Transaction Script Language Operators, Constants, and Symbols

2022-03-04

# Transaction Script Language Operators, Constants, and Symbols

Tables and descriptions sourced from <https://en.bitcoin.it/wiki/Script>.

[table\_title](#tx_script_ops_table_pushdata) shows operators for pushing values onto the stack.

Push value onto stack

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_0 or OP\_FALSE | 0x00 | An empty array is pushed onto the stack |
| 1 to 75 | 0x01 to 0x4b | Push the next N bytes onto the stack, where N is 1 to 75 bytes |
| OP\_PUSHDATA1 | 0x4c | The next script byte contains N, push the following N bytes onto the stack |
| OP\_PUSHDATA2 | 0x4d | The next two script bytes contain N, push the following N bytes onto the stack |
| OP\_PUSHDATA4 | 0x4e | The next four script bytes contain N, push the following N bytes onto the stack |
| OP\_1NEGATE | 0x4f | Push the value "–1" onto the stack |
| OP\_RESERVED | 0x50 | Halt—Invalid transaction unless found in an unexecuted OP\_IF clause |
| OP\_1 or OP\_TRUE | 0x51 | Push the value "1" onto the stack |
| OP\_2 to OP\_16 | 0x52 to 0x60 | For OP\_N, push the value "N" onto the stack, e.g., OP\_2 pushes "2" |

[table\_title](#tx_script_ops_table_control) shows conditional flow control operators.

Conditional flow control

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_NOP | 0x61 | Do nothing |
| OP\_VER | 0x62 | Halt—Invalid transaction unless found in an unexecuted OP\_IF clause |
| OP\_IF | 0x63 | Execute the statements following if top of stack is not 0 |
| OP\_NOTIF | 0x64 | Execute the statements following if top of stack is 0 |
| OP\_VERIF | 0x65 | Halt—Invalid transaction |
| OP\_VERNOTIF | 0x66 | Halt—Invalid transaction |
| OP\_ELSE | 0x67 | Execute only if the previous statements were not executed |
| OP\_ENDIF | 0x68 | End the OP\_IF, OP\_NOTIF, OP\_ELSE block |
| OP\_VERIFY | 0x69 | Check the top of the stack, halt and invalidate transaction if not TRUE |
| OP\_RETURN | 0x6a | Halt and invalidate transaction |

[table\_title](#tx_script_ops_table_timelock) shows operators used for timelocks.

Timelock operations

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_CHECKLOCKTIMEVERIFY (previously OP\_NOP2) | 0xb1 | Marks transaction as invalid if the top stack item is greater than the transaction’s nLockTime field, otherwise script evaluation continues as though an OP\_NOP was executed. Transaction is also invalid if 1. the stack is empty; or 2. the top stack item is negative; or 3. the top stack item is greater than or equal to 500000000 while the transaction’s nLockTime field is less than 500000000, or vice versa; or 4. the input’s nSequence field is equal to 0xffffffff. The precise semantics are described in BIP-65 |
| OP\_CHECKSEQUENCEVERIFY (previously OP\_NOP3) | 0xb2 | Marks transaction as invalid if the relative lock time of the input (enforced by BIP 0068 with nSequence) is not equal to or longer than the value of the top stack item. The precise semantics are described in BIP-112 |

[table\_title](#tx_script_ops_table_stack) shows operators used to manipulate the stack.

Stack operations

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_TOALTSTACK | 0x6b | Pop top item from stack and push to alternative stack |
| OP\_FROMALTSTACK | 0x6c | Pop top item from alternative stack and push to stack |
| OP\_2DROP | 0x6d | Pop top two stack items |
| OP\_2DUP | 0x6e | Duplicate top two stack items |
| OP\_3DUP | 0x6f | Duplicate top three stack items |
| OP\_2OVER | 0x70 | Copy the third and fourth items in the stack to the top |
| OP\_2ROT | 0x71 | Move the fifth and sixth items in the stack to the top |
| OP\_2SWAP | 0x72 | Swap the two top pairs of items in the stack |
| OP\_IFDUP | 0x73 | Duplicate the top item in the stack if it is not 0 |
| OP\_DEPTH | 0x74 | Count the items on the stack and push the resulting count |
| OP\_DROP | 0x75 | Pop the top item in the stack |
| OP\_DUP | 0x76 | Duplicate the top item in the stack |
| OP\_NIP | 0x77 | Pop the second item in the stack |
| OP\_OVER | 0x78 | Copy the second item in the stack and push it onto the top |
| OP\_PICK | 0x79 | Pop value N from top, then copy the Nth item to the top of the stack |
| OP\_ROLL | 0x7a | Pop value N from top, then move the Nth item to the top of the stack |
| OP\_ROT | 0x7b | Rotate the top three items in the stack |
| OP\_SWAP | 0x7c | Swap the top two items on the stack |
| OP\_TUCK | 0x7d | Copy the top item and insert it between the top and second item |

[table\_title](#tx_script_ops_table_splice) shows string operators.

String splice operations

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| *OP\_CAT* | 0x7e | Disabled (concatenates top two items) |
| *OP\_SUBSTR* | 0x7f | Disabled (returns substring) |
| *OP\_LEFT* | 0x80 | Disabled (returns left substring) |
| *OP\_RIGHT* | 0x81 | Disabled (returns right substring) |
| OP\_SIZE | 0x82 | Calculate string length of top item and push the result |

[table\_title](#tx_script_ops_table_binmath) shows binary arithmetic and boolean logic operators.

Binary arithmetic and conditionals

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| *OP\_INVERT* | 0x83 | Disabled (Flip the bits of the top item) |
| *OP\_AND* | 0x84 | Disabled (Boolean AND of two top items) |
| *OP\_OR* | 0x85 | Disabled (Boolean OR of two top items) |
| *OP\_XOR* | 0x86 | Disabled (Boolean XOR of two top items) |
| OP\_EQUAL | 0x87 | Push TRUE (1) if top two items are exactly equal, push FALSE (0) otherwise |
| OP\_EQUALVERIFY | 0x88 | Same as OP\_EQUAL, but run OP\_VERIFY after to halt if not TRUE |
| OP\_RESERVED1 | 0x89 | Halt—Invalid transaction unless found in an unexecuted OP\_IF clause |
| OP\_RESERVED2 | 0x8a | Halt—Invalid transaction unless found in an unexecuted OP\_IF clause |

[table\_title](#tx_script_ops_table_numbers) shows numeric (arithmetic) operators.

Numeric operators

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_1ADD | 0x8b | Add 1 to the top item |
| OP\_1SUB | 0x8c | Subtract 1 from the top item |
| *OP\_2MUL* | 0x8d | Disabled (multiply top item by 2) |
| *OP\_2DIV* | 0x8e | Disabled (divide top item by 2) |
| OP\_NEGATE | 0x8f | Flip the sign of top item |
| OP\_ABS | 0x90 | Change the sign of the top item to positive |
| OP\_NOT | 0x91 | If top item is 0 or 1 Boolean flip it, otherwise return 0 |
| OP\_0NOTEQUAL | 0x92 | If top item is 0 return 0, otherwise return 1 |
| OP\_ADD | 0x93 | Pop top two items, add them and push result |
| OP\_SUB | 0x94 | Pop top two items, subtract first from second, push result |
| *OP\_MUL* | 0x95 | Disabled (multiply top two items) |
| *OP\_DIV* | 0x96 | Disabled (divide second item by first item) |
| *OP\_MOD* | 0x97 | Disabled (remainder divide second item by first item) |
| *OP\_LSHIFT* | 0x98 | Disabled (shift second item left by first item number of bits) |
| *OP\_RSHIFT* | 0x99 | Disabled (shift second item right by first item number of bits) |
| OP\_BOOLAND | 0x9a | Boolean AND of top two items |
| OP\_BOOLOR | 0x9b | Boolean OR of top two items |
| OP\_NUMEQUAL | 0x9c | Return TRUE if top two items are equal numbers |
| OP\_NUMEQUALVERIFY | 0x9d | Same as NUMEQUAL, then OP\_VERIFY to halt if not TRUE |
| OP\_NUMNOTEQUAL | 0x9e | Return TRUE if top two items are not equal numbers |
| OP\_LESSTHAN | 0x9f | Return TRUE if second item is less than top item |
| OP\_GREATERTHAN | 0xa0 | Return TRUE if second item is greater than top item |
| OP\_LESSTHANOREQUAL | 0xa1 | Return TRUE if second item is less than or equal to top item |
| OP\_GREATERTHANOREQUAL | 0xa2 | Return TRUE if second item is great than or equal to top item |
| OP\_MIN | 0xa3 | Return the smaller of the two top items |
| OP\_MAX | 0xa4 | Return the larger of the two top items |
| OP\_WITHIN | 0xa5 | Return TRUE if the third item is between the second item (or equal) and first item |

[table\_title](#tx_script_ops_table_crypto) shows cryptographic function operators.

Cryptographic and hashing operations

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_RIPEMD160 | 0xa6 | Return RIPEMD160 hash of top item |
| OP\_SHA1 | 0xa7 | Return SHA1 hash of top item |
| OP\_SHA256 | 0xa8 | Return SHA256 hash of top item |
| OP\_HASH160 | 0xa9 | Return RIPEMD160(SHA256(x)) hash of top item |
| OP\_HASH256 | 0xaa | Return SHA256(SHA256(x)) hash of top item |
| OP\_CODESEPARATOR | 0xab | Mark the beginning of signature-checked data |
| OP\_CHECKSIG | 0xac | Pop a public key and signature and validate the signature for the transaction’s hashed data, return TRUE if matching |
| OP\_CHECKSIGVERIFY | 0xad | Same as CHECKSIG, then OP\_VERIFY to halt if not TRUE |
| OP\_CHECKMULTISIG | 0xae | Run CHECKSIG for each pair of signature and public key provided. All must match. Bug in implementation pops an extra value, prefix with OP\_0 as workaround |
| OP\_CHECKMULTISIGVERIFY | 0xaf | Same as CHECKMULTISIG, then OP\_VERIFY to halt if not TRUE |

[table\_title](#tx_script_ops_table_nop) shows nonoperator symbols.

Nonoperators

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_NOP1 to OP\_NOP10 | 0xb0 to 0xb9 | Does nothing, ignored |

[table\_title](#tx_script_ops_table_internal) shows operator codes reserved for use by the internal script parser.

Reserved OP codes for internal use by the parser

| Symbol | Value (hex) | Description |
| --- | --- | --- |
| OP\_SMALLDATA | 0xf9 | Represents small data field |
| OP\_SMALLINTEGER | 0xfa | Represents small integer data field |
| OP\_PUBKEYS | 0xfb | Represents public key fields |
| OP\_PUBKEYHASH | 0xfd | Represents a public key hash field |
| OP\_PUBKEY | 0xfe | Represents a public key field |
| OP\_INVALIDOPCODE | 0xff | Represents any OP code not currently assigned |