible: Befunge in 2025

Robin Brown (she/her)

Outline

1. Why Befunge?
2. What is Befunge?
3. Key Concepts
 - a. Program Flow (Grid)
 - b. Data Flow (Stack)
4. Instructions
5. Demos
6. Extending Befunge?

Why Befunge?

Why talk about and work on Befunge?

Befunge is beautiful.

I want to give more people
that "ah ha!" moment.

It may help people
understand other things
(e.g. Java bytecode, Wasm)

Sometimes it's fun to put
way more effort into things
than you can justify.



What is Befunge?

Basics

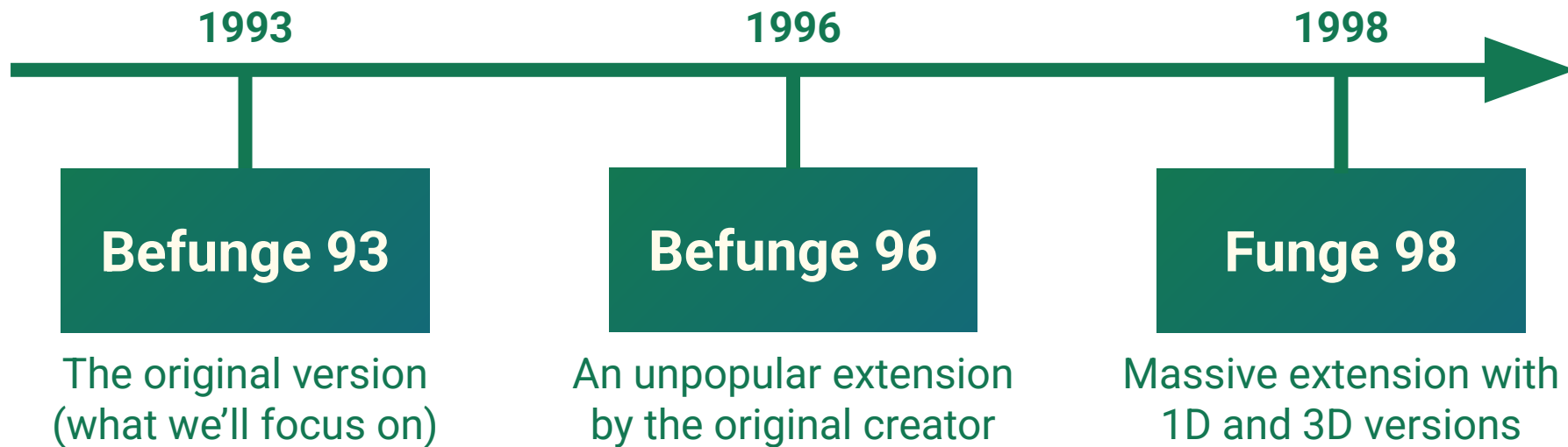
An **esoteric**
programming language

Designed to be effectively
impossible to compile

A very “**small**” **language** with
few features or instructions

Generally very **confusing**
to write programs in

Release History



Name History

- **Real Answer**
 - The name was inspired by a misspelling of "before"
- **Accepted Retcon Answer**

Be-

as in the prefix "bi-" meaning two
because it is 2-dimensional

funge

as in fungible because it treats
code and data as interchangeable

Key Concepts

Program Flow

Line-oriented

```
p = "who are you?"
```

```
name = input(p)
```

```
msg = f"Hi {name}!"
```

```
print(msg)
```

Python code

Grid-oriented

"	o	l	l	"	v
v	"	h	e	"	<
>	:	v			
^	,	-	@		

Befunge code

Program Flow

Line-oriented

```
p = "who are you?"
```

```
name = input(p)
```

```
msg = f"Hi {name}!"
```

```
print(msg)
```

lines made of tokens

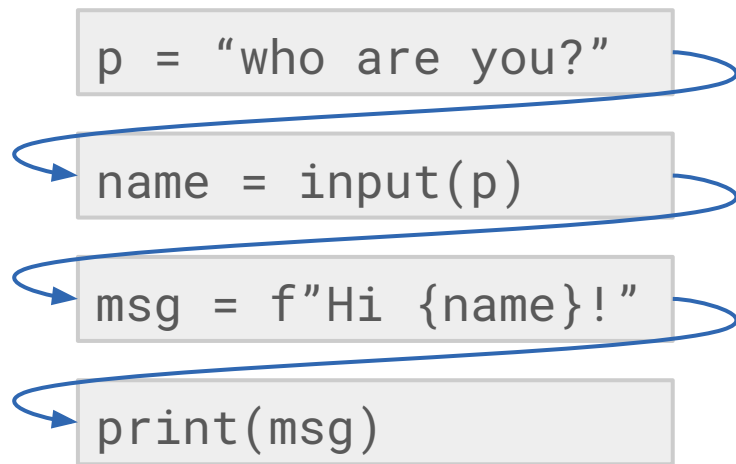
Grid-oriented

"	o	l	l	"	v
v	"	h	e	"	<
>	:	v			
^	,	-	@		

each character is an instruction!!

Program Flow

Line-oriented



(before → after)

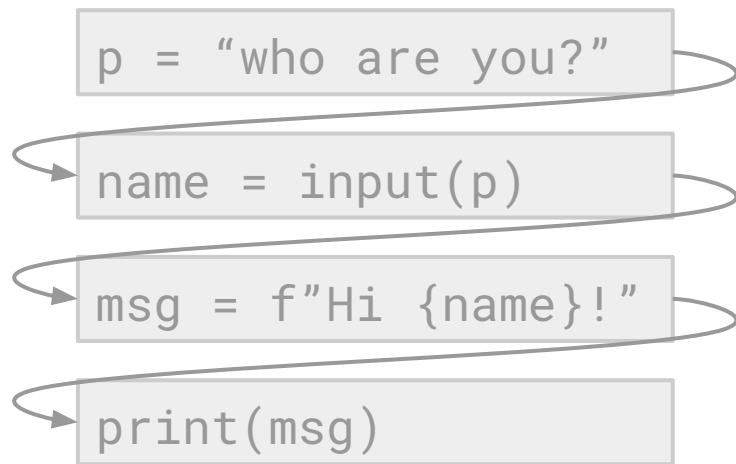
Grid-oriented



each character is an instruction!!

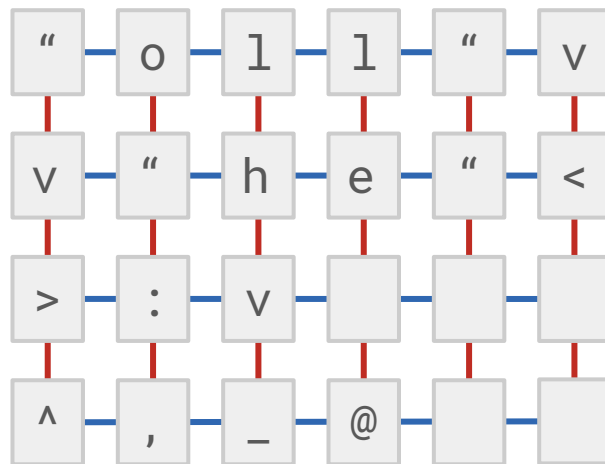
Program Flow

Line-oriented



(before→after)

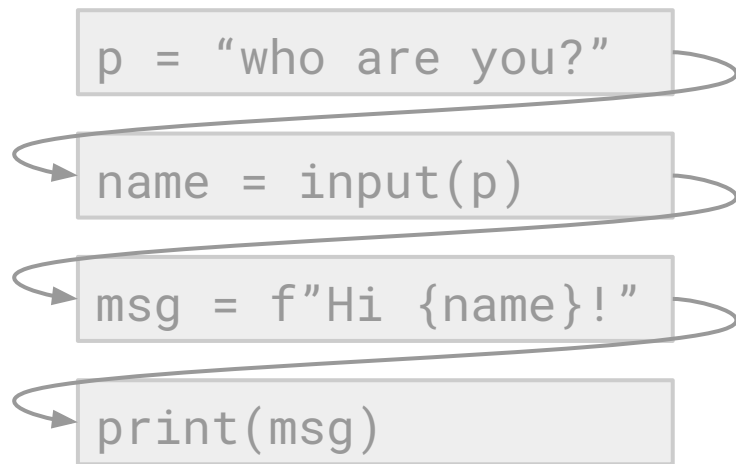
Grid-oriented



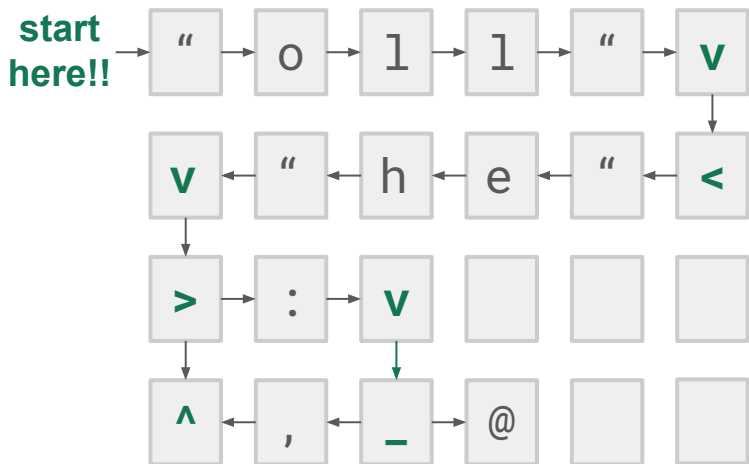
(left – right, above – below)

Program Flow

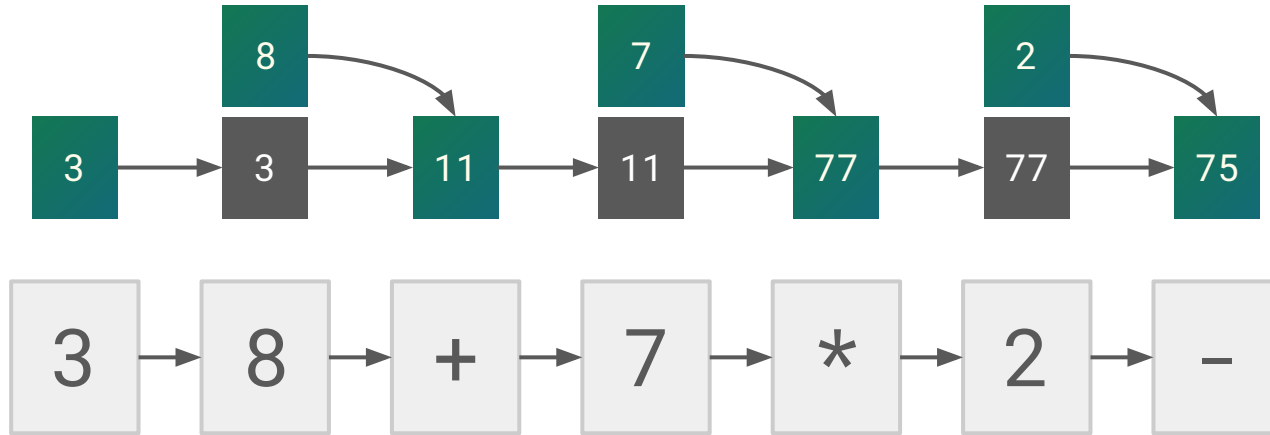
Line-oriented



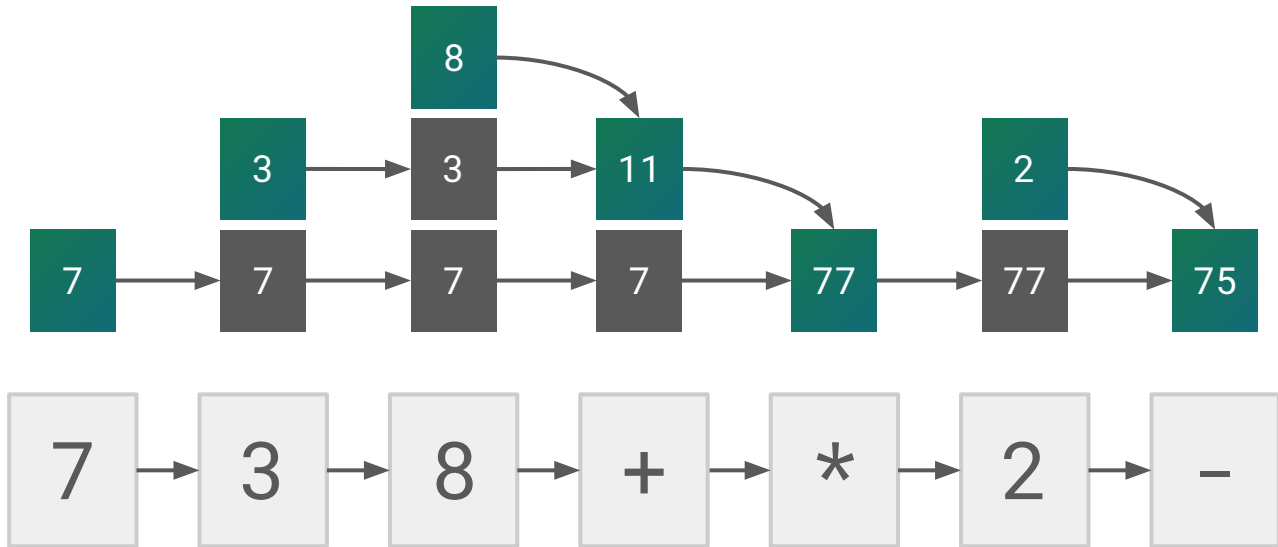
Grid-oriented



Data Flow



Data Flow

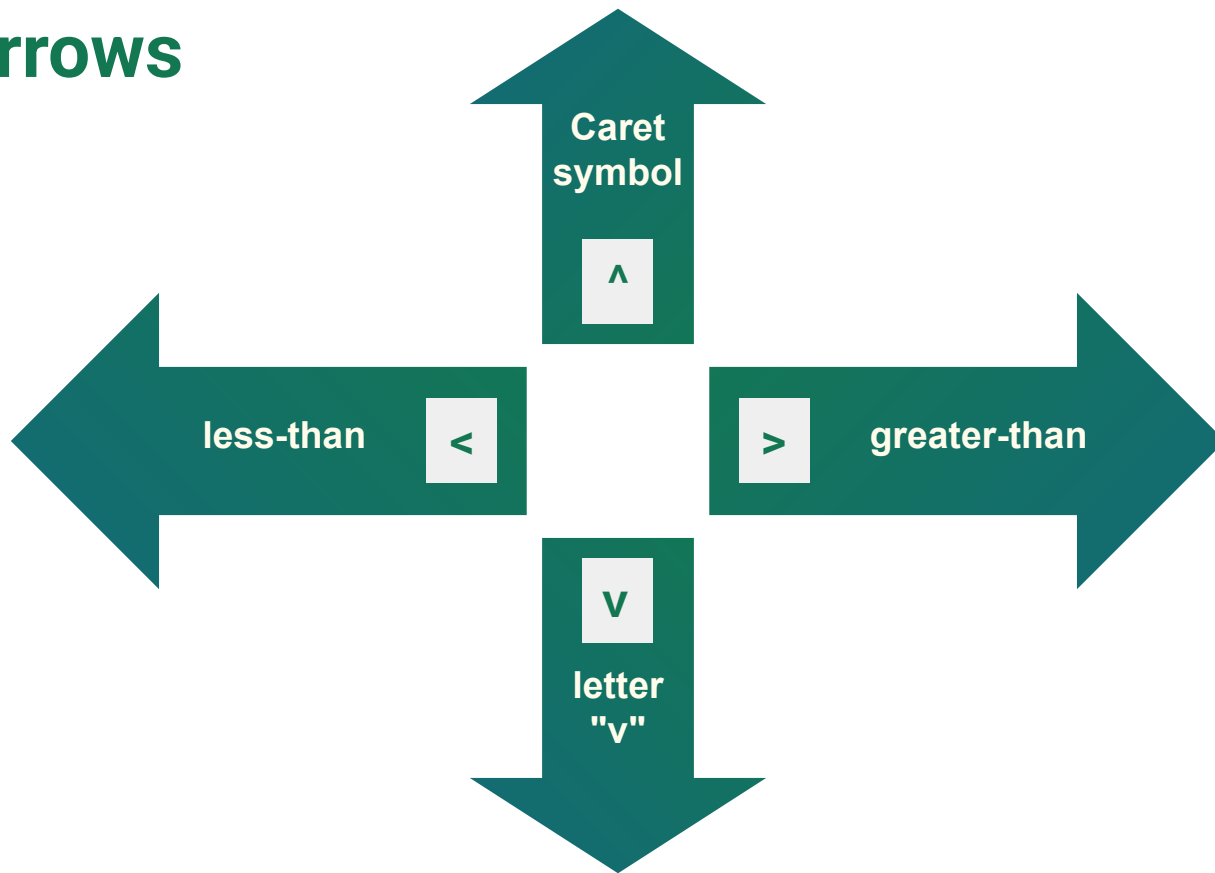


Demo: Program & data flow



Instructions

Basic Arrows



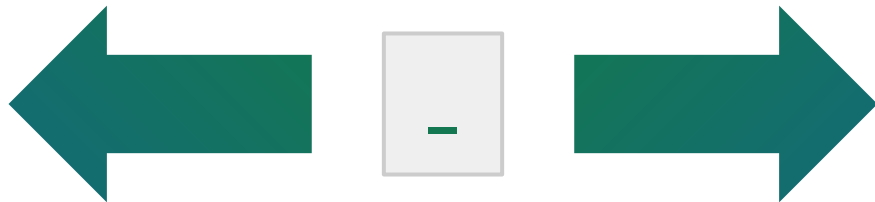
Decimal Digit Literals



Push the corresponding numeric value onto the stack

Underscore

Pops a value off the stack then...



Goes **left** if the **value != 0**

Goes **right** if the **value == 0**

Pipe

Pops a value off the stack then...

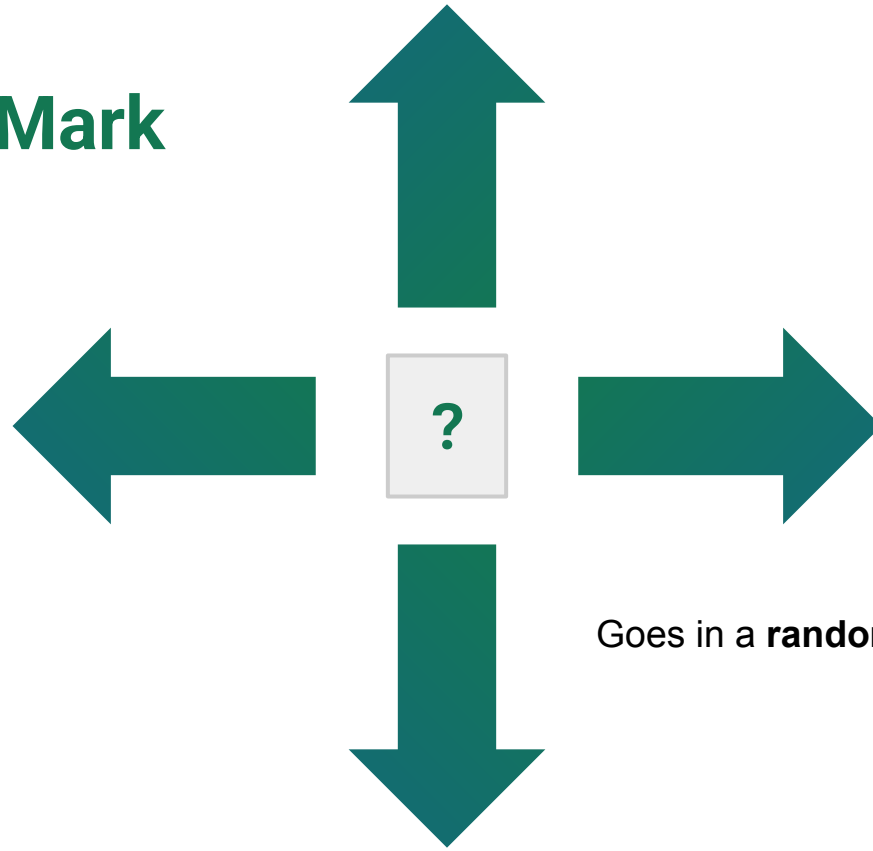


Goes **up** if
the **value != 0**



Goes **down** if
the **value == 0**

Question Mark

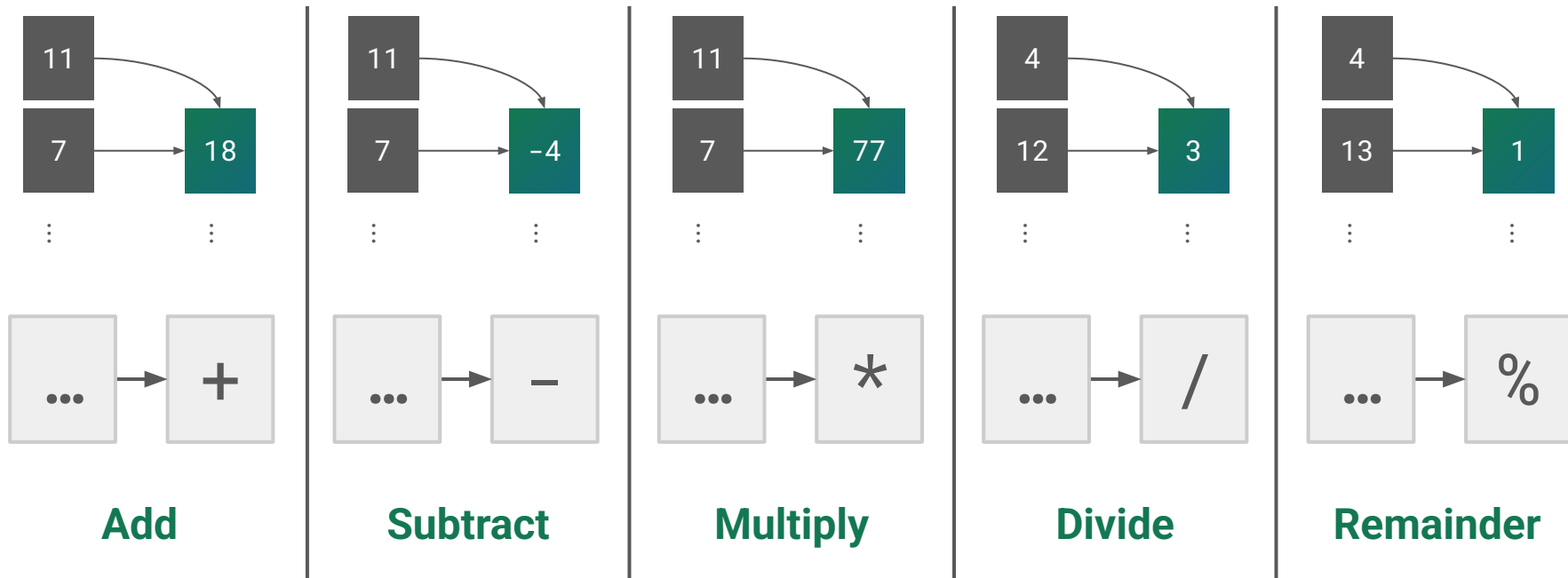


Goes in a **random direction!!!**

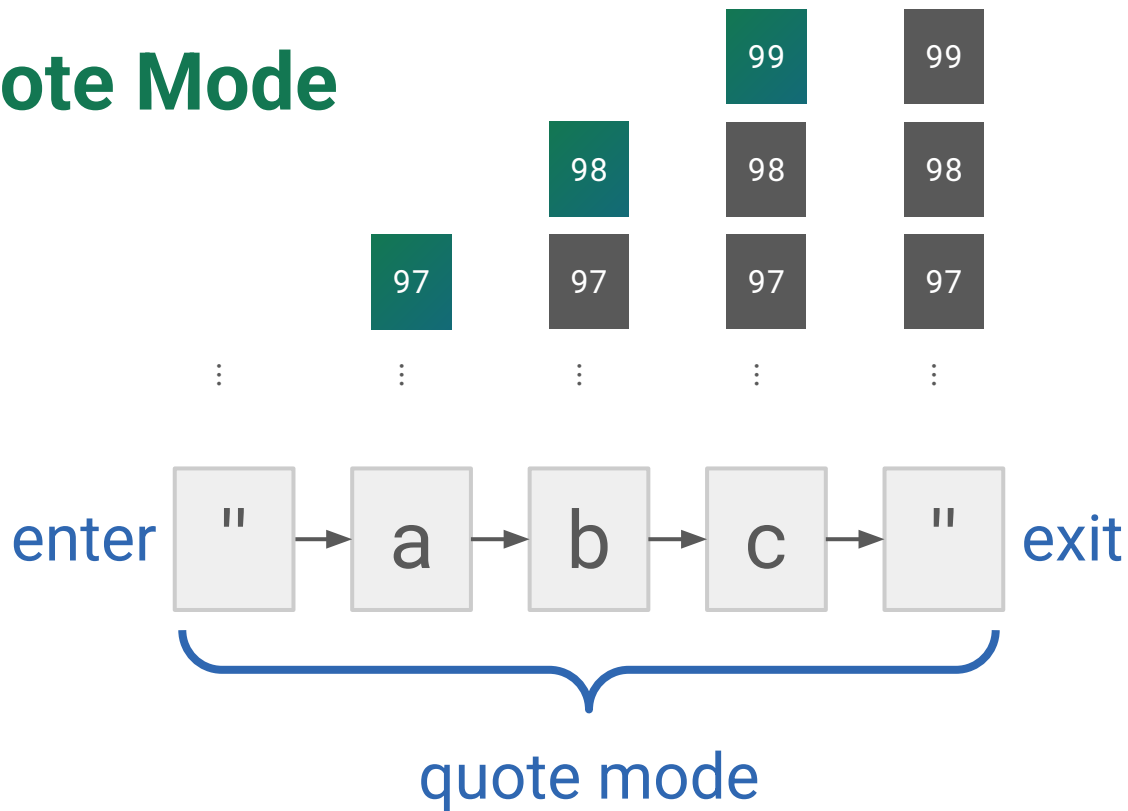
Demo: Instructions (part 1)



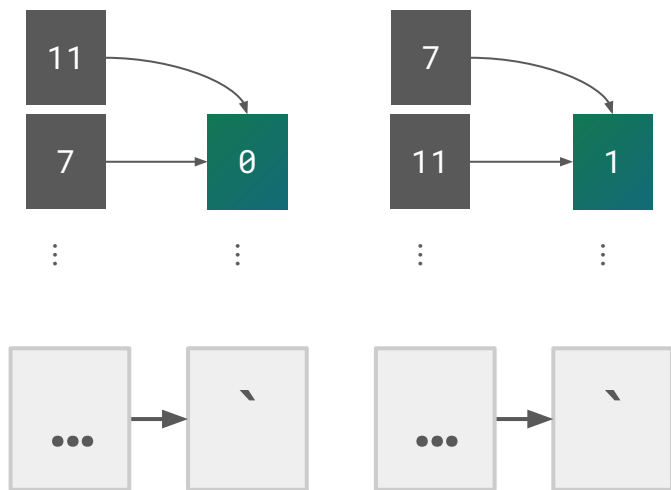
Arithmetic



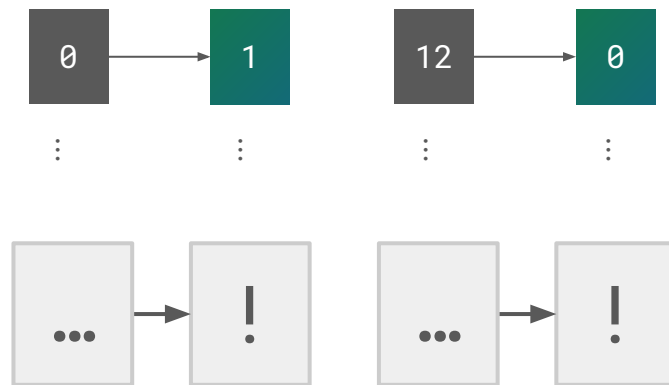
Quote Mode



Logic and Comparison



Greater Than



Logical NOT

Standard I/O

Read

Write

ASCII

~

,

Integers

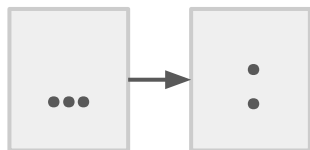
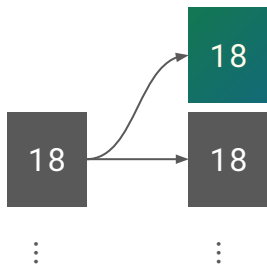
&

.

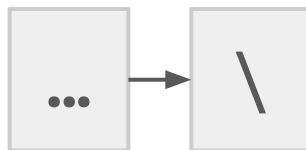
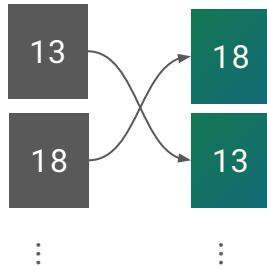
Demo: Instructions (part 2)



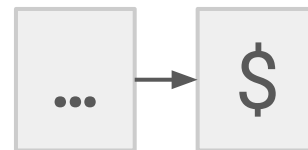
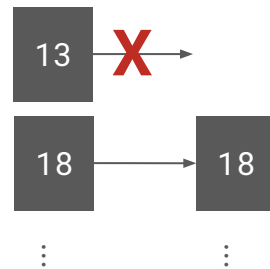
Stack Manipulation



Duplicate
(Colon)



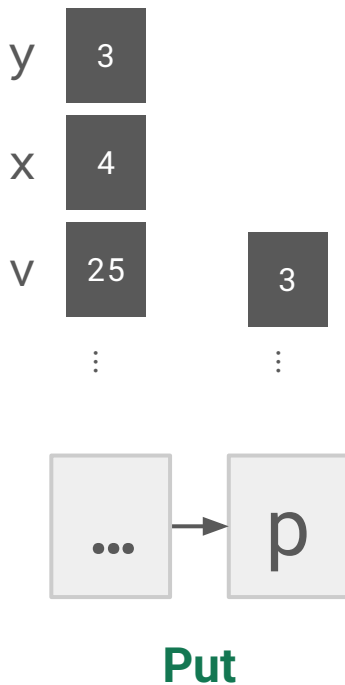
Swap
(Slash)



Delete
(Dollar Sign)

Grid Manipulation

Pop three values,
update the cell
at position (**x**, **y**)
to the value **v**



Pop two values
lookup the value
at position (**x**, **y**)

Demo: Instructions (part 3)



Web Demo

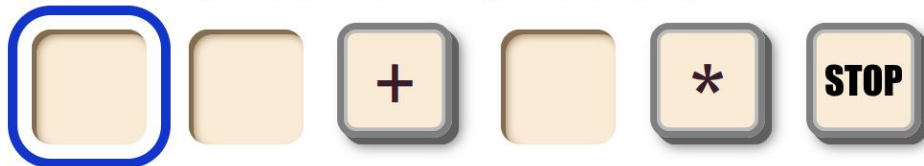
Demo: Befunge Tools - Web

Interactive Befunge Test

This is a test/demo of an interactive drag-and-drop environment for Befunge.

Puzzle 1

Drag tiles onto the program grid so that it computes the number 77!



Your Supply of Tiles!



Extending Befunge

Core Themes of Befunge

1. Programs are 2D grids with toroidal topology
2. The program counter is 2D and moves short relative distances
3. Every cell in the grid is an instruction
4. Instructions operands and results use a stack
5. Programs can modify themselves while running

Extending Befunge: My Thoughts

- **Start with Befunge 93** and stick to 2D
- Update it to use **32-bit everywhere** and **Unicode**
- Identify and improve pain points and awkward patterns
- Consider both
 - **Dialects** that retain the core characteristics
 - **Derivatives** that make significant departures
- Whatever you do, please write a decent spec!!!

Thanks for attending!

github.com/**esoterra**

@**esoterra** @hachyderm.io

esoterra.dev

Talk info available at <https://esoterra.dev/talks/2025-fossy>

