



**Bitcoin  
Script**

# Script



- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks



# Script



- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks

## Example Script

<signature><publicKey> OP\_CHECKSIG

# Script



- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks

## Example Script

`<signature><publicKey> OP_CHECKSIG`

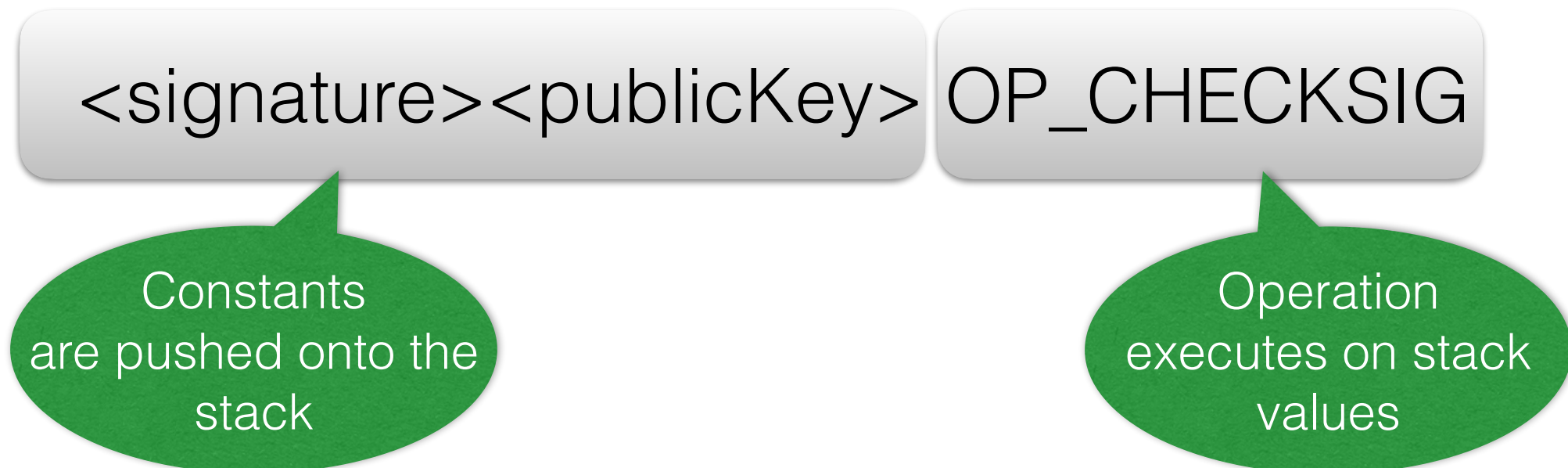
Constants  
are pushed onto the  
stack

# Script

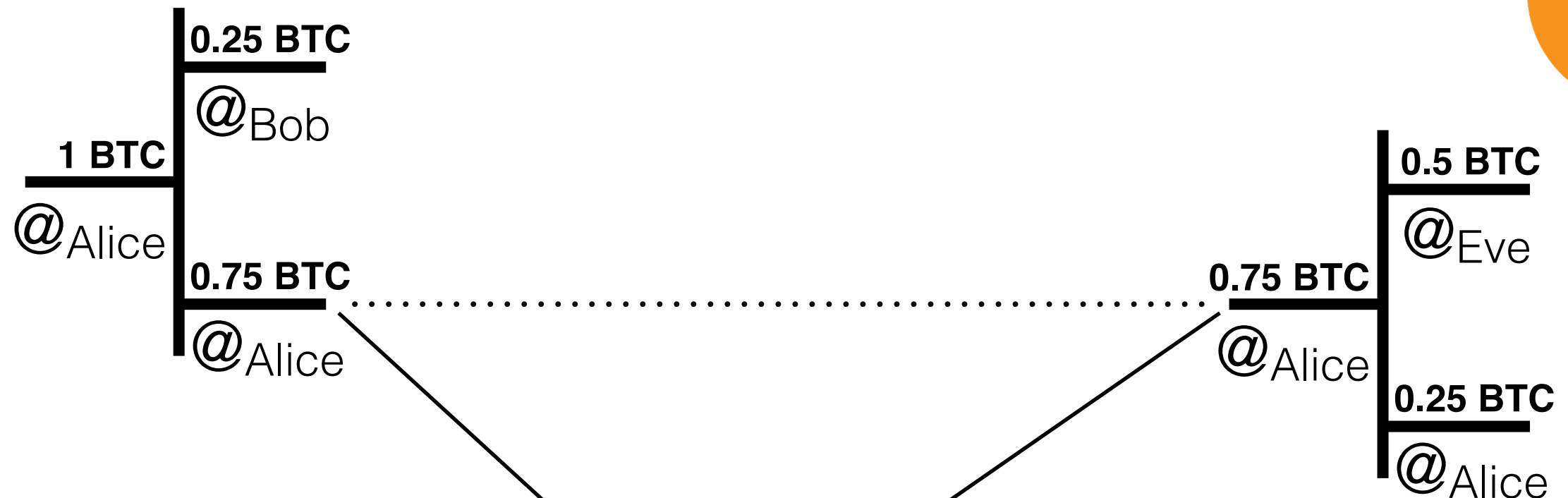


- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks

## Example Script



# Script Execution



`<Sig> <PubKey> OP_DUP OP_HASH160 <PubKeyHash> OP_EQUALVERIFY OP_CHECKSIG`

<PubKey>

<Sig>

OP\_DUP

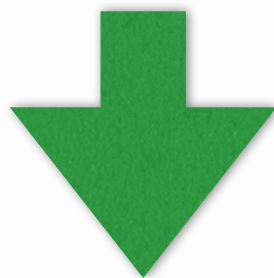
Execution Stack

Execution Code

Constants are pushed  
onto the stack

<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<PubKey>	
<Sig>	OP_DUP
Execution Stack	Execution Code



<PubKey>	
<PubKey>	
<Sig>	OP_HASH160
Execution Stack	Execution Code

Constants are pushed  
onto the stack

<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG



<PubKey> <Sig>	OP_DUP
Execution Stack	Execution Code



<PubKey> <PubKey> <Sig>	OP_HASH160
Execution Stack	Execution Code

<PubKeyHash> <PubKeyHash> <PubKey> <Sig>	OP_EQUALVERIFY
Execution Stack	Execution Code

Constants are pushed  
onto the stack

<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<PubKey> <Sig>	OP_DUP
Execution Stack	Execution Code



<PubKey> <PubKey> <Sig>	OP_HASH160
Execution Stack	Execution Code

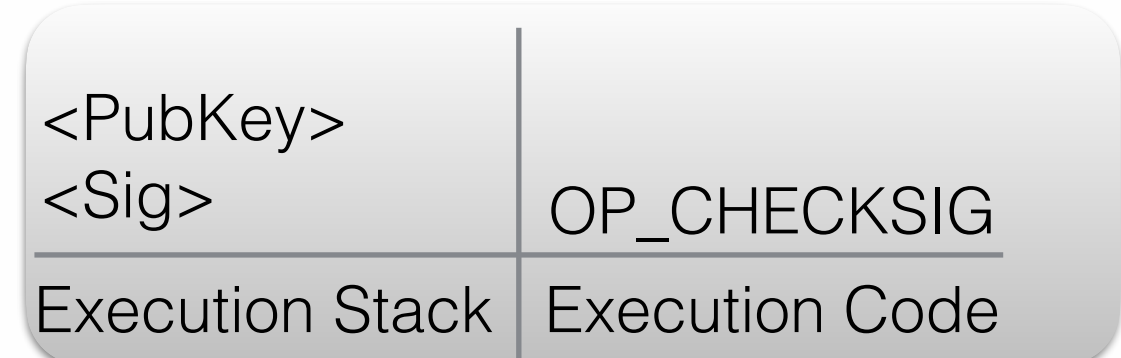
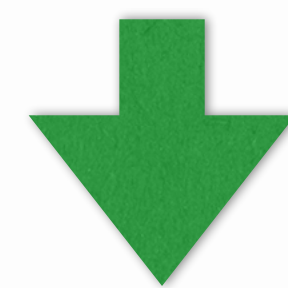
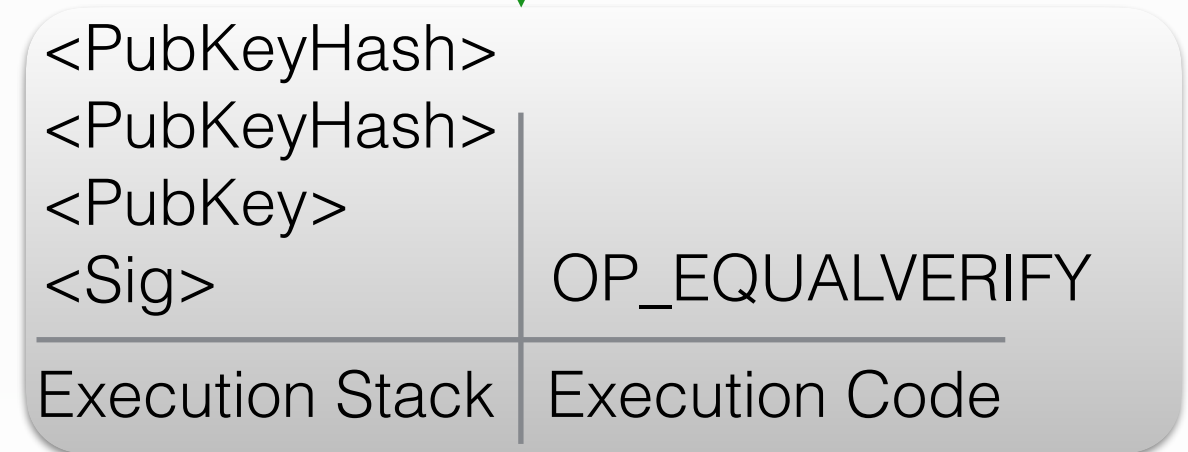
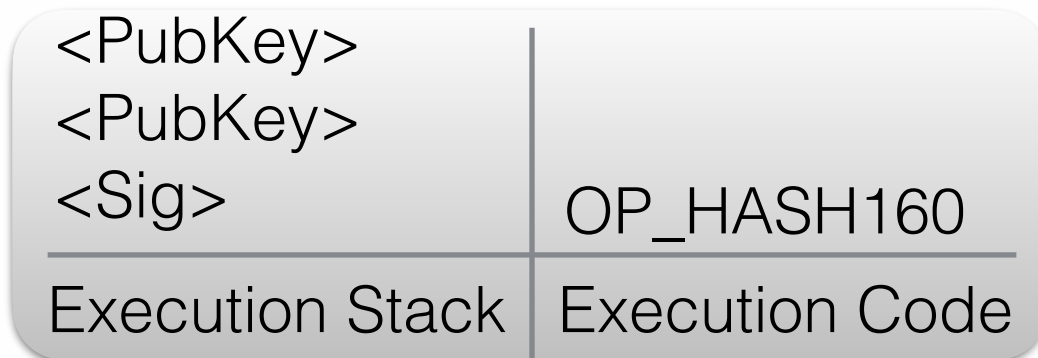
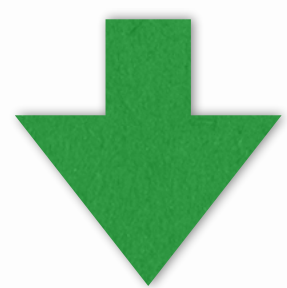
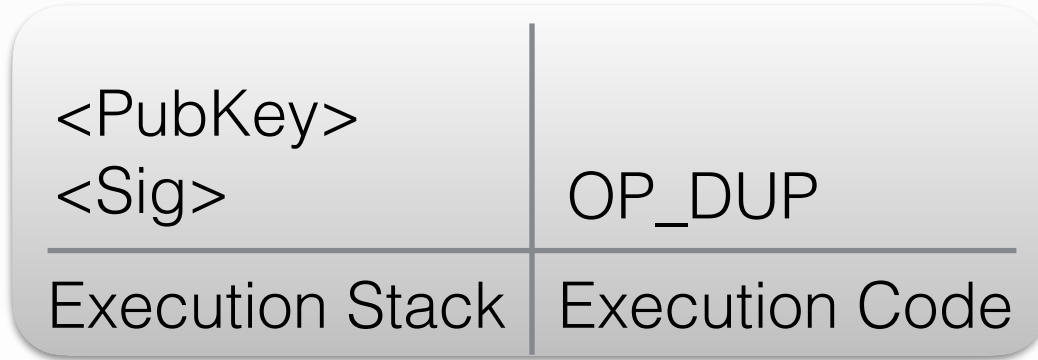
<PubKeyHash> <PubKeyHash> <PubKey> <Sig>	OP_EQUALVERIFY
Execution Stack	Execution Code



<PubKey> <Sig>	OP_CHECKSIG
Execution Stack	Execution Code

Constants are pushed  
onto the stack

<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG



**True**

Constants are pushed onto the stack

<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

# Transaction Types



- P2PKH - Pay to Public Key Hash
  - Redeemer needs a public key and signature
- P2SH - Pay to Script Hash
  - Redeemer needs a script that matches a pre-defined hash
- Multisignature (m-n)
  - Requires multiples signatures to be redeemable
  - **m** out of **n** signatures required