

RPN Calculator in Python 3

Notation

- Ways of writing an expression
- How operators and operands are placed in an expression

operator



2

+

3



operand

operand

$$2 + 3$$

Infix Notation

- Operator is placed between operands

$$2 + 3 * 4$$

Infix Notation

- In absence of precedence rules
- Order of operations is ambiguous
- 14 or 20 depending on order

$$(2 + 3) * 4$$

Infix Notation

- Requires parentheses to remove ambiguity

2 3 +

Postfix Notation

- Operator follows all its operands
- Also called Reverse Polish Notation (RPN)
- Invented in the 1920
- Value to computer science recognized 1950-60s
 - Works in a way similar to how computers execute expressions

infix

2 + 3 * 4

postfix
(RPN)

2 3 4 * +

2 3 4 * +

$$2 \quad \boxed{3 \quad 4 \quad * \quad +}$$

$$3 \quad 4 \quad * \quad = \quad 12$$

$$2 \quad \boxed{3 \quad 4 \quad * \quad +}$$

$$3 \quad 4 \quad * \quad = \quad 12$$

$$2 \quad 12 \quad +$$

$$2 \quad \boxed{3 \quad 4 \quad *} \quad +$$

$$3 \quad 4 \quad * \quad = \quad 12$$

$$\boxed{2 \quad 12 \quad +}$$

$$2 \quad 12 \quad + \quad = \quad 14$$

$$2 \quad \boxed{3 \quad 4 \quad * \quad +}$$

$$3 \quad 4 \quad * \quad = \quad 12$$

$$\boxed{2 \quad 12 \quad +}$$

$$2 \quad 12 \quad + \quad = \quad 14$$

14

infix

2 + 3 * 4

(2 + 3) * 4

postfix
(RPN)

2 3 4 * +

2 3 + 4 *

postfix notation requires NO parentheses

Stack

Wikipedia says:

In computer science, a stack is an abstract data type that serves as a collection of elements, with two principal operations: push, which adds an element to the collection, and pop, which removes the most recently added element that was not yet removed.

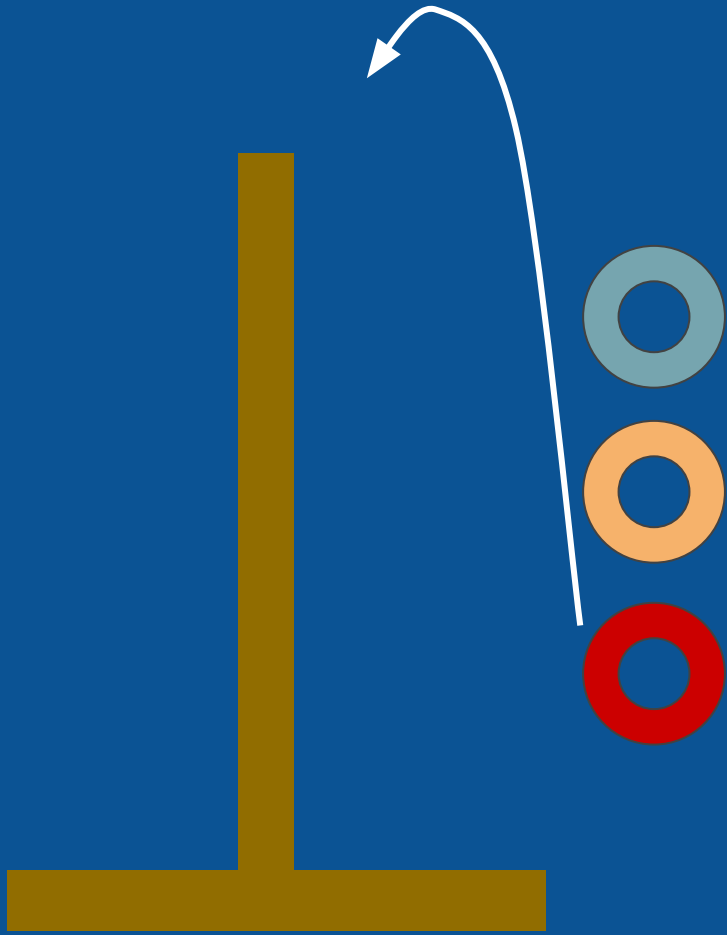
Stack

Wikipedia says:

In computer science, a stack is a collection of elements, ordered from top to bottom, to which you can add elements to the collection, and remove elements from the collection, but you can only remove elements that were not yet removed.

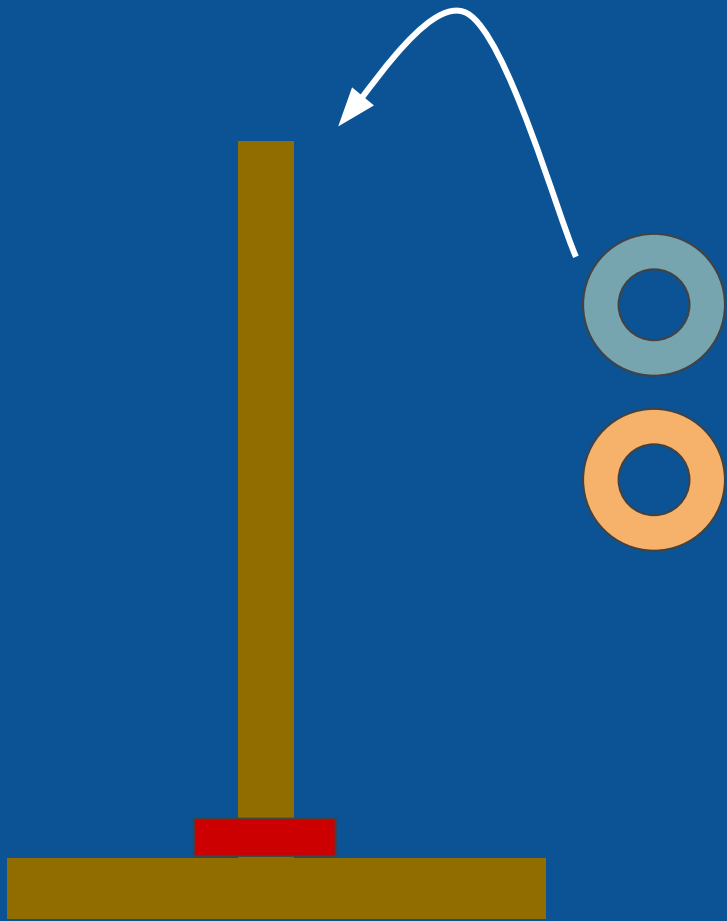


serves as a
which adds an element
ded element that



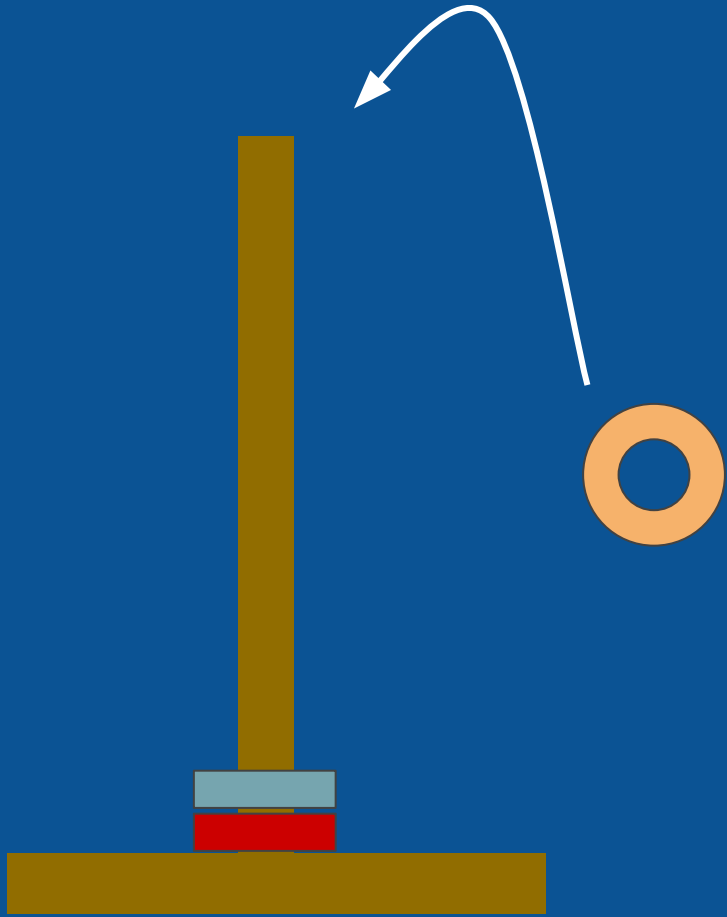
push



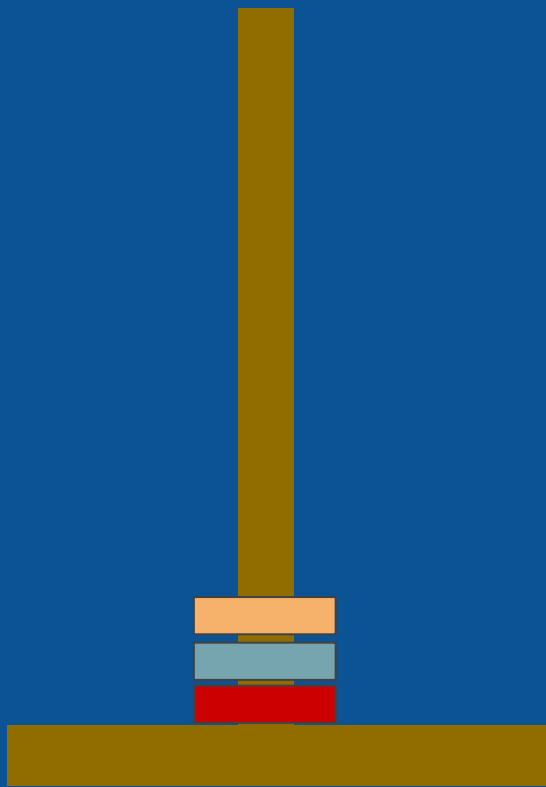


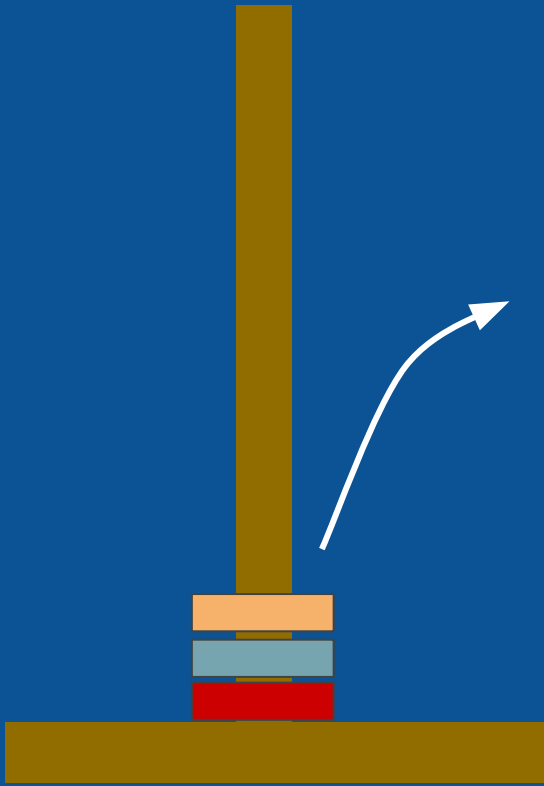
push



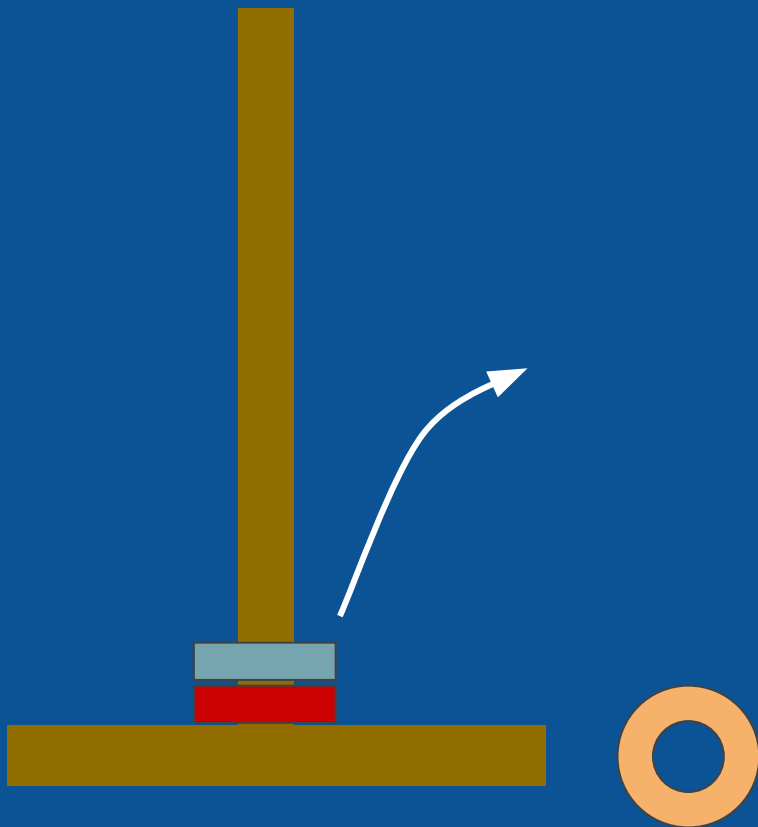


push ○

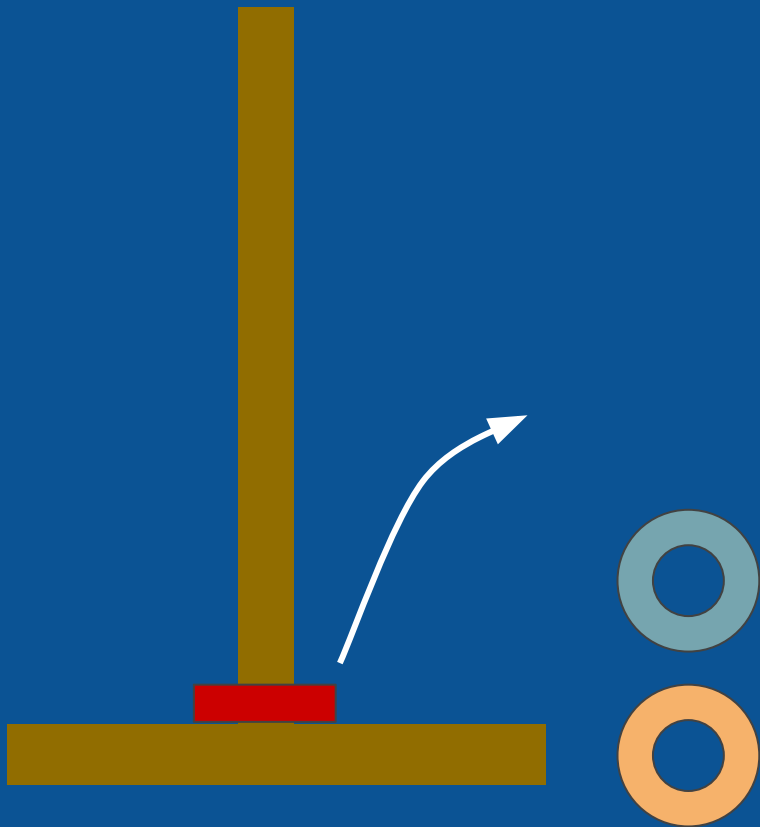




pop



pop



pop



pop

Stacks and postfix notation

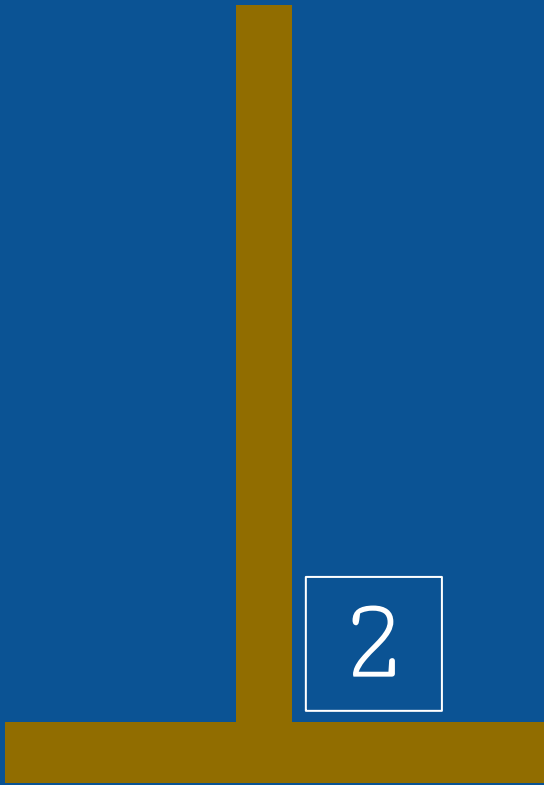
- Stacks are the natural way to evaluate postfix expression



2 3 4 * +

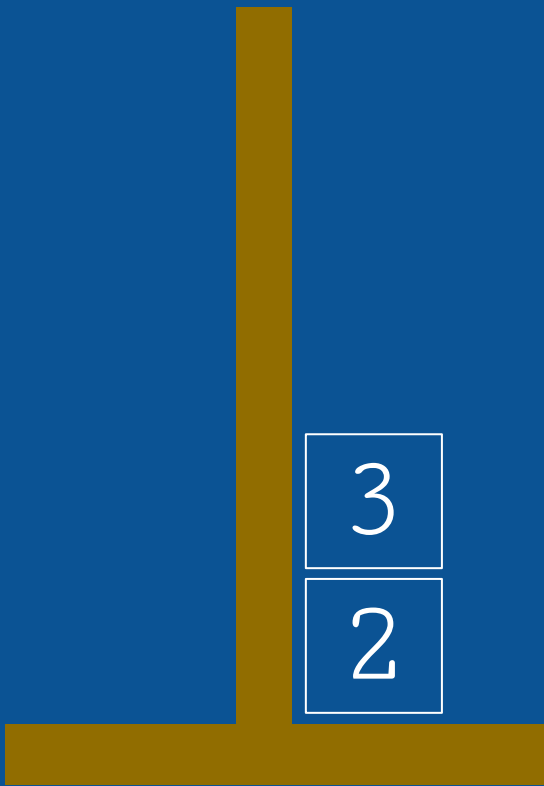


push



2 3 4 * +
↑

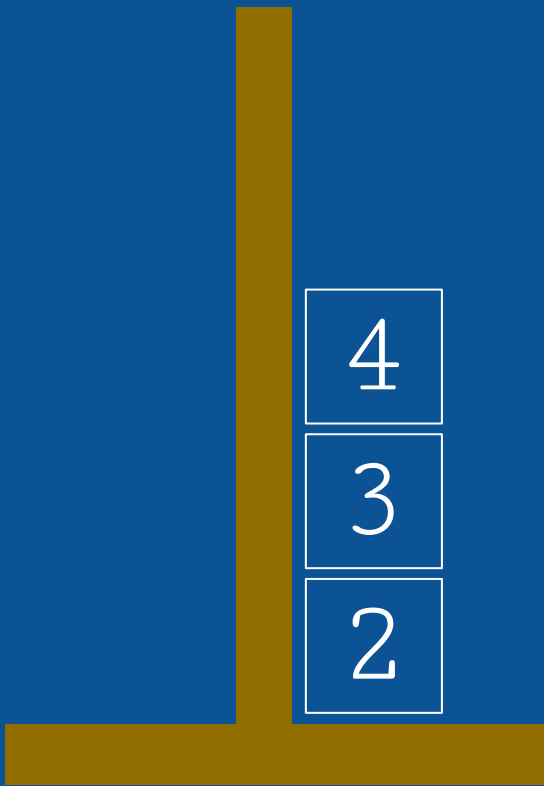
push



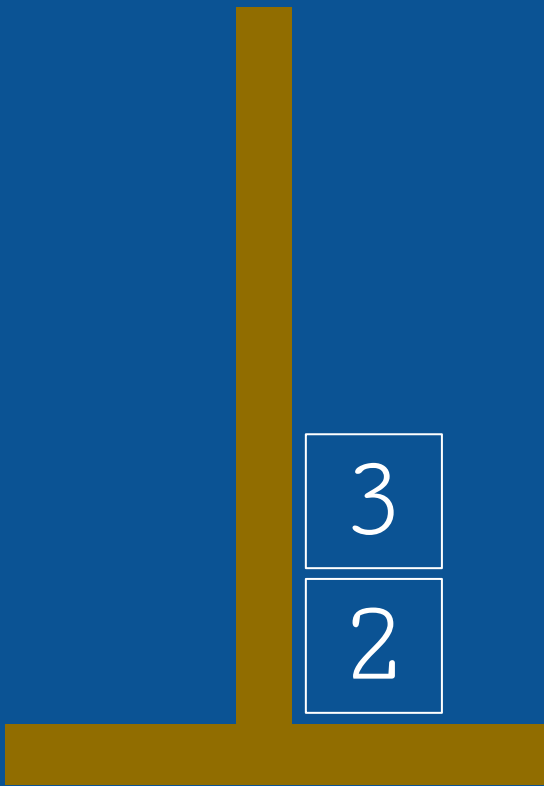
2 3 4 * +



push



2 3 4 * +
↑

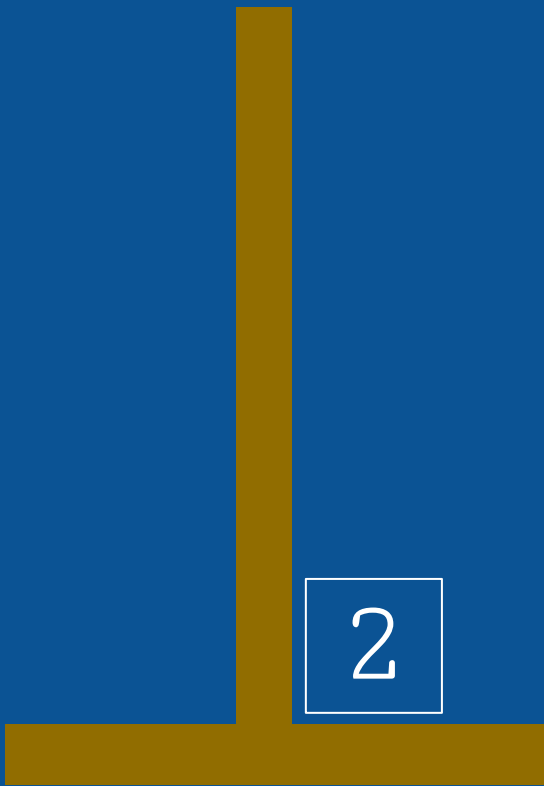


2 3 4 * +



4

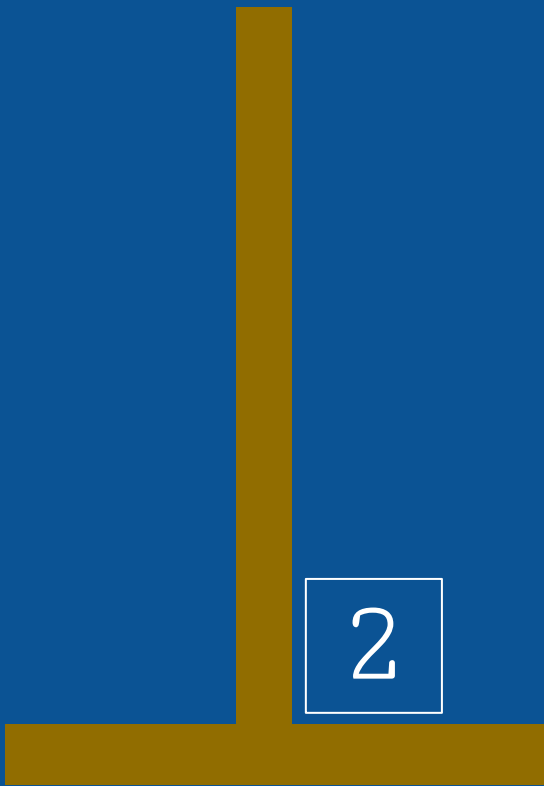
pop



2 3 4 * +

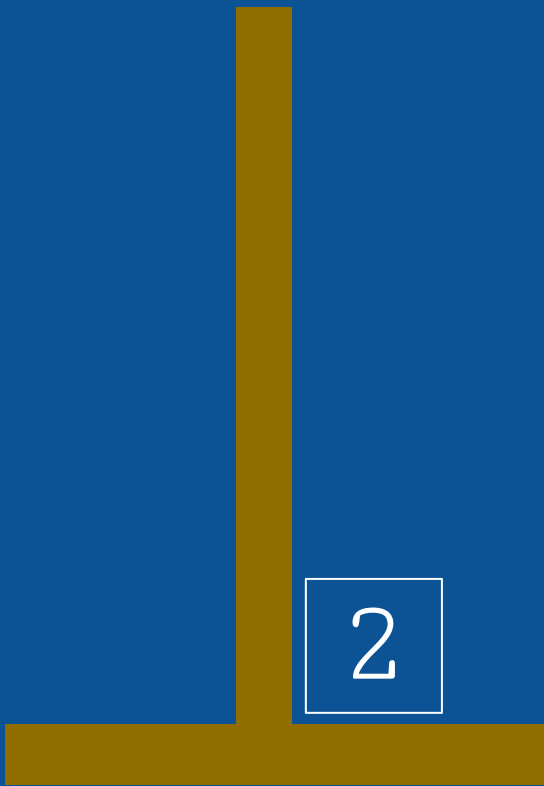


pop



2 3 4 * +



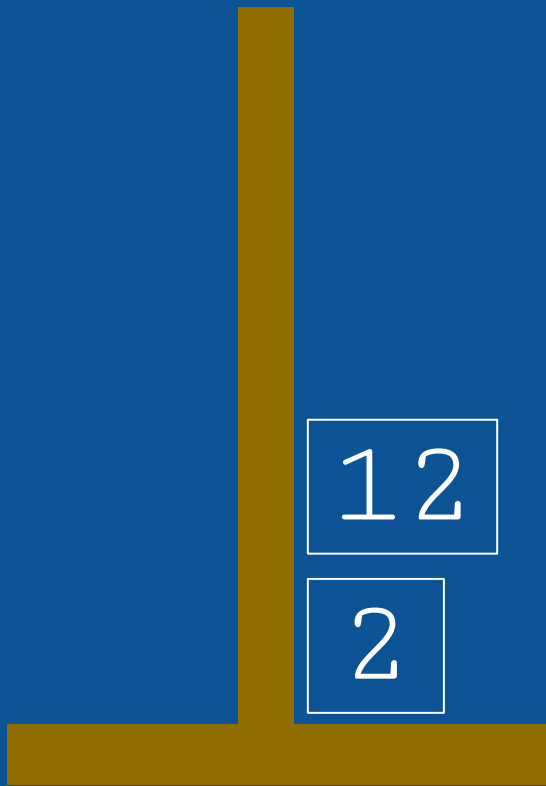


2 3 4 * +

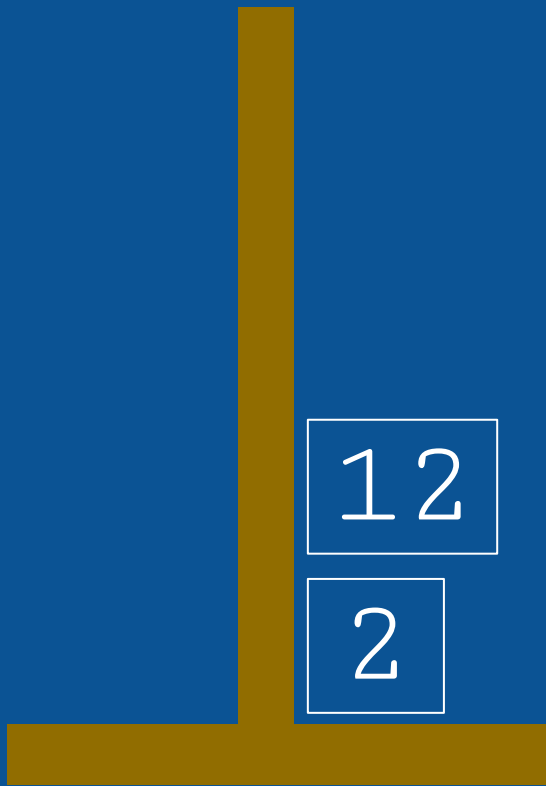


12

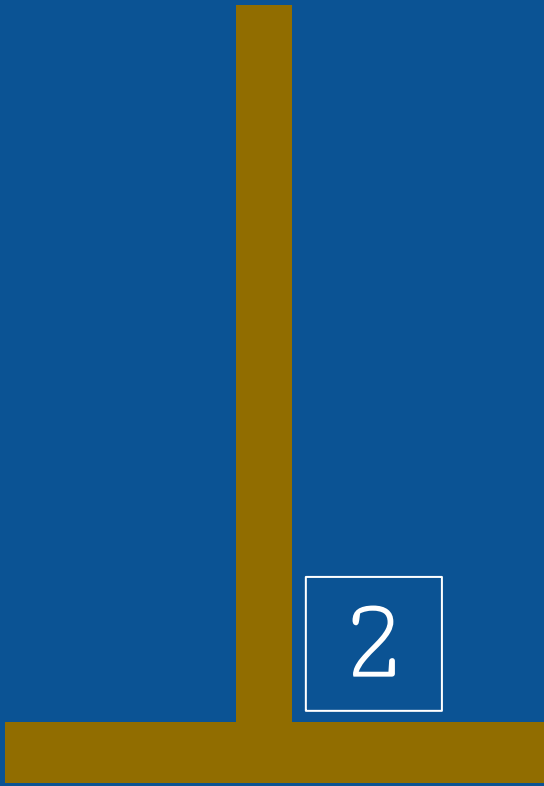
push



2 3 4 * +
↑



2 3 4 * +
↑



2 3 4 * +

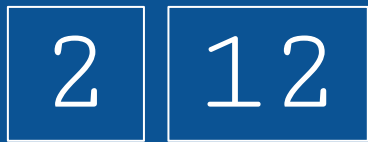
12

pop





2 3 4 * +



pop





2 3 4 * +





2 3 4 * +



14

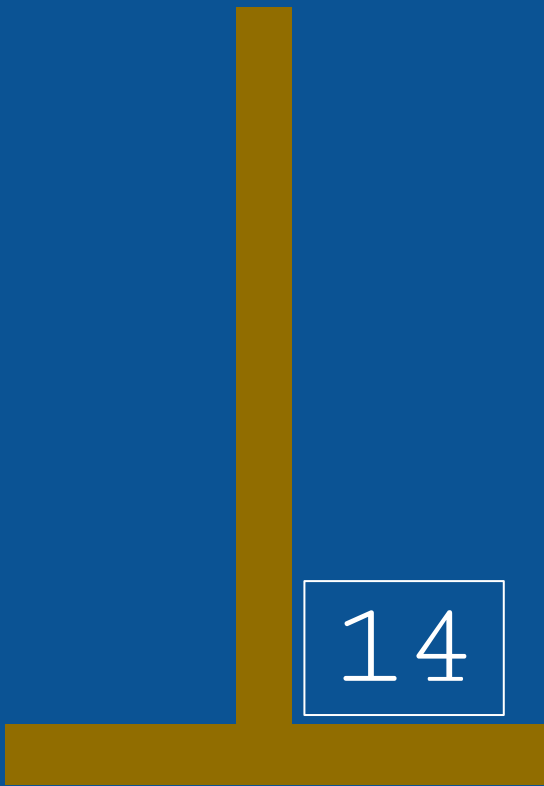


2 3 4 * +

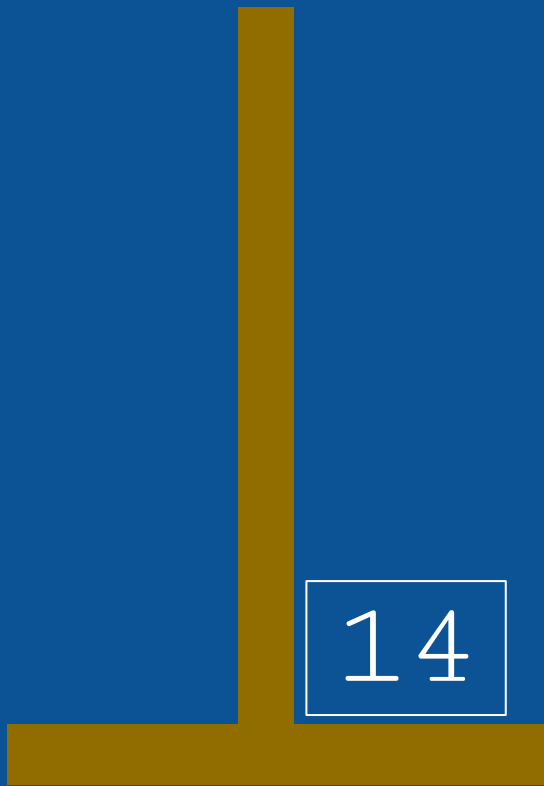
14

push





2 3 4 * +
↑



2 3 4 * +



Top of stack has the
expression value

Python `list` as a stack

- Push operation is `list.append()`
- Pop operation is `list.pop()`

```
>>>> a = []
>>>> a.append(2)
>>>> a.append(3)
>>>> a.append(4)
>>>> print(a)
[2, 3, 4]
```

```
>>>> top = a.pop()
>>>> print(top)
4
>>>> print(a)
[2, 3]
```

```
>>>> top = a.pop()
>>>> print(top)
3
```

```
>>>> top = a.pop()
>>>> print(top)
2
```

```
>>>> print(a)
[]
```

Homework

