Práctica 8 - Opcodes

Wing Manuel Ortiz Uribe

Cesar Padilla Lazos

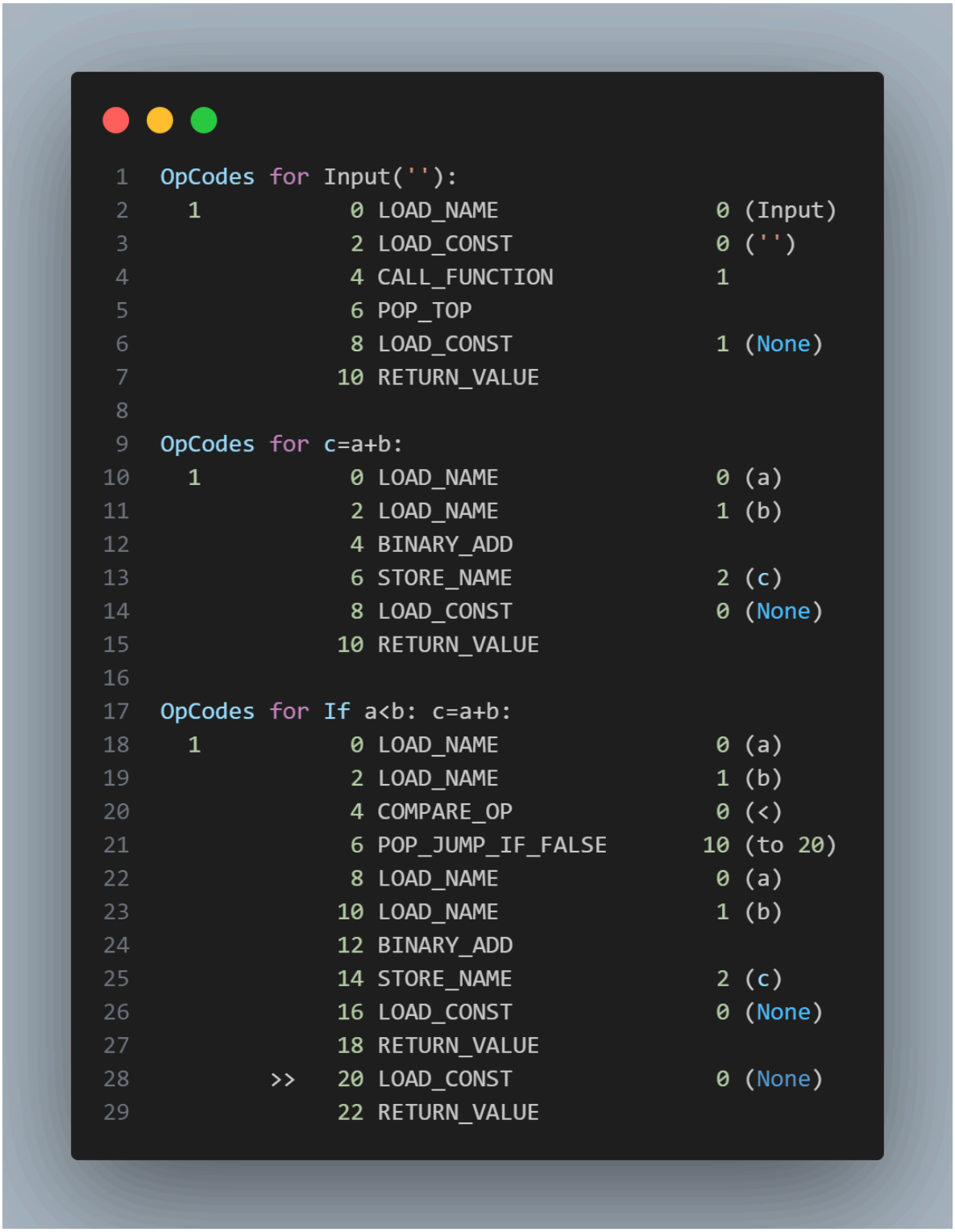
Objetivo: Identificar operaciones en bytecode para la implementación de instrucciones en python.

Instrucciones:

1. Elija una de las siguientes instrucciones y obtenga el bytecode que la reemplaza, importe la librería dis y use el método dis con cada uno de ellos.
    1) print(“”)
    2) print(“”+val)
    3) input()
    4) Input(“”)
    5) c=a+b
    6) If a<b:
    7) c=a+b

```python
import dis

bytecode_print_empty = compile("print('')", "", "exec")
print("Bytecode for print(''):")
print(bytecode_print_empty.co_consts[0])

# Mostrar opcodes
print("\nOpCodes:")
dis.dis(bytecode_print_empty)

bytecode_print_plus_val = compile("print(''+val)", "", "exec")
print("\nOpCodes for print(''+val):")
dis.dis(bytecode_print_plus_val)

bytecode_input = compile("input()", "", "exec")
print("\nOpCodes for input():")
dis.dis(bytecode_input)

bytecode_Input = compile("Input('')", "", "exec")
print("\nOpCodes for Input(''):")
dis.dis(bytecode_Input)

bytecode_Input = compile("Input('')", "", "exec")
print("\nOpCodes for Input(''):")
dis.dis(bytecode_Input)

bytecode_c_assignment = compile("c=a+b", "", "exec")
print("\nOpCodes for c=a+b:")
dis.dis(bytecode_c_assignment)

bytecode_if_statement = compile("if a<b: c=a+b", "", "exec")
print("\nOpCodes for If a<b: c=a+b:")
dis.dis(bytecode_if_statement)

```

```
Bytecode for print(''):


OpCodes:
  1           0 LOAD_NAME                0 (print)
              2 LOAD_CONST               0 ('')
              4 CALL_FUNCTION            1
              6 POP_TOP
              8 LOAD_CONST               1 (None)
             10 RETURN_VALUE

OpCodes for print(''+val):
  1           0 LOAD_NAME                0 (print)
              2 LOAD_CONST               0 ('')
              4 LOAD_NAME                1 (val)
              6 BINARY_ADD
              8 CALL_FUNCTION            1
             10 POP_TOP
             12 LOAD_CONST               1 (None)
             14 RETURN_VALUE

OpCodes for input():
  1           0 LOAD_NAME                0 (input)
              2 CALL_FUNCTION            0
              4 POP_TOP
              6 LOAD_CONST               0 (None)
              8 RETURN_VALUE

OpCodes for Input(''):
  1           0 LOAD_NAME                0 (Input)
              2 LOAD_CONST               0 ('')
              4 CALL_FUNCTION            1
              6 POP_TOP
              8 LOAD_CONST               1 (None)
             10 RETURN_VALUE

```



```
OpCodes for Input(''):
  1           0 LOAD_NAME                0 (Input)
              2 LOAD_CONST               0 ('')
              4 CALL_FUNCTION            1
              6 POP_TOP
              8 LOAD_CONST               1 (None)
             10 RETURN_VALUE

OpCodes for c=a+b:
  1           0 LOAD_NAME                0 (a)
              2 LOAD_NAME                1 (b)
              4 BINARY_ADD
              6 STORE_NAME               2 (c)
              8 LOAD_CONST               0 (None)
             10 RETURN_VALUE

OpCodes for If a<b: c=a+b:
  1           0 LOAD_NAME                0 (a)
              2 LOAD_NAME                1 (b)
              4 COMPARE_OP               0 (<)
              6 POP_JUMP_IF_FALSE       10 (to 20)
              8 LOAD_NAME                0 (a)
             10 LOAD_NAME                1 (b)
             12 BINARY_ADD
             14 STORE_NAME               2 (c)
             16 LOAD_CONST               0 (None)
             18 RETURN_VALUE
        >>   20 LOAD_CONST               0 (None)
             22 RETURN_VALUE
```