

```
def_op('STOP_CODE', 0)
def_op('POP_TOP', 1)
def_op('ROT_TWO', 2)
def_op('ROT_THREE', 3)
def_op('DUP_TOP', 4)
def_op('ROT_FOUR', 5)
def_op('NOP', 9)
def_op('UNARY_POSITIVE', 10)
def_op('UNARY_NEGATIVE', 11)
def_op('UNARY_NOT', 12)
def_op('UNARY_CONVERT', 13)
def_op('UNARY_INVERT', 15)
def_op('BINARY_POWER', 19)
def_op('BINARY_MULTIPLY', 20)
def_op('BINARY_DIVIDE', 21)
def_op('BINARY_MODULO', 22)
def_op('BINARY_ADD', 23)
def_op('BINARY_SUBTRACT', 24)
def_op('BINARY_SUBSCR', 25)
def_op('BINARY_FLOOR_DIVIDE', 26)
def_op('BINARY_TRUE_DIVIDE', 27)
def_op('INPLACE_FLOOR_DIVIDE', 28)
def_op('INPLACE_TRUE_DIVIDE', 29)
def_op('SLICE+0', 30)
def_op('SLICE+1', 31)
def_op('SLICE+2', 32)
def_op('SLICE+3', 33)
def_op('STORE_SLICE+0', 40)
def_op('STORE_SLICE+1', 41)
def_op('STORE_SLICE+2', 42)
def_op('STORE_SLICE+3', 43)
def_op('DELETE_SLICE+0', 50)
def_op('DELETE_SLICE+1', 51)
def_op('DELETE_SLICE+2', 52)
def_op('DELETE_SLICE+3', 53)
def_op('STORE_MAP', 54)
def_op('INPLACE_ADD', 55)
def_op('INPLACE_SUBTRACT', 56)
def_op('INPLACE_MULTIPLY', 57)
def_op('INPLACE_DIVIDE', 58)
def_op('INPLACE_MODULO', 59)
def_op('STORE_SUBSCR', 60)
def_op('DELETE_SUBSCR', 61)
def_op('BINARY_LSHIFT', 62)
def_op('BINARY_RSHIFT', 63)
def_op('BINARY_AND', 64)
def_op('BINARY_XOR', 65)
def_op('BINARY_OR', 66)
def_op('INPLACE_POWER', 67)
def_op('GET_ITER', 68)
def_op('PRINT_EXPR', 70)
def_op('PRINT_ITEM', 71)
def_op('PRINT_NEWLINE', 72)
def_op('PRINT_ITEM_TO', 73)
def_op('PRINT_NEWLINE_TO', 74)
def_op('INPLACE_LSHIFT', 75)
def_op('INPLACE_RSHIFT', 76)
def_op('INPLACE_AND', 77)
def_op('INPLACE_XOR', 78)
def_op('INPLACE_OR', 79)
def_op('BREAK_LOOP', 80)
def_op('WITH_CLEANUP', 81)
def_op('LOAD_LOCALS', 82)
def_op('RETURN_VALUE', 83)
```

```

def_op('IMPORT_STAR', 84)
def_op('EXEC_STMT', 85)
def_op('YIELD_VALUE', 86)
def_op('POP_BLOCK', 87)
def_op('END_FINALLY', 88)
def_op('BUILD_CLASS', 89)
HAVE_ARGUMENT = 90
name_op('STORE_NAME', 90)
name_op('DELETE_NAME', 91)
def_op('UNPACK_SEQUENCE', 92)
jrel_op('FOR_ITER', 93)
def_op('LIST_APPEND', 94)
name_op('STORE_ATTR', 95)
name_op('DELETE_ATTR', 96)
name_op('STORE_GLOBAL', 97)
name_op('DELETE_GLOBAL', 98)
def_op('DUP_TOPX', 99)
def_op('LOAD_CONST', 100)
name_op('LOAD_NAME', 101)
def_op('BUILD_TUPLE', 102)
def_op('BUILD_LIST', 103)
def_op('BUILD_SET', 104)
def_op('BUILD_MAP', 105)
name_op('LOAD_ATTR', 106)
def_op('COMPARE_OP', 107)
name_op('IMPORT_NAME', 108)
name_op('IMPORT_FROM', 109)
jrel_op('JUMP_FORWARD', 110)
jabs_op('JUMP_IF_FALSE_OR_POP', 111)
jabs_op('JUMP_IF_TRUE_OR_POP', 112)
jabs_op('JUMP_ABSOLUTE', 113)
jabs_op('POP_JUMP_IF_FALSE', 114)
jabs_op('POP_JUMP_IF_TRUE', 115)
name_op('LOAD_GLOBAL', 116)
jabs_op('CONTINUE_LOOP', 119)
jrel_op('SETUP_LOOP', 120)
jrel_op('SETUP_EXCEPT', 121)
jrel_op('SETUP_FINALLY', 122)
def_op('LOAD_FAST', 124)
def_op('STORE_FAST', 125)
def_op('DELETE_FAST', 126)
def_op('RAISE_VARARGS', 130)
def_op('CALL_FUNCTION', 131)
def_op('MAKE_FUNCTION', 132)
def_op('BUILD_SLICE', 133)
def_op('MAKE_CLOSURE', 134)
def_op('LOAD_CLOSURE', 135)
def_op('LOAD_DEREF', 136)
def_op('STORE_DEREF', 137)
def_op('CALL_FUNCTION_VAR', 140)
def_op('CALL_FUNCTION_KW', 141)
def_op('CALL_FUNCTION_VAR_KW', 142)
jrel_op('SETUP_WITH', 143)
def_op('EXTENDED_ARG', 145)
EXTENDED_ARG = 145
def_op('SET_ADD', 146)
def_op('MAP_ADD', 147)

```

*# Opcodes from here have an argument:*

*# Index in name list*

*# ""*

*# Number of tuple items*

*# Index in name list*

*# ""*

*# ""*

*# ""*

*# number of items to duplicate*

*# Index in const list*

*# Index in name list*

*# Number of tuple items*

*# Number of list items*

*# Number of set items*

*# Number of dict entries (upto 255)*

*# Index in name list*

*# Comparison operator*

*# Index in name list*

*# Index in name list*

*# Number of bytes to skip*

*# Target byte offset from beginning of code*

*# ""*

*# ""*

*# ""*

*# Index in name list*

*# Target address*

*# Distance to target address*

*# ""*

*# ""*

*# Local variable number*

*# Local variable number*

*# Local variable number*

*# Number of raise arguments (1, 2, or 3)*

*# #args + (#kwargs << 8)*

*# Number of args with default values*

*# Number of items*

*# #args + (#kwargs << 8)*

*# #args + (#kwargs << 8)*

*# #args + (#kwargs << 8)*