

# Script: Bitcoin's Programming Language

Ryan X. Charles  
Blockchain University  
San Francisco, Oct. 26 – Oct. 30, 2015



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Outline

Script Interpreter

P2SH

Standard Transactions

Opcodes

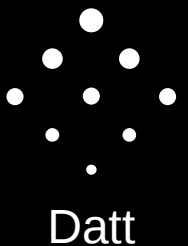
Validating Transactions and Blocks

Advanced Scripts



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Script Interpreter

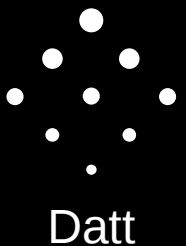
Input scriptSig 0	Output scriptPubkey 0
Input scriptSig 1	Output scriptPubkey 1
Input scriptSig 2	Output scriptPubkey 2

Transactions have inputs and outputs.  
Inputs have scripts. Outputs have scripts.

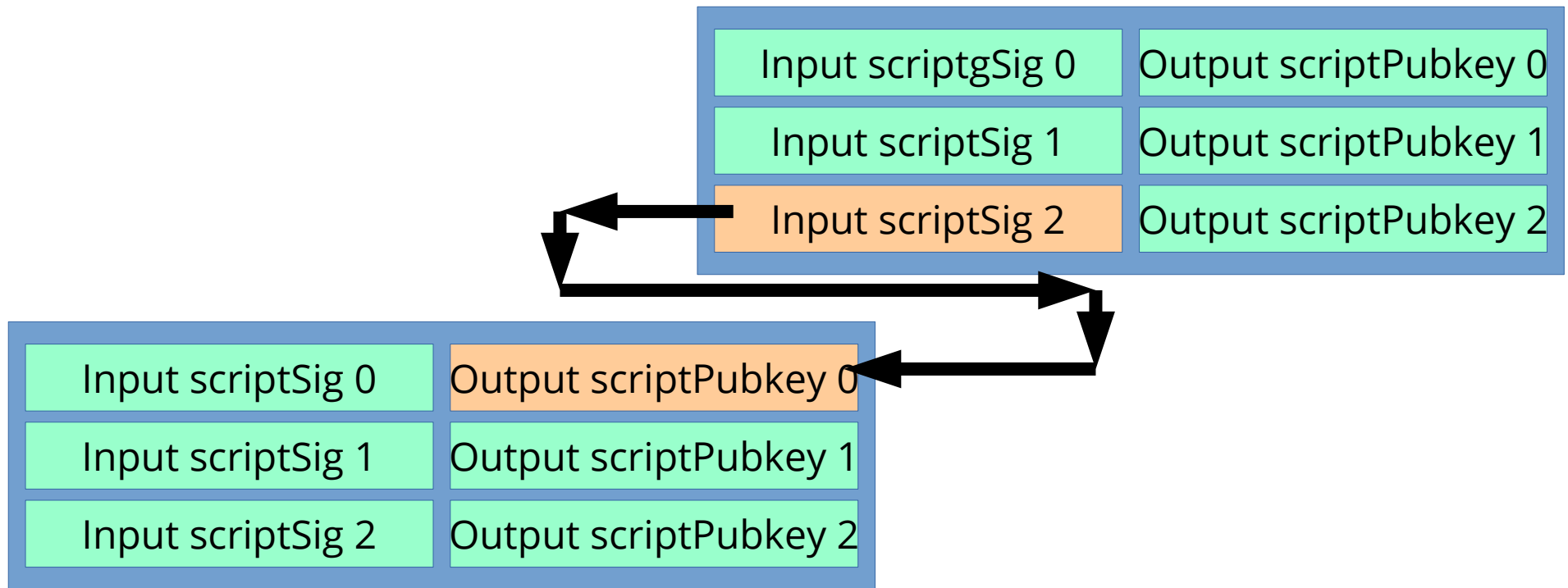


Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Script Interpreter



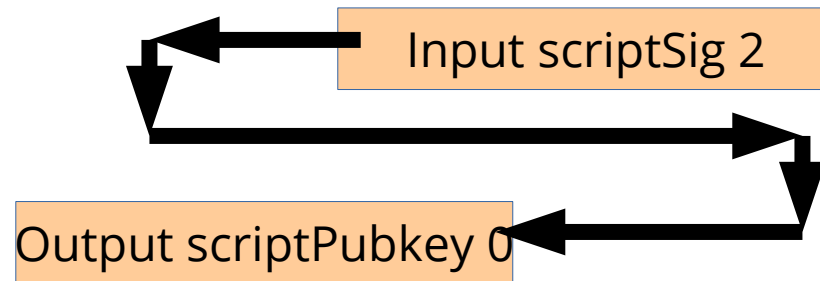
Each input links to the output of an earlier transaction.

Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Script Interpreter

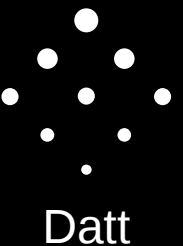


To validate an input, the scriptSig is executed and the scriptPubkey from the earlier transaction is executed.



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Script Interpreter

- The **stack** is the memory of bitcoin. Bitcoin does not have a heap. There is also an **alt stack** and things can be moved from stack to alt stack or from alt stack to stack.
- You can push and pop to the stack.
- Pubkeys and sigs are pushed to the stack. Other things like numbers can be pushed to the stack, if that's what the script does.

## stack

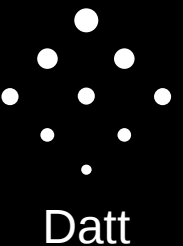
some other data

signature

pubkey

Ryan X. Charles  
Founder of Datt (datt.co)  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Script Interpreter

- To validate an input, the scriptSig is executed and the scriptPubkey from the earlier transaction is executed.
- After the scriptSig is executed, the stack is left the same, and the scriptPubkey runs starting with the same stack.
- Note that they are executed in “reverse” order – scriptSig first, from the later transaction, then scriptPubkey, from the earlier transaction

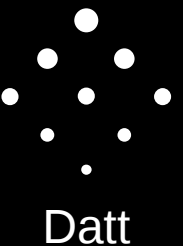
Input scriptSig 2

Output scriptPubkey 0



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Script Interpreter

pubkeyhash example, a.k.a. normal bitcoin address and transaction

## Stack

Empty.

<sig> <pubKey>

<sig> <pubKey> <pubKey>

<sig> <pubKey> <pubHashA>

<sig> <pubKey> <pubHashA> <pubKeyHash>

<sig> <pubKey>

true

## Script

<sig> <pubKey>

(scriptSig is now finished - run scriptPubKey next)

OP\_DUP OP\_HASH160 <pubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

OP\_HASH160 <pubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<pubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

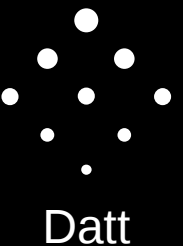
OP\_EQUALVERIFY OP\_CHECKSIG

OP\_CHECKSIG

Empty.

Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)





# Opcodes

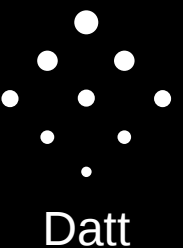
- One byte specifying an operation.

- OP\_FALSE: 0x00,
- OP\_0: 0x00,
- OP\_PUSHDATA1: 0x4c,
- OP\_PUSHDATA2: 0x4d,
- OP\_PUSHDATA4: 0x4e,
- OP\_1NEGATE: 0x4f,
- OP\_RESERVED: 0x50,
- OP\_TRUE: 0x51,
- OP\_1: 0x51,
- OP\_2: 0x52,
- OP\_3: 0x53,
- OP\_4: 0x54,
- OP\_5: 0x55,
- OP\_6: 0x56,
- OP\_7: 0x57,
- OP\_8: 0x58,
- OP\_9: 0x59,
- OP\_10: 0x5a,
- OP\_11: 0x5b,
- OP\_12: 0x5c,
- OP\_13: 0x5d,
- OP\_14: 0x5e,
- OP\_15: 0x5f,
- OP\_16: 0x60,
- OP\_NOP: 0x61,
- OP\_VER: 0x62,
- OP\_IF: 0x63,
- OP\_NOTIF: 0x64,
- OP\_VERIF: 0x65,
- OP\_VERNOTIF: 0x66,
- OP\_ELSE: 0x67,
- OP\_ENDIF: 0x68,
- OP\_VERIFY: 0x69,
- OP\_RETURN: 0x6a,
- ... ~175 total,  
including push data



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



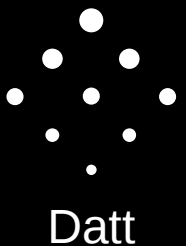
# Push Ops

- 0 – 0x80: That many bytes are pushed to the stack.
  - e.g., 0x05 0x0404040404 pushes “0404040404” to the stack
- PUSHDATA1: The following byte specifies amount of data to push
  - e.g.: OP\_PUSHDATA1 0x03 0x010203 pushes “010203” to the stack
- PUSHDATA2: The following two bytes (Uint16BE) specify the amount of data to push
- PUSHDATA4: The following four bytes (Uint32BE) specify the amount of data to push
- Anything 80 bytes or less can be pushed with a single byte push OP rather than PUSHDATAx – that includes signatures, usually ~70 bytes, and pubkeys, ~33 or ~65 bytes. p2sh redeemScripts, often containing multiple pubkeys, require PUSHDATA1 or PUSHDATA2



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Number Ops

- $0 = 0x00 = OP\_0 = OP\_FALSE$
- $OP\_1 = 81 = 0x51 = OP\_TRUE$
- $OP\_2 = 82 = 0x52$
- ...
- $OP\_16 = 96 = 0x60$
- Pushes that number to the stack. The number is a ScriptNum – zero bytes if 0, one byte if 1 - 16



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



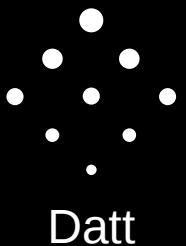
# Control Ops

- OP\_IF: 0x63 – checks that top item on stack is true
- OP\_NOTIF: 0x64
- OP\_ELSE: 0x67
- OP\_ENDIF: 0x68
- OP\_RETURN: 0x6a ← **commonly used**



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Stack Ops

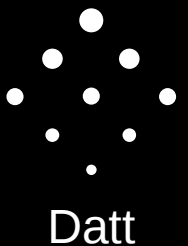
- Rearrange things on the stack or alt stack

- OP\_TOALTSTACK: 0x6b
- OP\_FROMALTSTACK: 0x6c
- OP\_2DROP: 0x6d
- OP\_2DUP: 0x6e
- OP\_3DUP: 0x6f
- OP\_2OVER: 0x70
- OP\_2ROT: 0x71
- OP\_2SWAP: 0x72
- OP\_IFDUP: 0x73
- OP\_DEPTH: 0x74
- OP\_DROP: 0x75
- OP\_DUP: 0x76 ← **commonly used**
- OP\_NIP: 0x77
- OP\_OVER: 0x78
- OP\_PICK: 0x79
- OP\_ROLL: 0x7a
- OP\_ROT: 0x7b
- OP\_SWAP: 0x7c
- OP\_TUCK: 0x7d



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



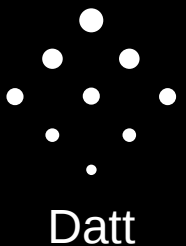
# Splice Ops

- Rearrange bytes of the top stack item
- OP\_CAT: 0x7e
- OP\_SUBSTR: 0x7f
- OP\_LEFT: 0x80
- OP\_RIGHT: 0x81
- OP\_SIZE: 0x82



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



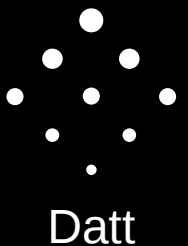
# Bitwise Ops

- Rearrange bytes of the top stack item
- OP\_INVERT: 0x83
- OP\_AND: 0x84
- OP\_OR: 0x85
- OP\_XOR: 0x86
- OP\_EQUAL: 0x87
- OP\_EQUALVERIFY: 0x88, ← **commonly used**
- OP\_RESERVED1: 0x89
- OP\_RESERVED2: 0x8a



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)


Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Numeric Ops

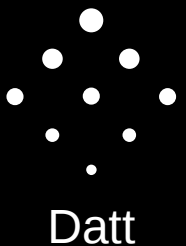
- Not commonly used – some are disabled

- OP\_1ADD: 0x8b
- OP\_1SUB: 0x8c
- OP\_2MUL: 0x8d
- OP\_2DIV: 0x8e
- OP\_NEGATE: 0x8f
- OP\_ABS: 0x90
- OP\_NOT: 0x91
- OP\_0NOTEQUAL: 0x92
- OP\_ADD: 0x93
- OP\_SUB: 0x94
- OP\_MUL: 0x95
- OP\_DIV: 0x96
- OP\_MOD: 0x97
- OP\_LSHIFT: 0x98
- OP\_RSHIFT: 0x99
- OP\_BOOLAND: 0x9a
- OP\_BOOLOR: 0x9b
- OP\_NUMEQUAL: 0x9c
- OP\_NUMEQUALVERIFY: 0x9d
- OP\_NUMNOTEQUAL: 0x9e
- OP\_LESSTHAN: 0x9f
- OP\_GREATERTHAN: 0xa0
- OP\_LESSTHANOREQUAL: 0xa1
- OP\_GREATERTHANOREQUAL: 0xa2
- OP\_MIN: 0xa3
- OP\_MAX: 0xa4



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)





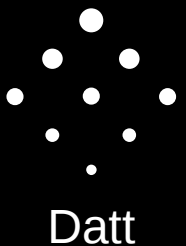
# Cryptography Ops

- OP\_RIPEMD160: 0xa6
- OP\_SHA1: 0xa7
- OP\_SHA256: 0xa8
- OP\_HASH160: 0xa9 ← **commonly used**
- OP\_HASH256: 0xaa
- OP\_CODESEPARATOR: 0xab
- OP\_CHECKSIG: 0xac ← **commonly used**
- OP\_CHECKSIGVERIFY: 0xad
- OP\_CHECKMULTISIG: 0xae ← **commonly used**
- OP\_CHECKMULTISIGVERIFY: 0xaf



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# Standard Script Types

- pubkeyhash ← most common
  - pubkey
  - multisig
  - p2sh
  - OP\_RETURN
- 
- Since p2sh redeemScript is itself a script, combinations like “p2sh multisig” are possible – p2sh multisig is most common use of p2sh



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



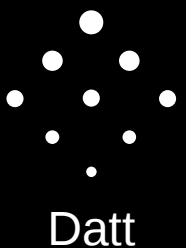
# pubkeyhash

- scriptSig:
  - <sig> <pubkey>
- scriptPubkey:
  - OP\_DUP OP\_HASH160 <address> OP\_EQUALVERIFY OP\_CHECKSIG



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



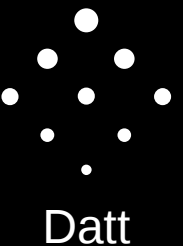
# pubkey (rare)

- scriptSig:
  - <sig>
- scriptPubkey:
  - <pubkey> OP\_CHECKSIG



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



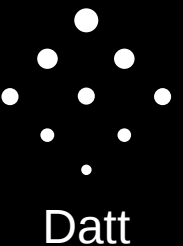
# multisig

- scriptSig:
- OP\_0 <sig1> <sig2> ... <sigm> ← starts with extra OP\_0 because of famous multisig bug – pops one too many items from stack
- scriptPubkey:
- OP\_m <pubkey1> <pubkey2> ... <pubkeyn> OP\_n  
OP\_CHECKMULTISIG



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



# p2sh multisig

- scriptSig:
  - OP\_0 <sig1> <sig2> ... <sigm> <redeemScript>
- scriptPubkey:
  - OP\_HASH160 <redeemScriptHash> OP\_EQUALVERIFY
- redeemScript:
  - OP\_m <pubkey1> <pubkey2> ... <pubkeyn> OP\_n  
OP\_CHECKMULTISIG



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



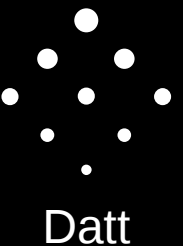
# OP\_RETURN

- scriptPubkey:
- OP\_RETURN <up to 40 (80?) bytes of data>
- How to put arbitrary data in an output if necessary.



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)



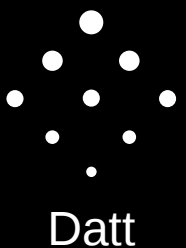
# Standard Transaction Rules

- If a transaction has all standard inputs/outputs, it is standard
- Standard transactions are relayed by default
- Non-standard transactions can still be valid, and can be in a block, if a miner receives it somehow and chooses to include it
- Complicated scripts are thus discouraged; preventing hypothetical DOS attacks on the network and blockchain bloat



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)





# Validating a Transaction

- Be sure that there is at least one input
- Be sure that transaction is not over MAX\_BLOCK\_SIZE
- Be sure that values are not negative or greater than MAX\_MONEY
- Be sure inputs are not duplicated
- Be sure that inputs are not null
- Run script interpreter on all inputs and be sure no inputs are invalid



Ryan X. Charles  
Founder of Datt ([datt.co](http://datt.co))  
[twitter.com/ryanxcharles](https://twitter.com/ryanxcharles)  
[github.com/ryanxcharles](https://github.com/ryanxcharles)

Code Samples and Slides at:  
[github.com/ryanxcharles/blockchain-university](https://github.com/ryanxcharles/blockchain-university)

