



Daily Coding Problem #163

Problem

This problem was asked by Jane Street.

Given an arithmetic expression in [Reverse Polish Notation](#), write a program to evaluate it.

The expression is given as a list of numbers and operands. For example: `[5, 3, '+']` should return `5 + 3 = 8`.

For example, `[15, 7, 1, 1, '+', '-', '/', 3, '*', 2, 1, 1, '+', '+', '-']` should return 5, since it is equivalent to $((15 / (7 - (1 + 1))) * 3) - (2 + (1 + 1)) = 5$.

You can assume the given expression is always valid.

Solution

The way to implement Reverse Polish Notation is to use a stack. When we encounter a value, then we add it to the stack, and if we encounter an operator such as '+', '-', '*', or '/', then we pop the last two things off the stack, use them as terms on the operator, and then pop the resulting value back on the stack. At the end of the function there should only be one thing remaining on the stack, so we just return that.

```
PLUS = '+'
MINUS = '-'
TIMES = '*'
DIVIDE = '/'

OPERANDS = [PLUS, MINUS, TIMES, DIVIDE]
```

```
def rpn(expr):
    stack = []
    for val in expr:
```

```
if val in OPERANDS:
    term1, term2 = stack.pop(), stack.pop()
    if val == PLUS:
        stack.append(term1 + term2)
    elif val == MINUS:
        stack.append(term1 - term2)
    elif val == TIMES:
        stack.append(term1 * term2)
    elif val == DIVIDE:
        stack.append(term1 / term2)
else:
    stack.append(val)
return stack[0]
```

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)