

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`





```
1  import dis
2
3  bytecode_print_empty = compile("print('')", "", "exec")
4  print("Bytecode for print(''):")
5  print(bytecode_print_empty.co_consts[0])
6
7  # Mostrar opcodes
8  print("\nOpCodes:")
9  dis.dis(bytecode_print_empty)
10
11 bytecode_print_plus_val = compile("print(''+val)", "", "exec")
12 print("\nOpCodes for print(''+val):")
13 dis.dis(bytecode_print_plus_val)
14
15 bytecode_input = compile("input()", "", "exec")
16 print("\nOpCodes for input():")
17 dis.dis(bytecode_input)
18
19 bytecode_Input = compile("Input('')", "", "exec")
20 print("\nOpCodes for Input(''):")
21 dis.dis(bytecode_Input)
22
23 bytecode_Input = compile("Input('')", "", "exec")
24 print("\nOpCodes for Input(''):")
25 dis.dis(bytecode_Input)
26
27 bytecode_c_assignment = compile("c=a+b", "", "exec")
28 print("\nOpCodes for c=a+b:")
29 dis.dis(bytecode_c_assignment)
30
31 bytecode_if_statement = compile("if a<b: c=a+b", "", "exec")
32 print("\nOpCodes for If a<b: c=a+b:")
33 dis.dis(bytecode_if_statement)
34
```



```
1 Bytecode for print(''):
2
3
4 OpCodes:
5     1          0 LOAD_NAME          0 (print)
6             2 LOAD_CONST          0 (')
7             4 CALL_FUNCTION        1
8             6 POP_TOP
9             8 LOAD_CONST          1 (None)
10            10 RETURN_VALUE
11
12 OpCodes for print(''+val):
13     1          0 LOAD_NAME          0 (print)
14             2 LOAD_CONST          0 (')
15             4 LOAD_NAME          1 (val)
16             6 BINARY_ADD
17             8 CALL_FUNCTION        1
18            10 POP_TOP
19            12 LOAD_CONST          1 (None)
20            14 RETURN_VALUE
21
22 OpCodes for input():
23     1          0 LOAD_NAME          0 (input)
24             2 CALL_FUNCTION        0
25             4 POP_TOP
26             6 LOAD_CONST          0 (None)
27             8 RETURN_VALUE
28
29 OpCodes for Input(''):
30     1          0 LOAD_NAME          0 (Input)
31             2 LOAD_CONST          0 (')
32             4 CALL_FUNCTION        1
33             6 POP_TOP
34             8 LOAD_CONST          1 (None)
35            10 RETURN_VALUE
36
```



```
1  OpCodes for Input(''):
2      1          0 LOAD_NAME          0 (Input)
3          2 LOAD_CONST          0 ('')
4          4 CALL_FUNCTION          1
5          6 POP_TOP
6          8 LOAD_CONST          1 (None)
7         10 RETURN_VALUE
8
9  OpCodes for c=a+b:
10     1          0 LOAD_NAME          0 (a)
11          2 LOAD_NAME          1 (b)
12          4 BINARY_ADD
13          6 STORE_NAME          2 (c)
14          8 LOAD_CONST          0 (None)
15         10 RETURN_VALUE
16
17  OpCodes for If a<b: c=a+b:
18     1          0 LOAD_NAME          0 (a)
19          2 LOAD_NAME          1 (b)
20          4 COMPARE_OP          0 (<)
21          6 POP_JUMP_IF_FALSE    10 (to 20)
22          8 LOAD_NAME          0 (a)
23         10 LOAD_NAME          1 (b)
24         12 BINARY_ADD
25         14 STORE_NAME          2 (c)
26         16 LOAD_CONST          0 (None)
27         18 RETURN_VALUE
28     >>      20 LOAD_CONST          0 (None)
29             22 RETURN_VALUE
```

