MIT Digital Currency Initiative and the University of Brasilia presents

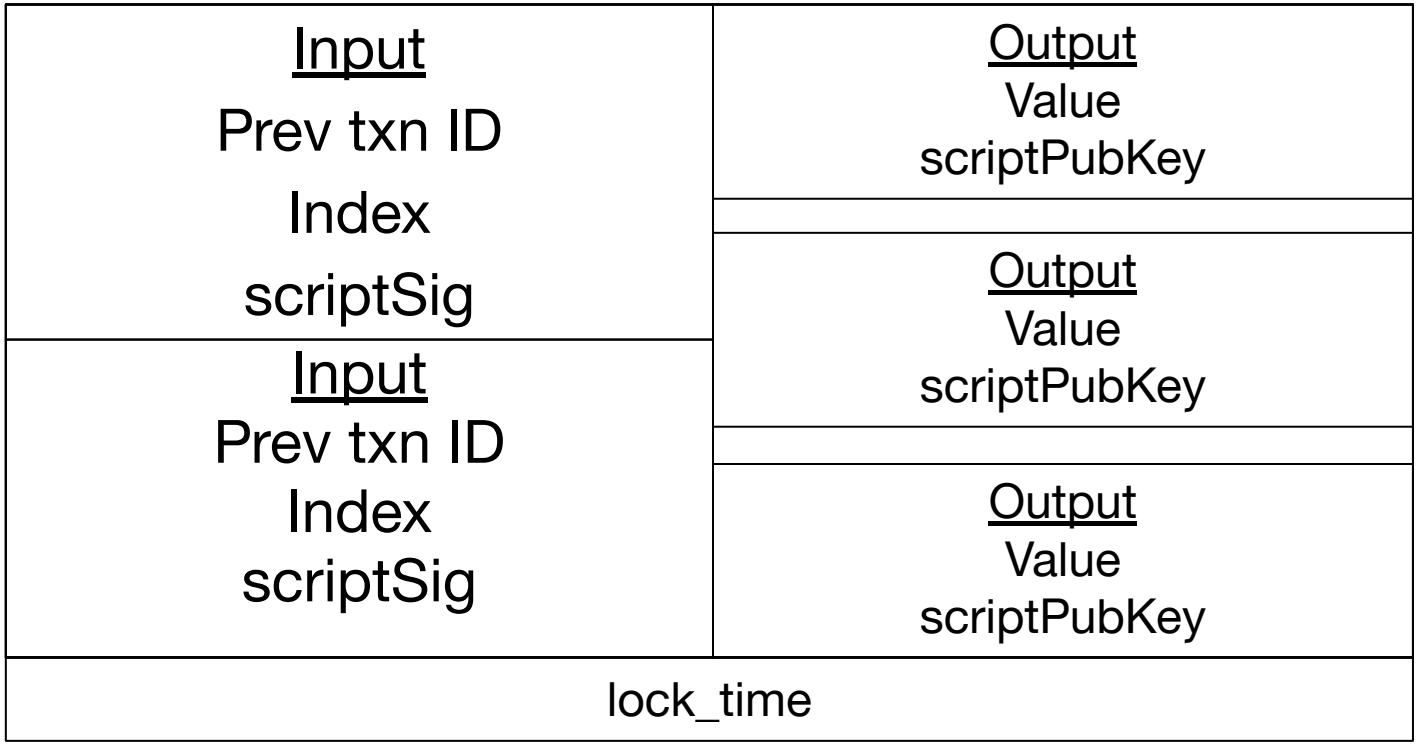
Cryptocurrency Design and Engineering

Lecture 10: Money Programmability
Taught by: Neha Narula
October 7, 2025
MAS.S62

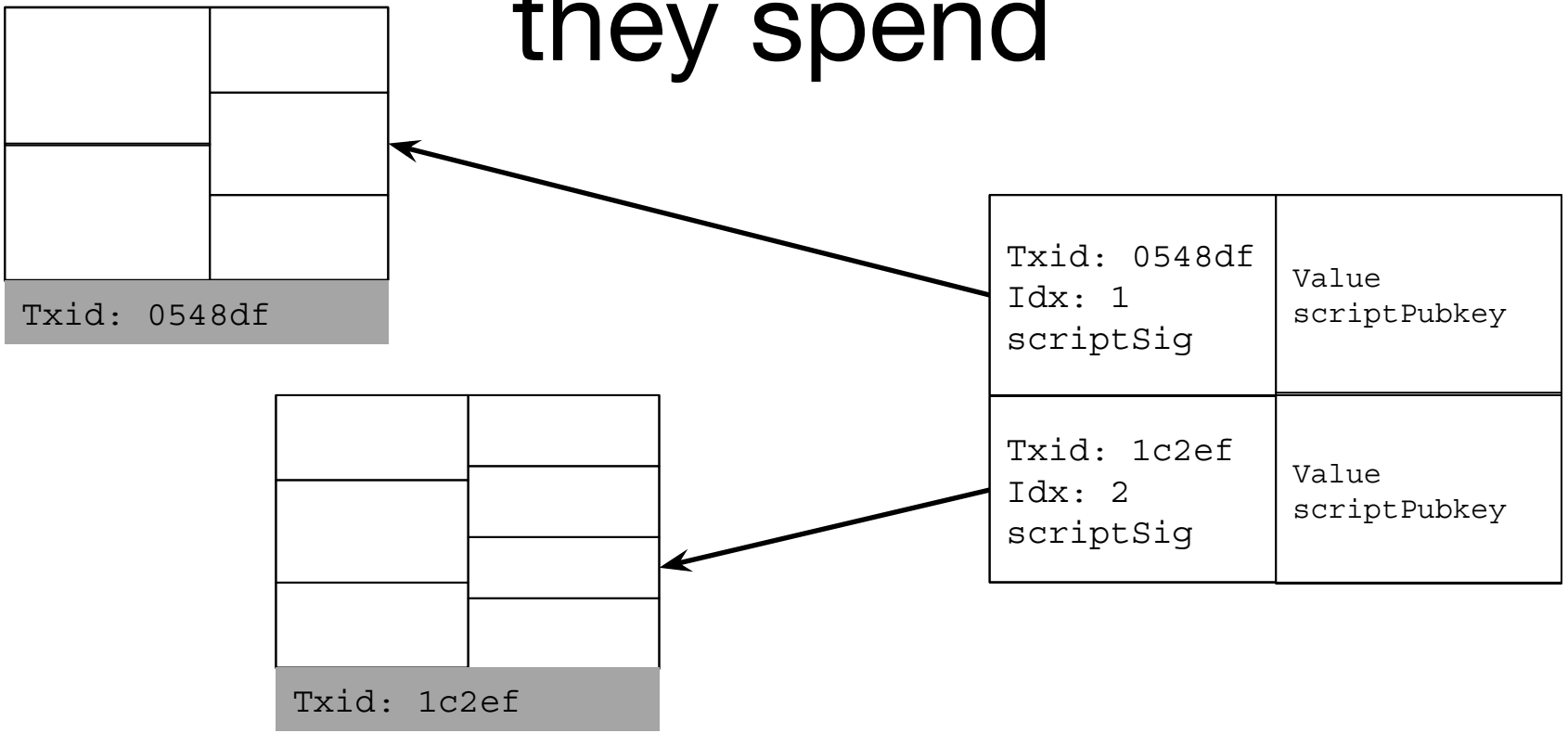
Why programmability?

- Conditional payments without intermediaries
- Better security (multisig, vaults)
- More complex applications with money

Transaction format



Inputs reference the outputs they spend



ScriptSigs and scriptPubkeys

- ScriptPubkeys are predicates
- ScriptSigs help satisfy the predicates
- When can you spend a coin? You know how to produce a satisfying scriptSig

Transaction validity rules

- Sum(inputs) >= Sum(outputs)
- For every input:
  - Eval(scriptSig+scriptPubKey) == true
  - Output has not already been spent
- lock_time

Bitcoin Script

- Stack-based
- Limited expressivity
- Limited set of opcodes
- Scoped execution
- Predictable execution (somewhat)

Pay to Pubkey Hash (P2PKH)

- Idea: Send money to a pubkey
- Pubkeys are big, a hash of a pubkey is only 32 bytes (+1 byte for prefix)
- scriptPubkey: instructions on how to verify a signature of a pubkey that is hashed
- scriptSig: signature, pubkey

Pay to Pubkey Hash (P2PKH)

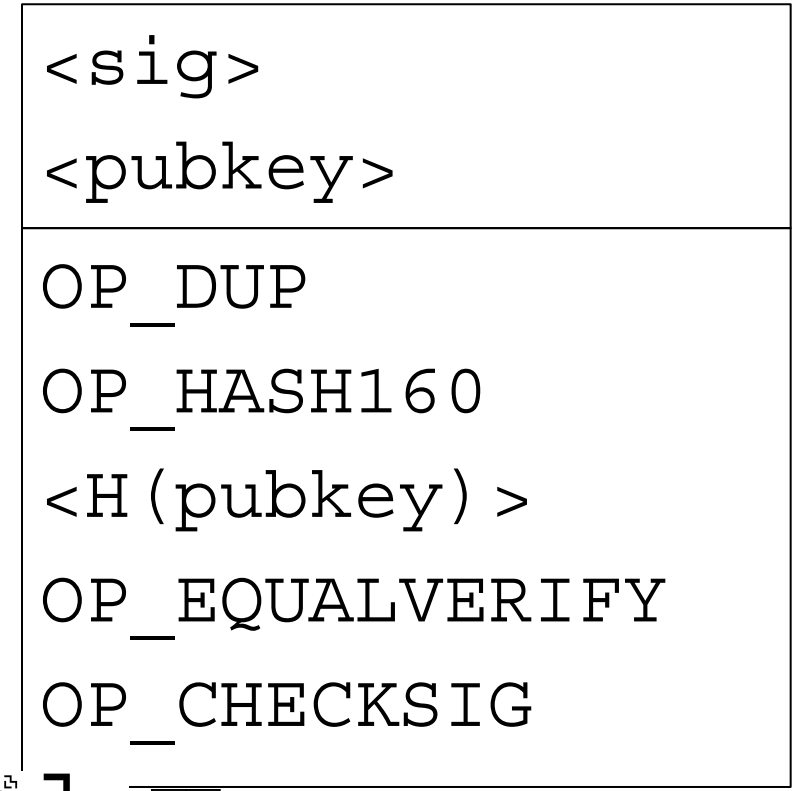
ScriptPubkey:

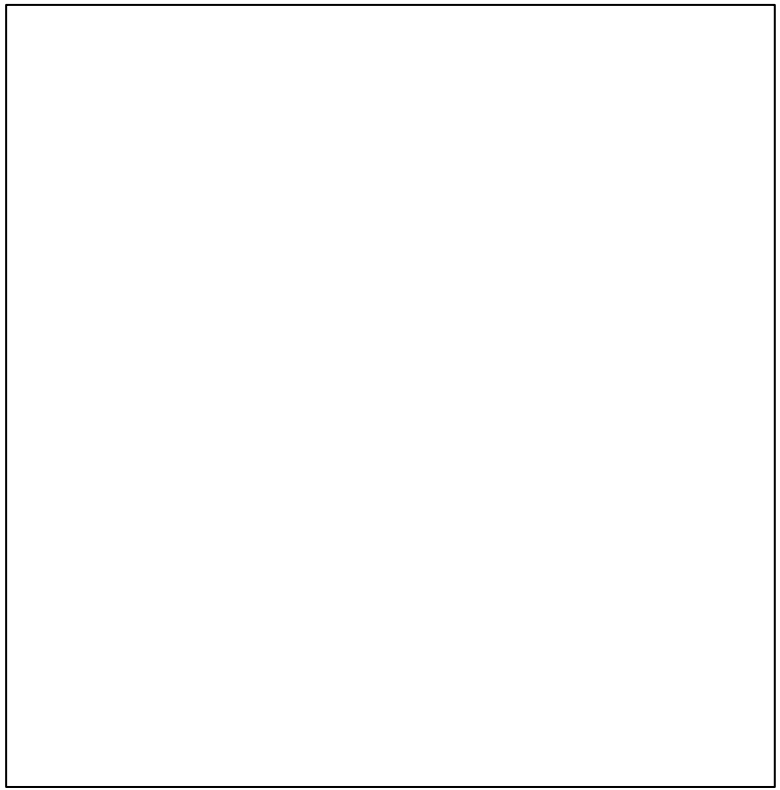
```
OP_DUP
OP_HASH160
<H(pubkey)>
OP_EQUALVERIFY
OP_CHECKSIG
```

ScriptSig:

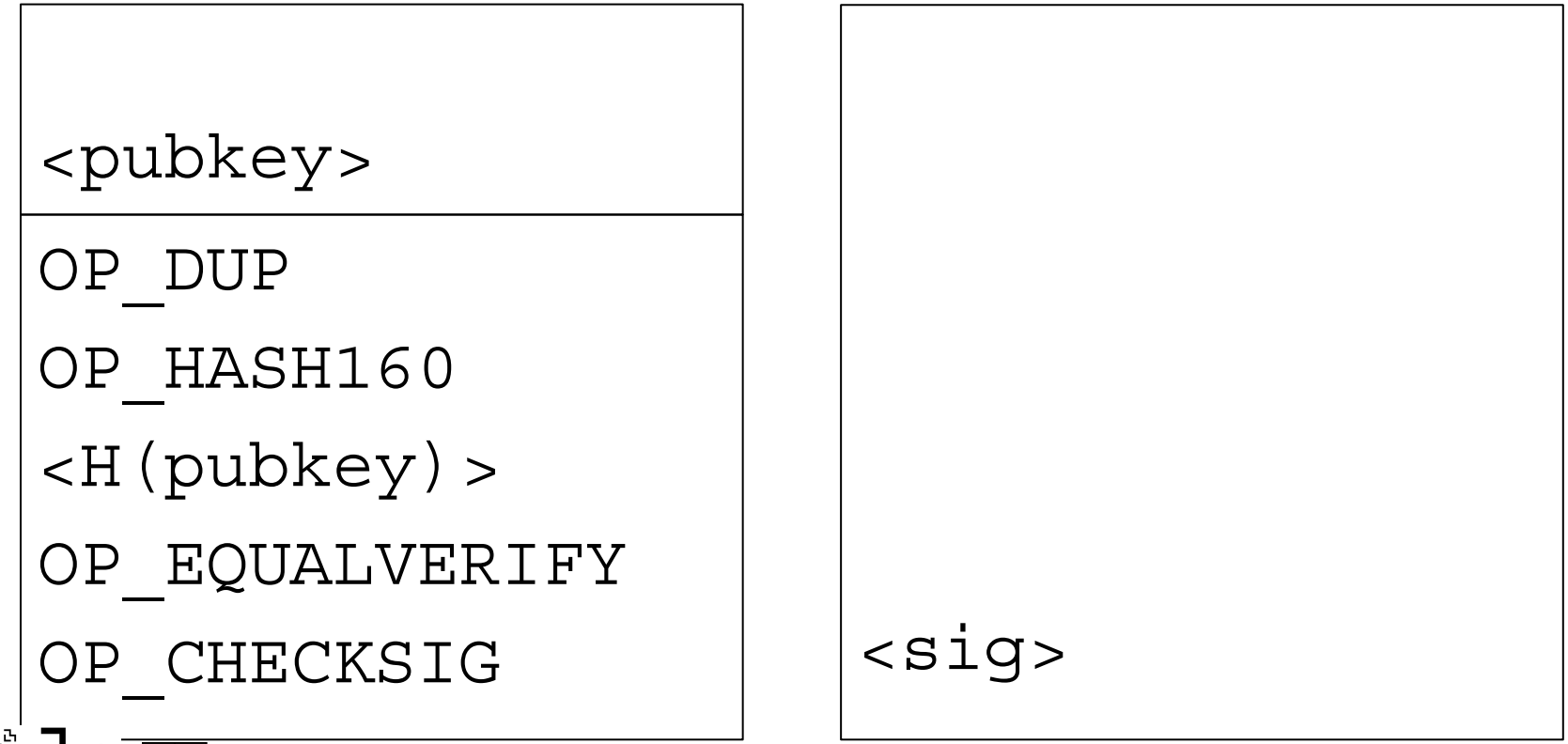
```
<sig>
<pubkey>
```

Pay to Pubkey Hash (P2PKH)

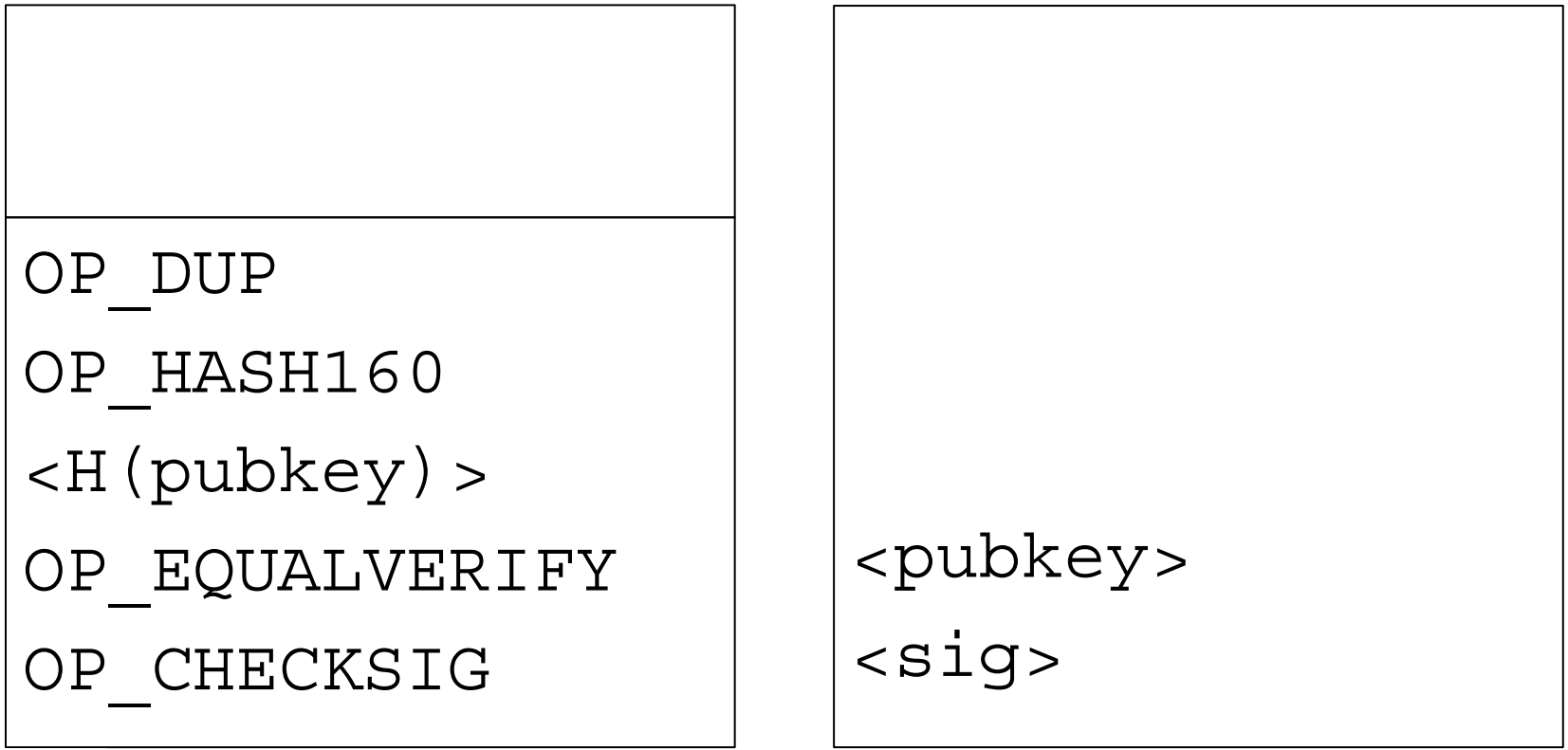




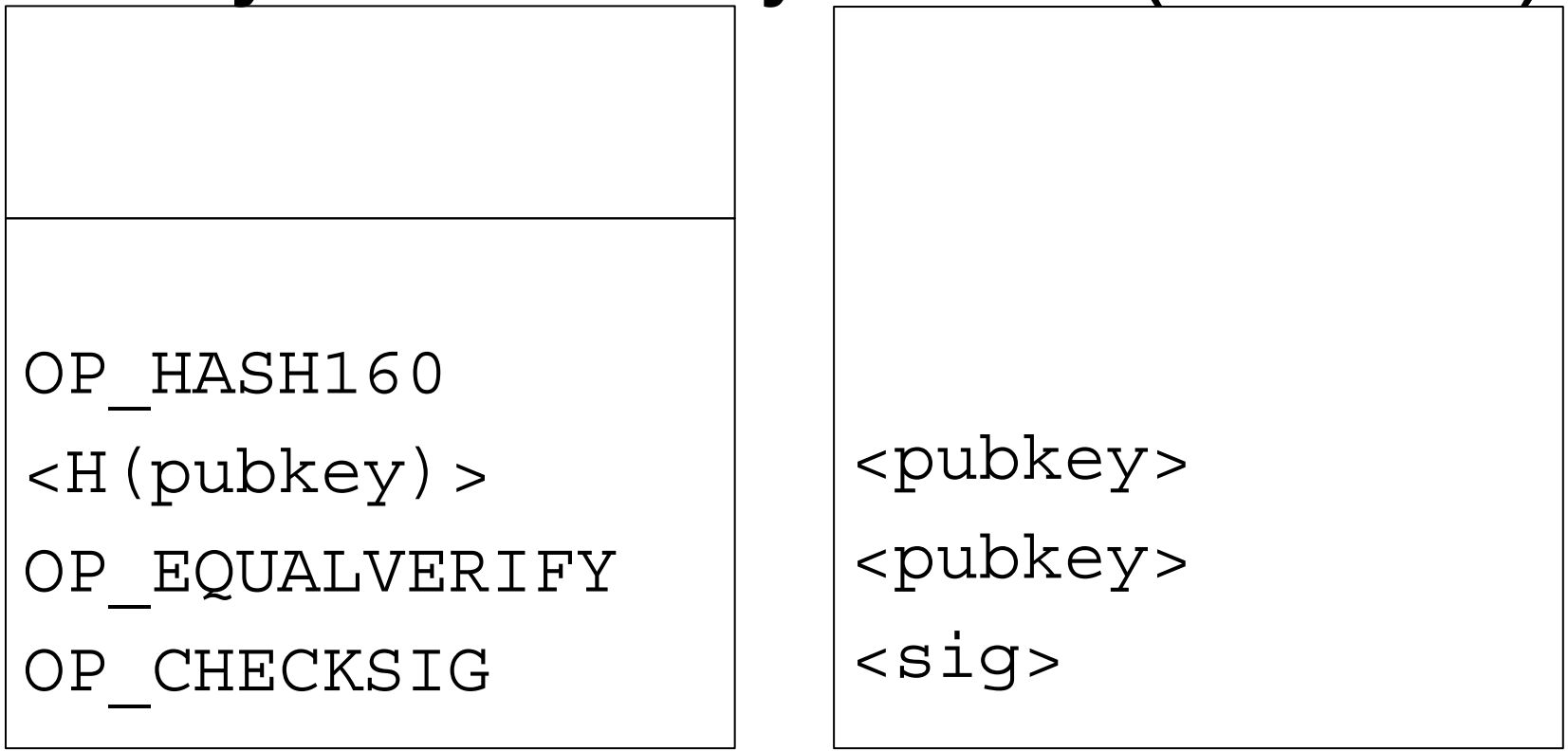
Pay to Pubkey Hash (P2PKH)



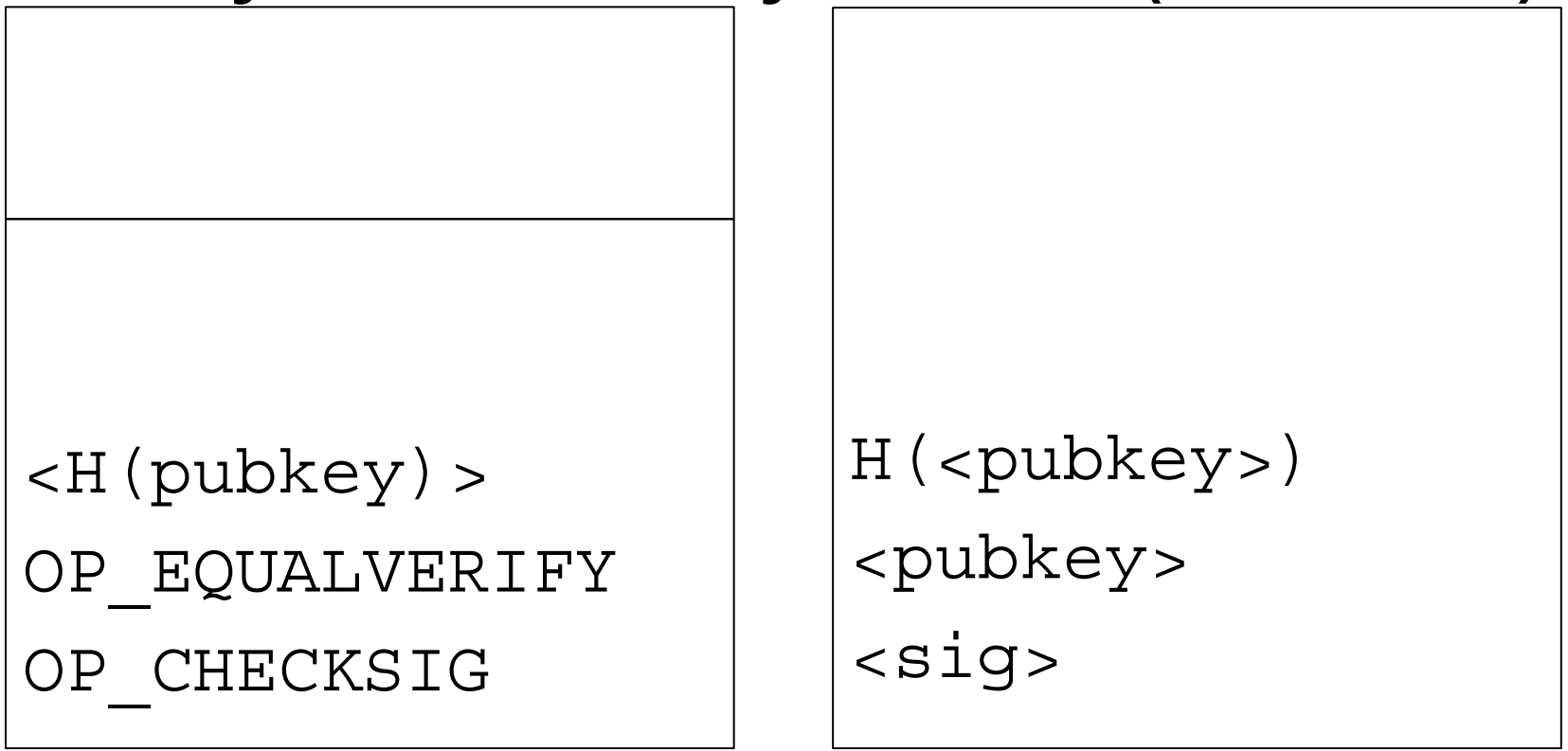
Pay to Pubkey Hash (P2PKH)



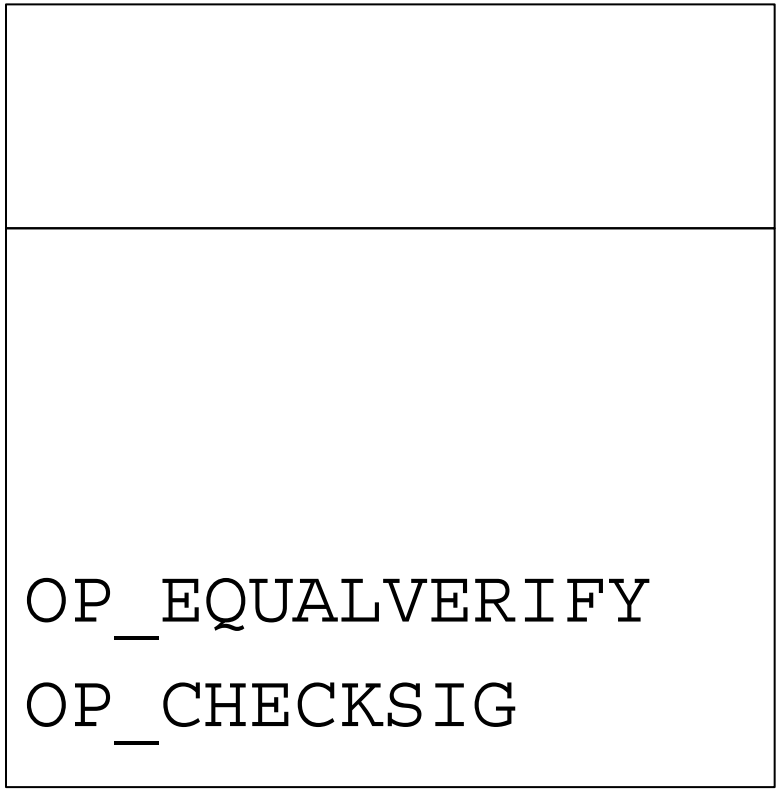
Pay to Pubkey Hash (P2PKH)

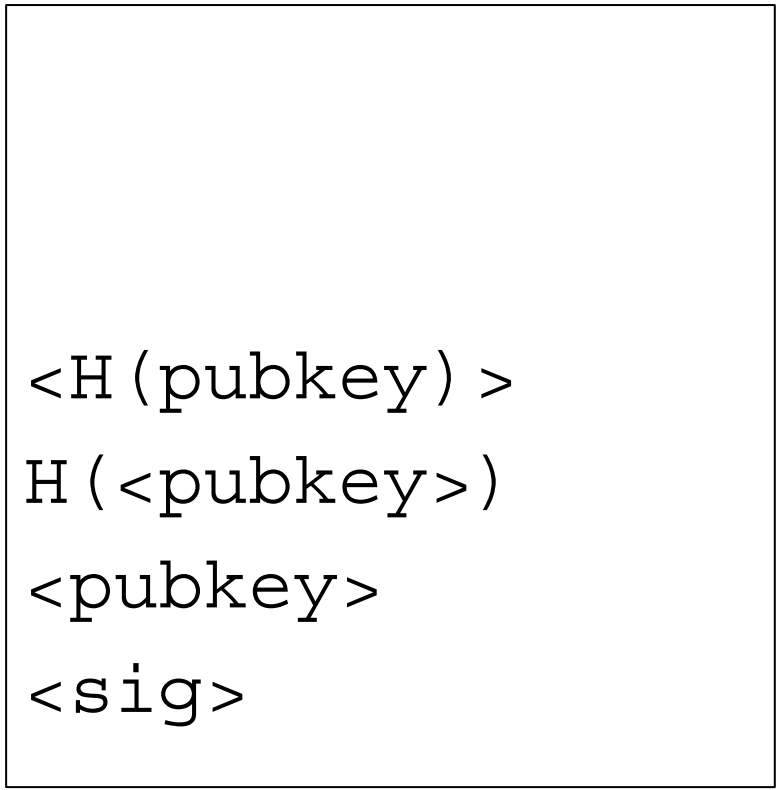


Pay to Pubkey Hash (P2PKH)

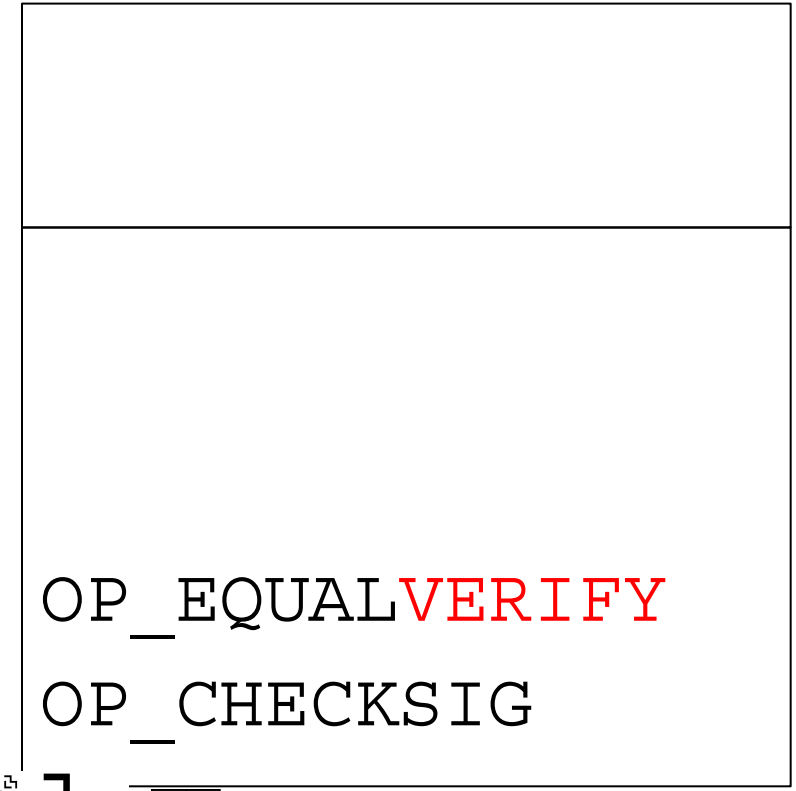


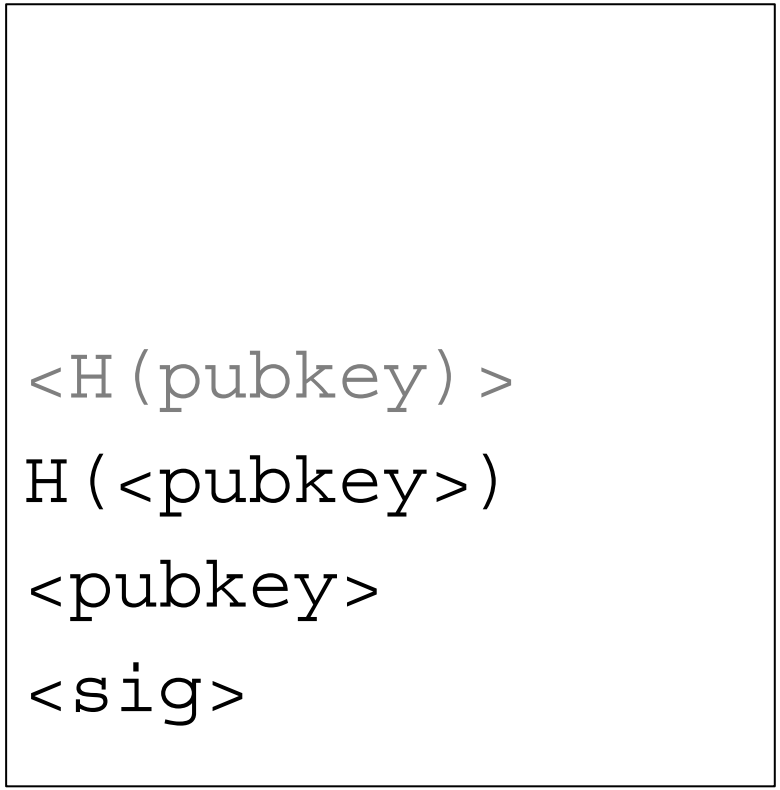
Pay to Pubkey Hash (P2PKH)



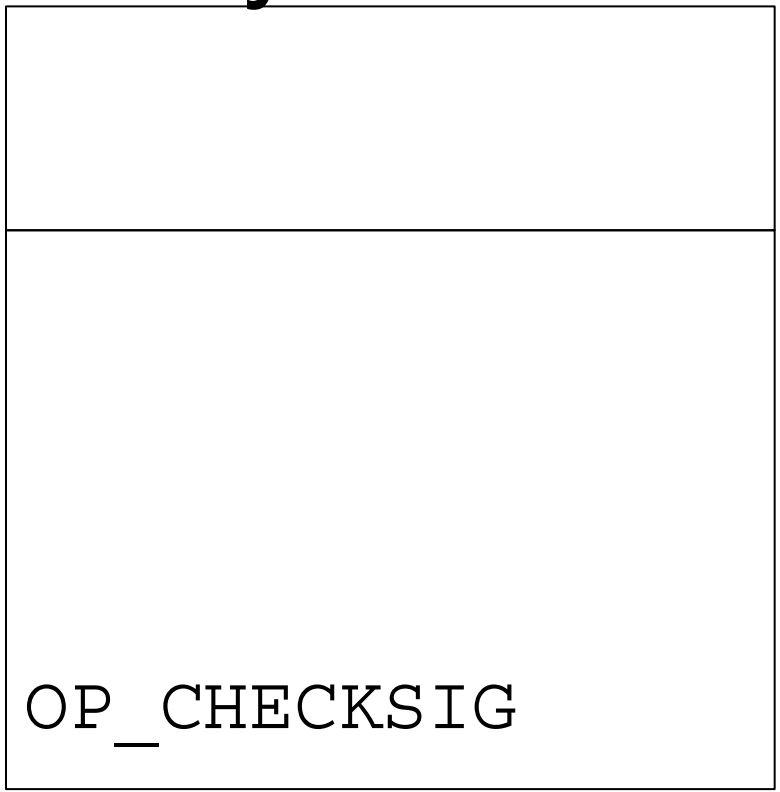


Pay to Pubkey Hash (P2PKH)

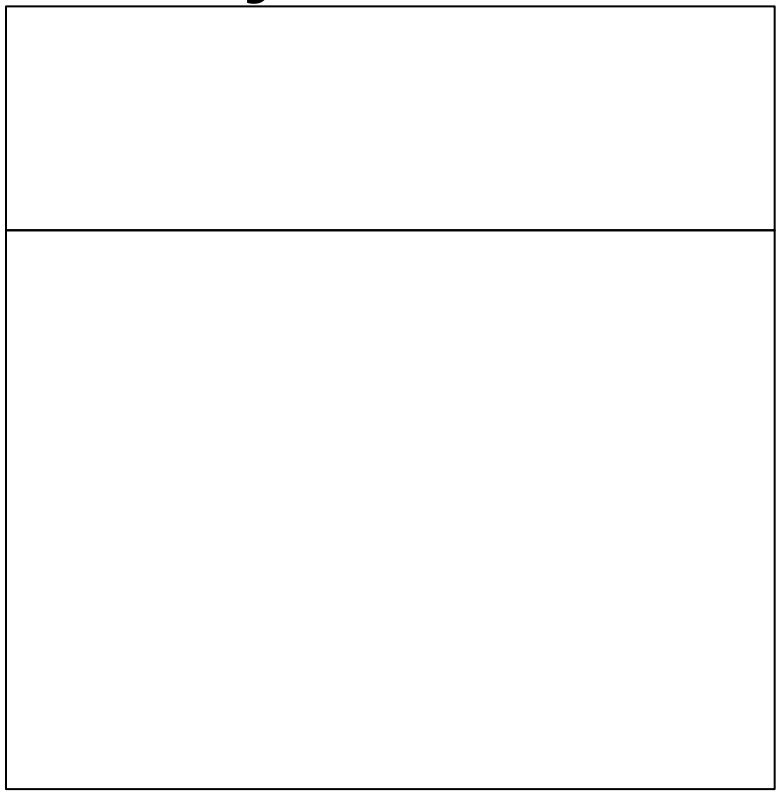


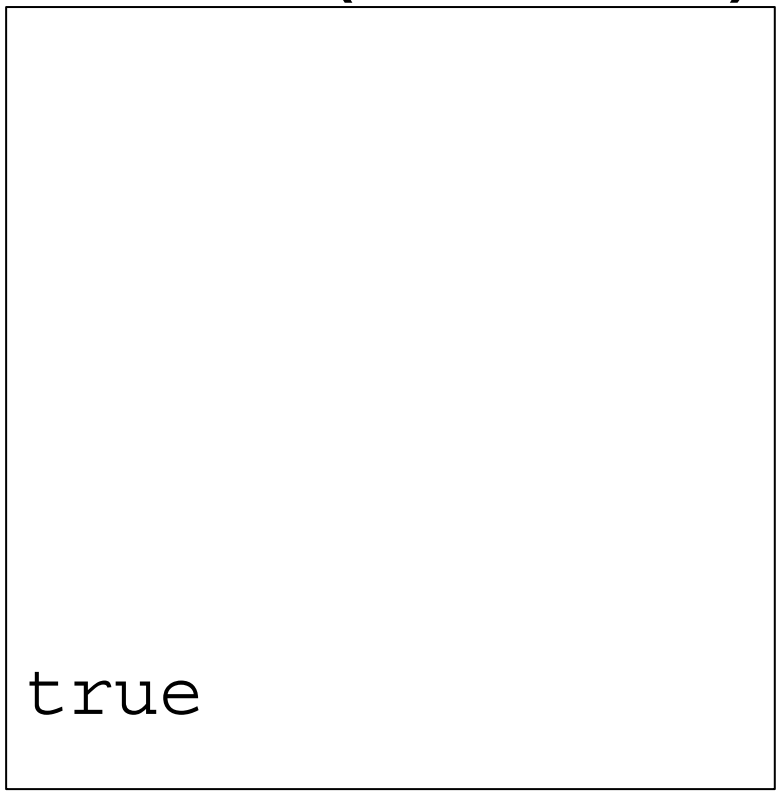


Pay to Pubkey Hash (P2PKH)

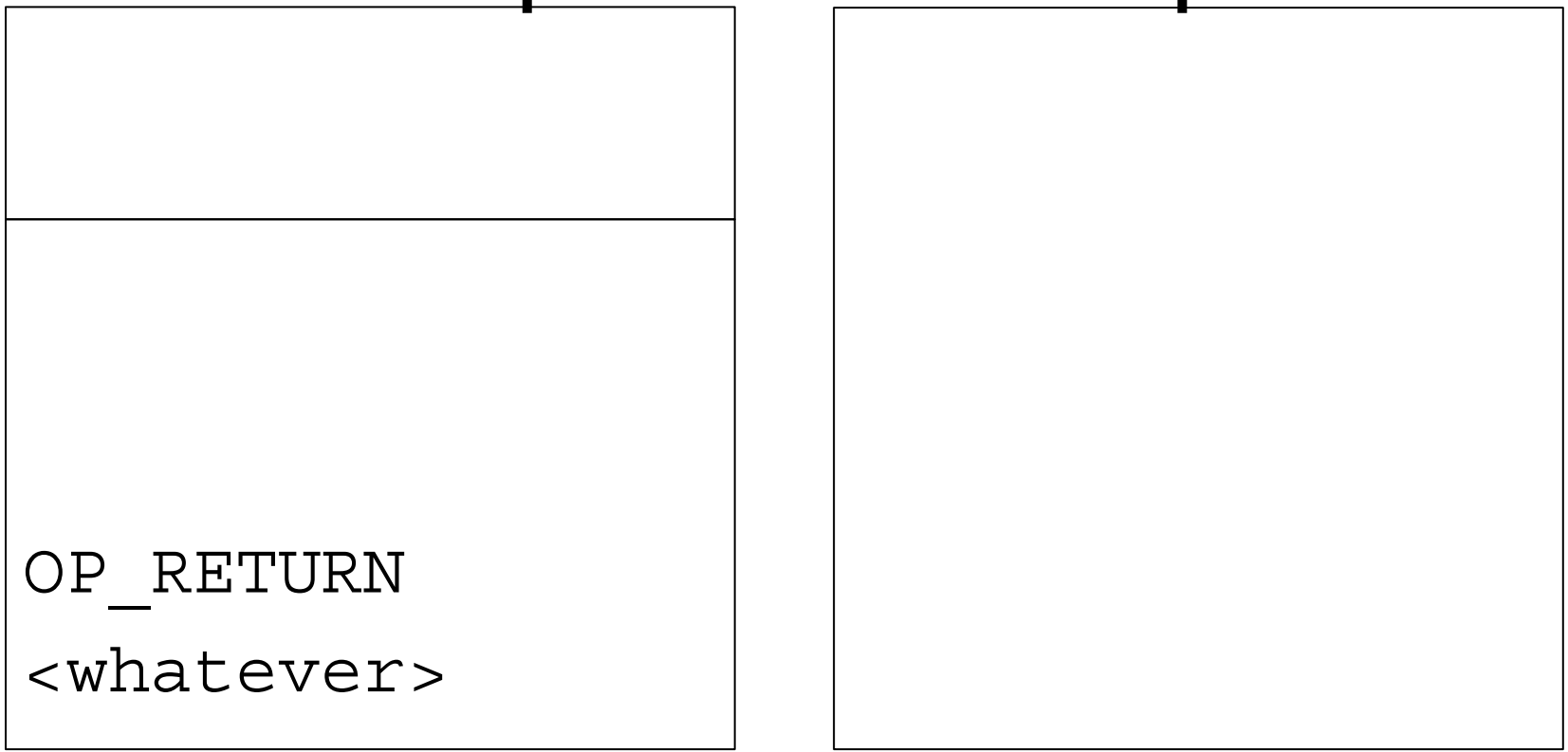


Pay to Pubkey Hash (P2PKH)

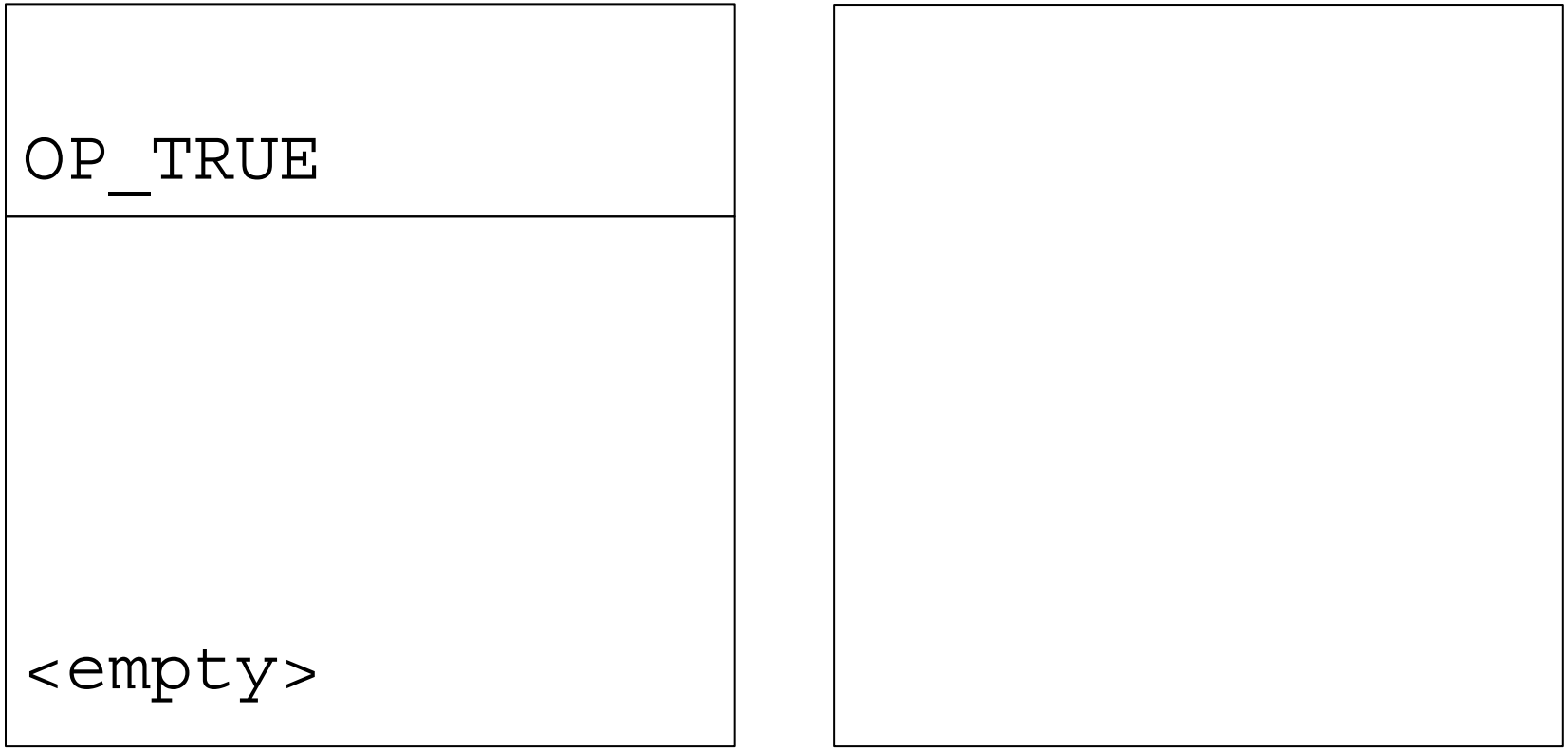




Unspendable output



Anyone can spend output



Other types of Bitcoin scripts

Type	ScriptPubKey
Pay to script hash (P2SH)	`OP_HASH160 <ScriptHash> OP_EQUAL`
Multisignature	`OP_n <PubKey1> <PubKey2> ... <PubKeyN> OP_m OP_CHECKMULTISIG`
Pay to taproot (P2TR)	`OP_1 <TaprootPubKey>`