Copy and paste rules — why functions?

Adam Głowacki

April 8th, 2022

123456.txt

```
Alice Green
Principal Office Assistant
Warsaw, Aleje Jerozolimskie
8900
...
```

Update the files

1. 123313: 11% salary raise
2. 123790: moved to Polarna street in Katowice
3. 123856: 10% salary raise

What to do

For each person/file:

- read the file contents
- find the right line
- replace the piece of information
- update the file contents

```
f = open('123313.txt', 'r')
lines = f.read().splitlines()
f.close()

f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

```
Alice Green
Principal Office Assistant
Warsaw, Aleje Jerozolimskie
8900
...
```

```
f = open('123313.txt', 'r')
lines = f.read().splitlines()
f.close()

old_salary = int(lines[3])
new_salary = 1.11 * old_salary
lines[3] = str(new_salary)

f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

```
f = open('123790.txt', 'r')
lines = f.read().splitlines()
f.close()

lines[2] = 'Katowice, Polarna'

f = open('123790.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

```
f = open('123313.txt', 'r')
lines = f.read().splitlines()
f.close()

old_salary = int(lines[3])
new_salary = 1.11 * old_salary
lines[3] = str(new_salary)

f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()

f = open('123790.txt', 'r')
lines = f.read().splitlines()
f.close()

lines[2] = 'Katowice, Polarna'

f = open('123790.txt', 'w')
f.write('\n'.join(lines))
f.close()

f = open('123856.txt', 'r')
lines = f.read().splitlines()
f.close()

old_salary = int(lines[3])
new_salary = 1.10 * old_salary
lines[3] = str(new_salary)

f = open('123856.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

```
f = open('123313.txt', 'r')
lines = f.read().splitlines()
f.close()

old_salary = int(lines[3])
new_salary = 1.11 * old_salary
lines[3] = str(new_salary)

f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()

f = open('123790.txt', 'r')
lines = f.read().splitlines()
f.close()

lines[2] = 'Katowice, Polarna'

f = open('123790.txt', 'w')
f.write('\n'.join(lines))
f.close()

f = open('123856.txt', 'r')
lines = f.read().splitlines()
f.close()

old_salary = int(lines[3])
new_salary = 1.10 * old_salary
lines[3] = str(new_salary)

f = open('123856.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

Similar task

- 697123: 12% salary raise
- 112001: 33% salary raise

```
f = open('123313.txt', 'r')
lines = f.read().splitlines()
f.close()

old_salary = int(lines[3])
new_salary = 1.11 * old_salary
lines[3] = str(new_salary)

f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()

# ---

f = open('123790.txt', 'r')
lines = f.read().splitlines()
f.close()

lines[2] = 'Katowice, Polarna'

f = open('123790.txt', 'w')
f.write('\n'.join(lines))
f.close()

# ---

f = open('123856.txt', 'r')
lines = f.read().splitlines()
f.close()

old_salary = int(lines[3])
new_salary = 1.10 * old_salary
lines[3] = str(new_salary)

f = open('123856.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

```
# READ FILE
f = open('123313.txt', 'r')
lines = f.read().splitlines()
f.close()

# CHANGE SALARY
old_salary = int(lines[3])
new_salary = 1.11 * old_salary
lines[3] = str(new_salary)

# WRITE FILE
f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()

# ---

# READ FILE
f = open('123790.txt', 'r')
lines = f.read().splitlines()
f.close()

# CHANGE ADDRESS
lines[2] = 'Katowice, Polarna'

# WRITE FILE
f = open('123790.txt', 'w')
f.write('\n'.join(lines))
f.close()

# ---

# READ FILE
f = open('123856.txt', 'r')
lines = f.read().splitlines()
f.close()

# CHANGE SALARY
old_salary = int(lines[3])
new_salary = 1.10 * old_salary
lines[3] = str(new_salary)

# WRITE FILE
f = open('123856.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

Function

$$f(x) = x + 1$$

$$f(x) = x + 1$$
$$g(x) = 2x - x + 1$$
$$h(x) = \frac{42 + x^2 \cdot x}{x^3 + 42} - x + 1$$

$$f(x) = x + 1$$

```
def f(x):
    return x + 1
```

```
def f(x):
    print("Hi, how are you?")
    return x + 1
```

```
def f(x, y):
    print("Hi, how are you?")
    return x - y + 1
```

```
def f():
    print("Hi, how are you?")
```

```
# READ FILE
f = open('123856.txt', 'r')
lines = f.read().splitlines()
f.close()
```

```
def read_file(person):
  f = open(person + '.txt', 'r')
  lines = f.read().splitlines()
  f.close()
```

```
# READ FILE
f = open('123856.txt', 'r')
lines = f.read().splitlines()
f.close()
```

```
def read_file(person):
  f = open(person + '.txt', 'r')
  lines = f.read().splitlines()
  f.close()
  return lines
```

```
# CHANGE SALARY
old_salary = int(lines[3])
new_salary = 1.10 * old_salary
lines[3] = str(new_salary)
```

```
def change_salary(lines, x):
  old_salary = int(lines[3])
  new_salary = x * old_salary
  lines[3] = str(new_salary)
```

```
# CHANGE ADDRESS
lines[2] = 'Katowice, Polarna'
```

```
def change_address(lines, new_address):
  lines[2] = new_address
```

```
# WRITE FILE
f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

```
def write_file(person, lines):
  f = open(person + '.txt', 'w')
  f.write('\n'.join(lines))
  f.close()
```

```
# READ FILE
f = open('123313.txt', 'r')
lines = f.read().splitlines()
f.close()

# CHANGE SALARY
old_salary = int(lines[3])
new_salary = 1.11 * old_salary
lines[3] = str(new_salary)

# WRITE FILE
f = open('123313.txt', 'w')
f.write('\n'.join(lines))
f.close()

# ---

# READ FILE
f = open('123790.txt', 'r')
lines = f.read().splitlines()
f.close()

# CHANGE ADDRESS
lines[2] = 'Katowice, Polarna'

# WRITE FILE
f = open('123790.txt', 'w')
f.write('\n'.join(lines))
f.close()

# ---

# READ FILE
f = open('123856.txt', 'r')
lines = f.read().splitlines()
f.close()

# CHANGE SALARY
old_salary = int(lines[3])
new_salary = 1.10 * old_salary
lines[3] = str(new_salary)

# WRITE FILE
f = open('123856.txt', 'w')
f.write('\n'.join(lines))
f.close()
```

```
# READ FILE
lines = read_file('123313')

# CHANGE SALARY
change_salary(lines, 1.11)

# WRITE FILE
write_file('123313', lines)

# ---

# READ FILE
lines = read_file('123790')

# CHANGE ADDRESS
change_address(lines, 'Katowice, Polarna')

# WRITE FILE
write_file('123790', lines)

# ---

# READ FILE
lines = read_file('123856')

# CHANGE SALARY
change_salary(lines, 1.10)

# WRITE FILE
write_file('123856', lines)
```

```
lines = read_file('123313')
change_salary(lines, 1.11)
write_file('123313', lines)

lines = read_file('123790')
change_address(lines, 'Katowice, Polarna')
write_file('123790', lines)

lines = read_file('123856')
change_salary(lines, 1.10)
write_file('123856', lines)
```

Pure vs “dirty” functions

```
def payEmployee(employeeId, hours):
  connectionId = 12332

  today = datetime.date.today()
  weekday = today.isoweekday() # 1 for Monday, 7 for Sunday

  if weekday == 7:
    r = paySundayTime(connectionId, employeeId, hours)
  else:
    r = payNormalTime(connectionId, employeeId, hours)
  return r
```

```
def payEmployee(employeeId, hours, today):
  connectionId = 12332

  weekday = today.isoweekday() # 1 for Monday, 7 for Sunday

  if weekday == 7:
    r = paySundayTime(connectionId, employeeId, hours)
  else:
    r = payNormalTime(connectionId, employeeId, hours)
  return r
```

```
# def payEmployee(employeeId, hours, today)

r1 = payEmployee(1434, 8, datetime.date(2022, 4, 1))
r2 = payEmployee(1434, 8, datetime.date(2022, 4, 2))
r3 = payEmployee(1434, 8, datetime.date(2022, 4, 3))
r4 = payEmployee(1434, 8, datetime.date(2022, 4, 4))
```

Why would anybody need a program that doesn't interact with him?

```python
import sys

print("I was called with those arguments:")
print(sys.argv)
sys.exit(24)
```

```
> python args.py 123 445 9080
I was called with those arguments:
['args.py', '123', '445', '9080']

> echo %errorlevel%
24
```

“Clean up” the code using functions

```
# This program accepts one, two or three command-line arguments, converts them
# into numbers (originally they are strings) and prints the result of some
# mathematical operations using these arguments. If the command-line arguments
# are invalid, an error exit code is returned by the whole program to the
# operating system.

import sys

# the first element in 'sys.argv' (at index 0) is always the script file name;
# therefore look for the numbers starting with the second element (index 1)
if len(sys.argv) == 2:
    arg1 = int(sys.argv[1])
    if arg1 < 0:
        print("Sorry, expected non-negative integers")
        sys.exit(1) # end with 1 to indicate error
    result = arg1 * 42 - 4
    print("Result of the function is: ", result)
elif len(sys.argv) == 3:
    arg1 = int(sys.argv[1])
    if arg1 < 0:
        print("Sorry, expected non-negative integers")
        sys.exit(1) # end with 1 to indicate error
    arg2 = int(sys.argv[2])
    if arg2 < 0:
        print("Sorry, expected non-negative integers")
        sys.exit(1) # end with 1 to indicate error
    result = arg1 * arg2 - 4
    print("Result of the function is: ", result)
elif len(sys.argv) == 4:
    arg1 = int(sys.argv[1])
    if arg1 < 0:
        print("Sorry, expected non-negative integers")
        sys.exit(1) # end with 1 to indicate error
    arg2 = int(sys.argv[2])
    if arg2 < 0:
        print("Sorry, expected non-negative integers")
        sys.exit(1) # end with 1 to indicate error # end with 1 to indicate error
    arg3 = int(sys.argv[3])
    if arg3 < 0:
        print("Sorry, expected non-negative integers")
        sys.exit(1) # end with 1 to indicate error
    result = arg1 * arg2 - arg3
    print("Result of the function is: ", result)
else:
    print("Invalid arguments - expected 1..3 non-negative integer numbers")
    sys.exit(1) # end with 1 to indicate error
```