LockTimeを利用した P2SH

@DG Lab Nakagawa

Locktimeとは？

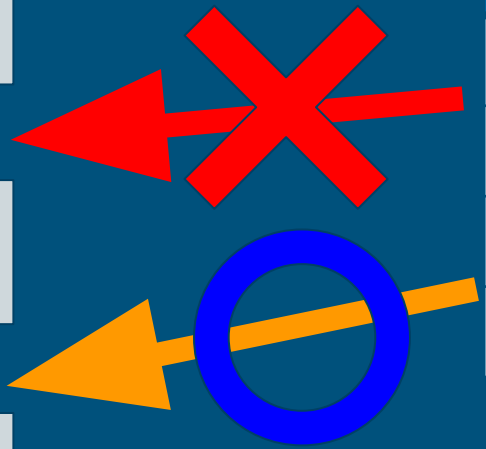
・Transactionに設定するlocktimeは、
ある**「ブロック高」**or**「Unixタイムスタンプ」**
に達するまでそのTransactionを利用できない
・通常は、すぐ使いたいので0（0x00）を設定
・locktimeはインプットのsequenceがMAX（0xffffffff）以外
の時有効
・値が、500,000,000までは、ブロック高
それ以上の場合は、Unixタイムスタンプとなる

Locktimeとは？

Block 199

Block 200

Block 201



Tx	Transaction
version	1
txins	インプット
txouts	アウトプット
locktime	200

※：インプットのSequenceがMAX以外の時、locktime有効

Agenda

- OP_CHECKLOCKTIMEVERIFY
- OP_CHECKSEQUENCEVERIFY

概要

「OP_CHECKLOCKTIMEVERIFY」と
「OP_CHECKSEQUENCEVERIFY」
は、P2SHなどで使われる、
ScriptのOPコードで、
locktimeなどの制御が可能となる

OP_CHECKLOCKTIMEVERIFY

OP_CHECKLOCKTIMEVERIFY

- OP_CHECKLOCKTIMEVERIFYとは

Transaction（以下、Tx）の「locktime」の
制限を設ける事が出来る！

OP_CHECKLOCKTIMEVERIFY

・Script定義

name	description
Word	OP_CHECKLOCKTIMEVERIFY (previously OP_NOP2)
Opcode（Hex）	177（0xb1）
Input	x
Output	x / fail

OP_NOP2のなごりの為、成功時にPOPした値をPUSHする為、すぐ後にOP_DROPなどを入れる必要がある

OP_CHECKLOCKTIMEVERIFY

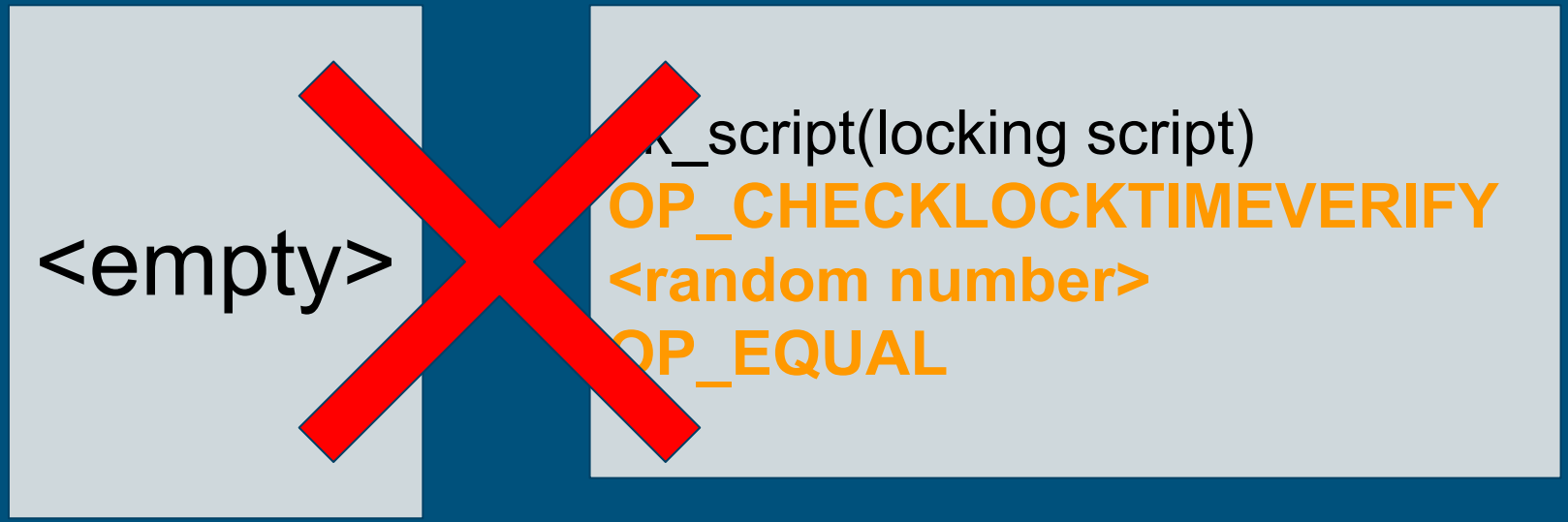
BIP:65より

- 以下の場合fail（エラー）となる
  - スタックが空の場合
  - POPした値がマイナスの場合
  - POPした値のタイプが異なる場合
    （ブロック高、Unixタイムスタンプ）
  - POPした値がTxのlocktimeがより大きい場合
  - TxインプットのSequenceが0xffffffffの場合

OP_CHECKLOCKTIMEVERIFY

BIP:65より

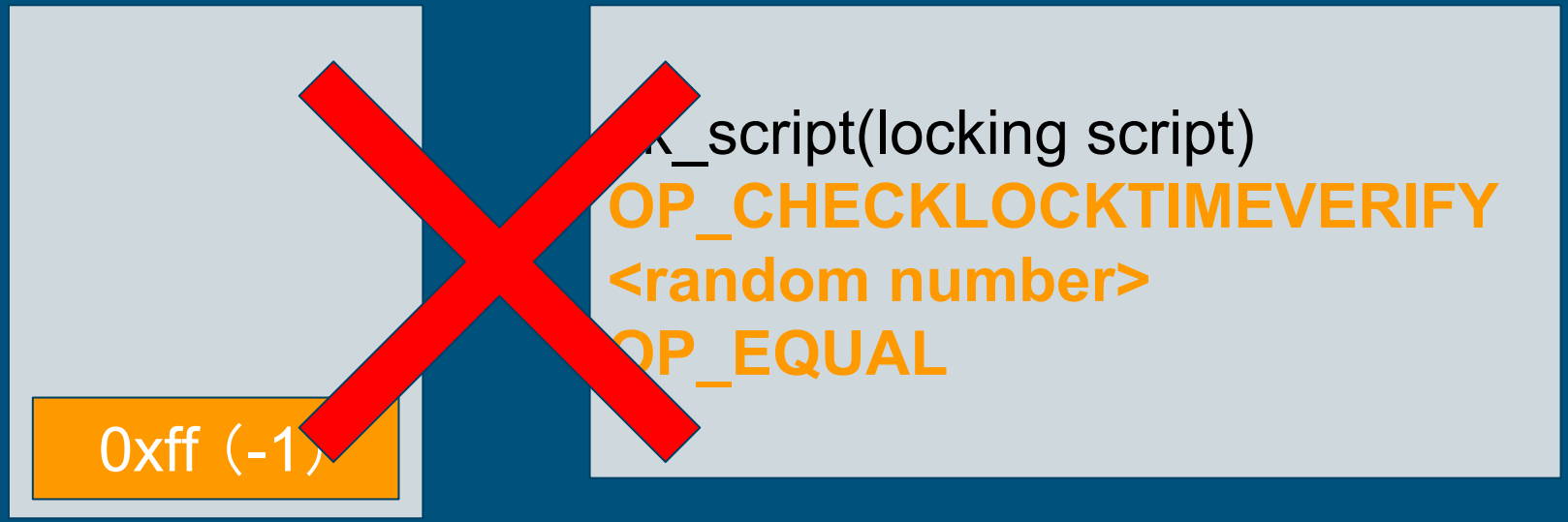
・スタックが空の場合



OP_CHECKLOCKTIMEVERIFY

BIP:65より

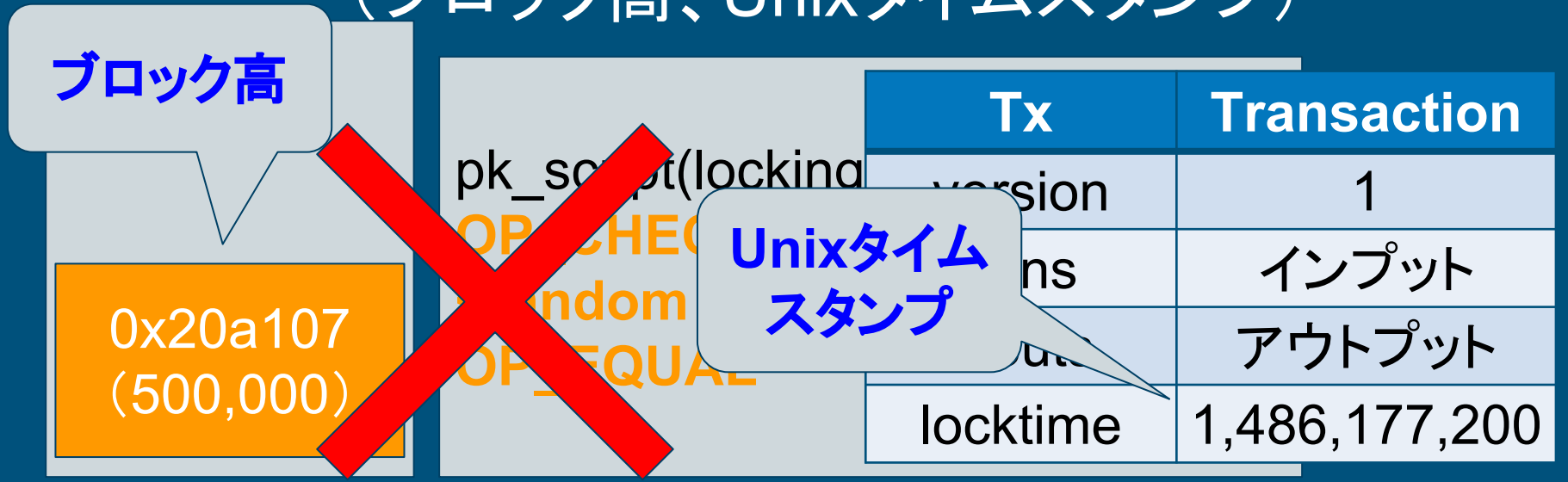
・POPした値がマイナスの場合



OP_CHECKLOCKTIMEVERIFY

BIP:65より

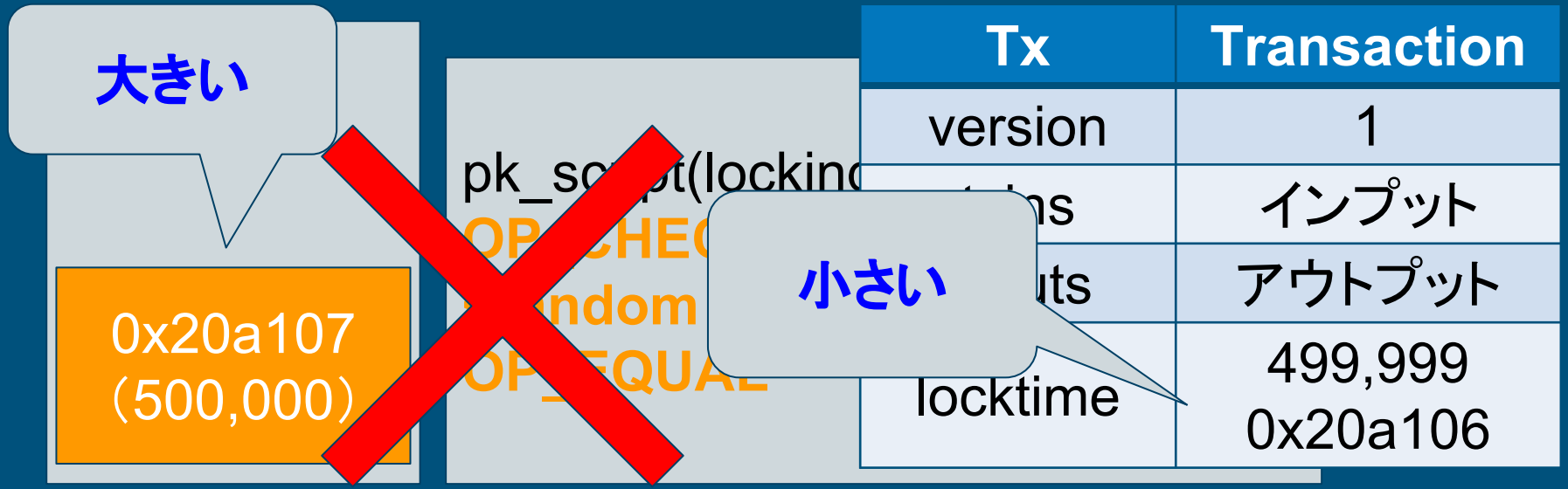
・POPした値のタイプが異なる場合
（ブロック高、Unixタイムスタンプ）



OP_CHECKLOCKTIMEVERIFY

BIP:65より

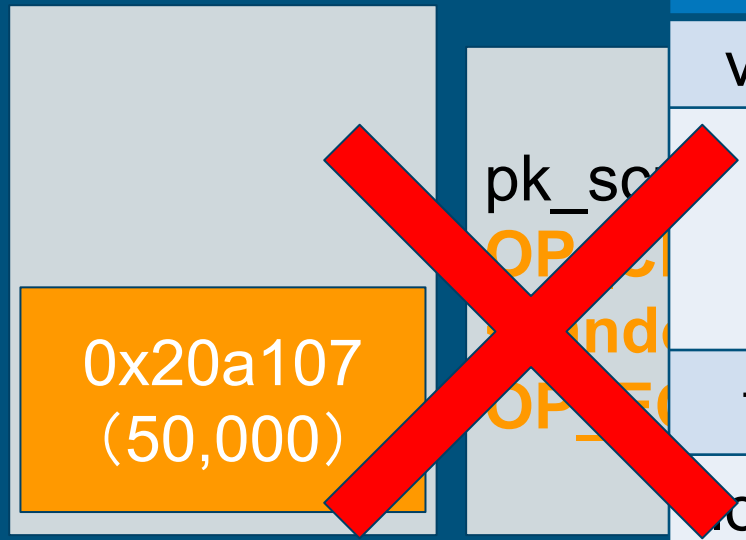
・POPした値がTxのlocktimeがより大きい場合



OP_CHECKLOCKTIMEVERIFY

BIP:65より

・TxインプットのSequenceが0xffffffffの場合



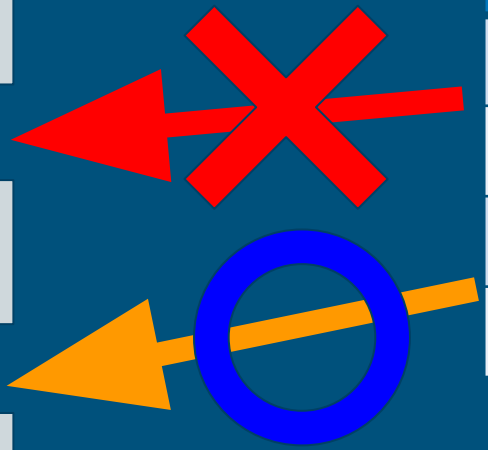
Tx	Transaction
version	1
txins	インプット txin[0] > Sequence **0xffffffff**
txouts	アウトプット
locktime	50,000

OP_CHECKLOCKTIMEVERIFY

Block 499,999

Block 500,000

Block 500,001



Tx	Transaction
version	1
txins	インプット
txouts	アウトプット
locktime	500,000

OP_CHECKLOCKTIMEVERIFY

- まとめ
  - locktimeの下限値を決める事ができる
  - 昔のopcodeのなごりが残っている為、
    OP_DROPを付加しなければならない

OP_CHECKSEQUENCEVERIFY

OP_CHECKSEQUENCEVERIFY

- OP_CHECKSEQUENCEVERIFYとは

Transaction（以下、Tx）を送信してからの
「時間（秒）」or「ブロック高」の
制限を設ける事が出来る！

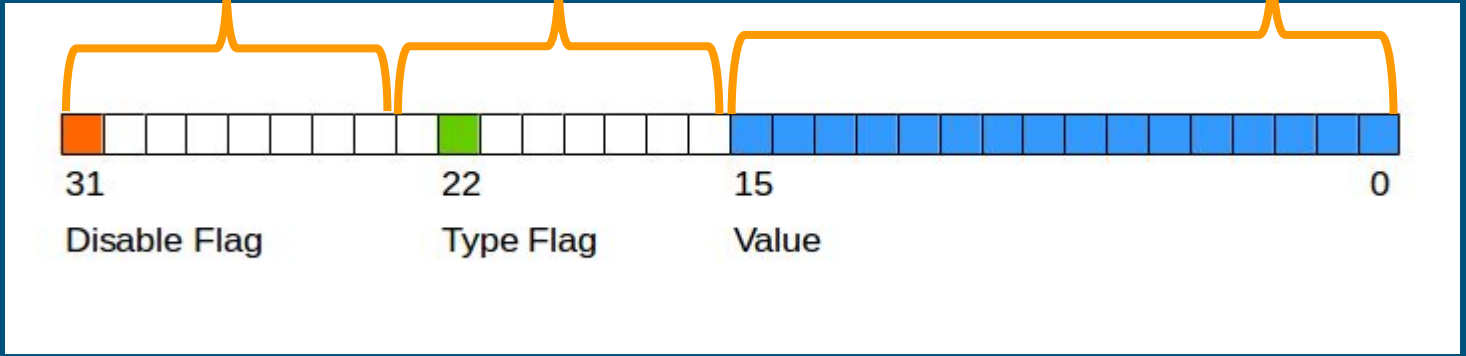
OP_CHECKSEQUENCEVERIFY

・TransactionインプットのSequence

使用不可フラグ
使用する場合0x00など
使用付加の場合0x80など

Typeフラグ
時間（秒）の場合、0x40など
ブロック高の場合0x00など

値
uint16
ブロック高：そのまま
時間：値×512（秒）



画像　https://github.com/bitcoin/bips/blob/master/bip-0068/encoding.png

OP_CHECKSEQUENCEVERIFY

・Script定義

name	description
Word	OP_CHECKSEQUENCEVERIFY (previously OP_NOP3)
Opcode（Hex）	178（0xb2）
Input	x
Output	x / fail

OP_NOP3のなごりの為、成功時にPOPした値をPUSHする為、
すぐ後にOP_DROPなどを入れる必要がある

OP_CHECKSEQUENCEVERIFY

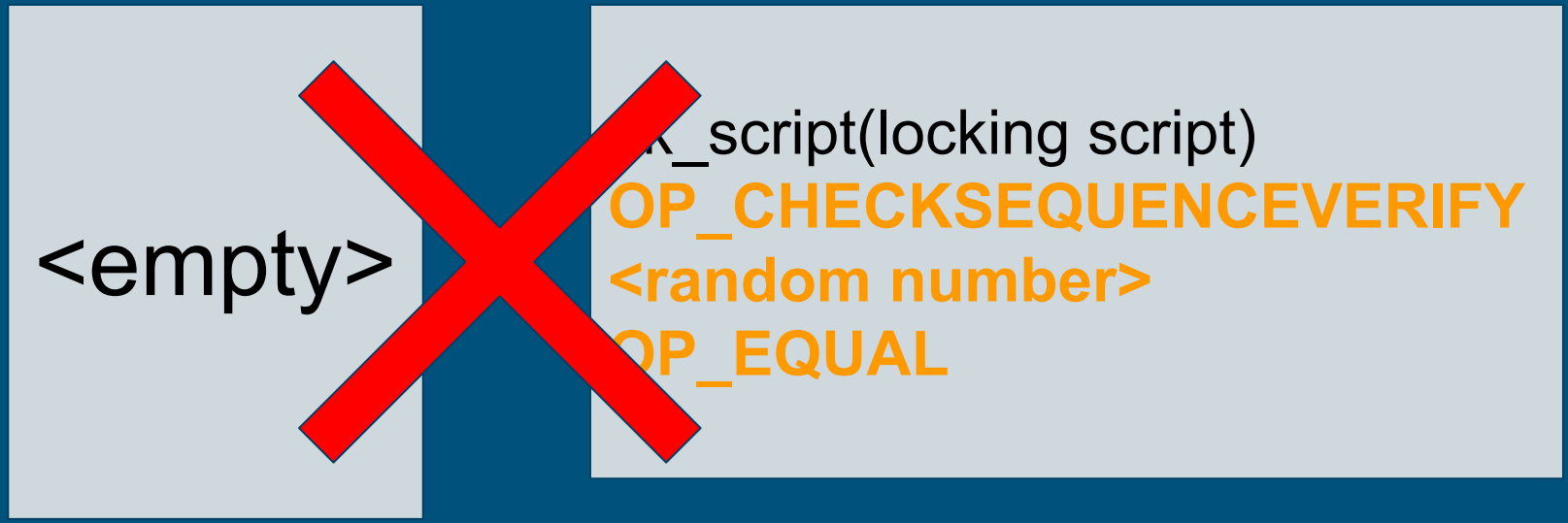
BIP：112より

- 以下の場合fail（エラー）となる
  - スタックが空の場合
  - POPした値がマイナスの場合
  - POPした値に使用不可フラグある場合
    - Txのversionが2より小さい場合
    - インプットのsequenceに使用不可フラグがある場合
    - POPした値のタイプが異なる場合（時間（秒）、ブロック高）
    - POPした値がインプットのsequenceより大きい場合

OP_CHECKSEQUENCEVERIFY

BIP：112より

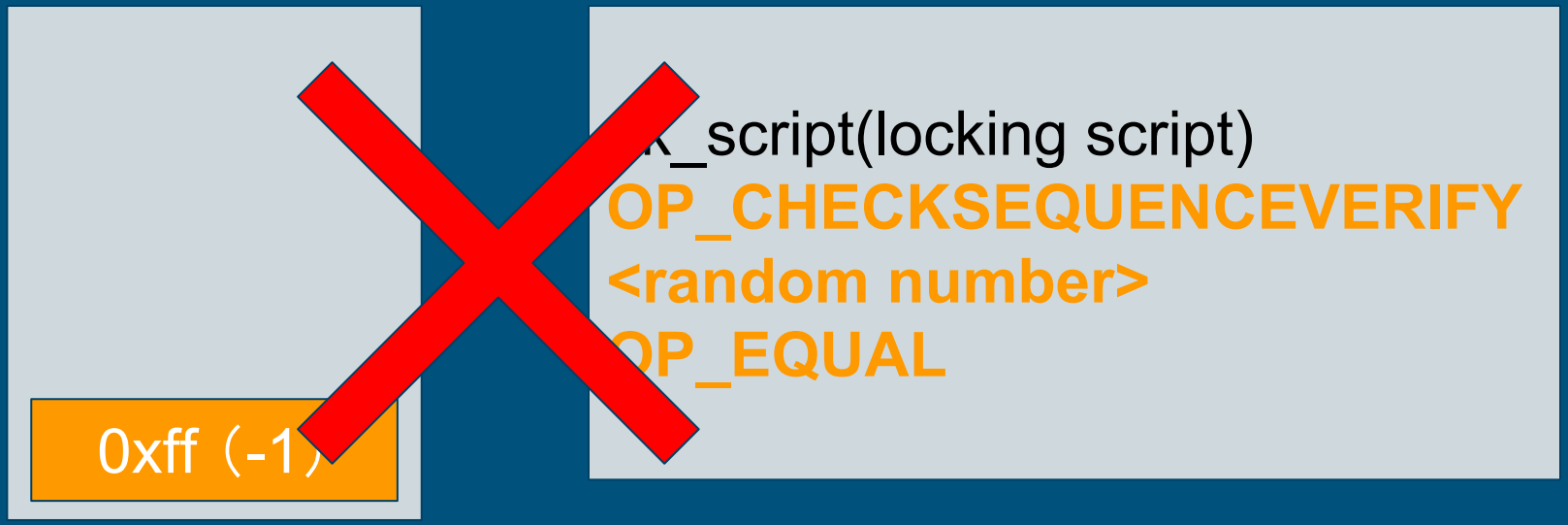
・スタックが空の場合



OP_CHECKSEQUENCEVERIFY

BIP：112より

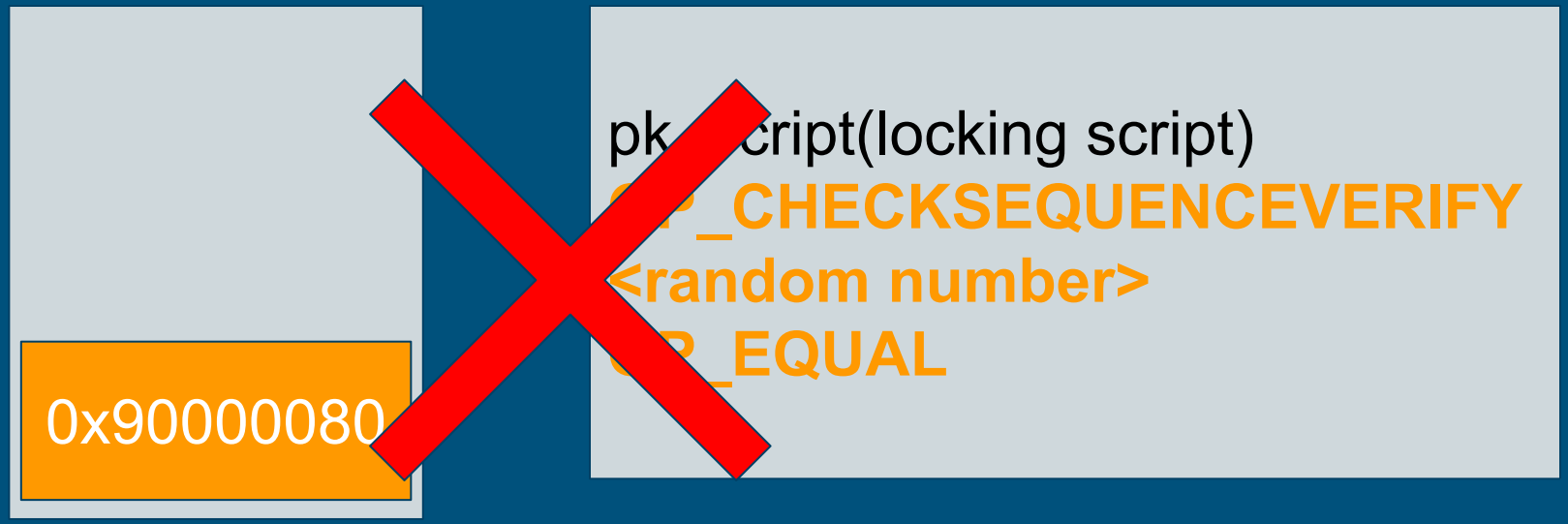
・POPした値がマイナスの場合



OP_CHECKSEQUENCEVERIFY

BIP：112より

・POPした値に使用不可フラグある場合



OP_CHECKSEQUENCEVERIFY

BIP：112より

・Txのversionが2より小さい場合

Tx	Transaction
version	1
txins	インプット txin[0] > Sequence 0x90000000
txouts	アウトプット
locktime	0

OP_CHECKSEQUENCEVERIFY

BIP:111より

・インプットのsequenceに使用不可フラグがある場合

Tx	Transaction
version	2
txins	インプット txin[0] > Sequence **0x90000080**
txouts	アウトプット
locktime	0

OP_CHECKSEQUENCEVERIFY

BIP：112より

・POPした値のタイプが異なる場合（時間（秒）、ブロック高）

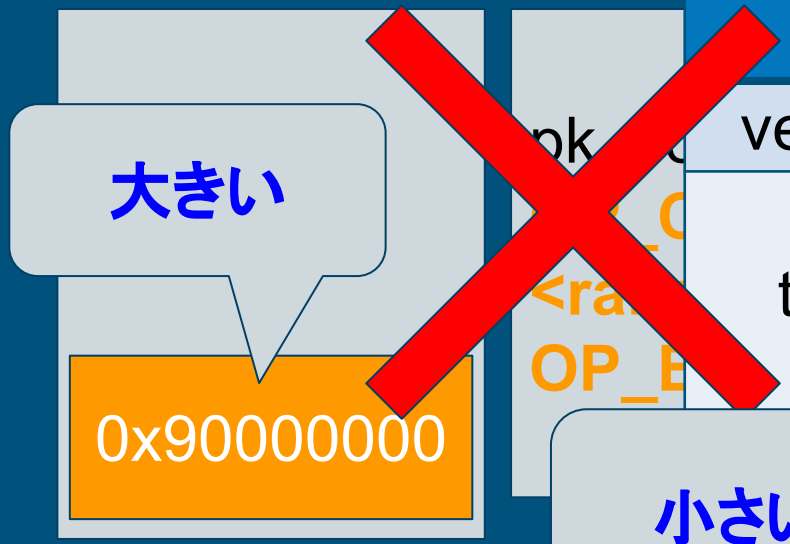
Tx	Transaction
version	2
txins	インプット txin[0] > Sequence **0xa9004000**
	アウトプット
e	0

時間（秒）

OP_CHECKSEQUENCEVERIFY

BIP：112より

・POPした値がインプットのsequenceより大きい場合



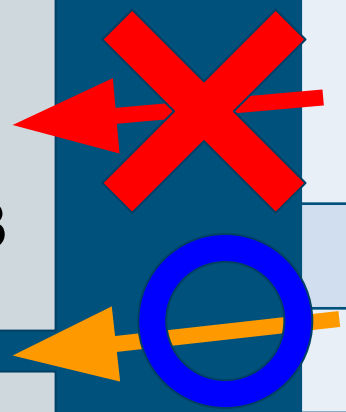
Tx	Transaction
version	2
txins	インプット txin[0] > Sequence **0x8f000000**
	アウトプット
	0

OP_CHECKSEQUENCEVERIFY

Block 500,000

Block 500,001
:
Block 500,143

Block 500,144



Tx	Transaction
version	2
txins	インプット txin[0] > Sequence **0x90000000**
txouts	アウトプット
locktime	0

OP_CHECKSEQUENCEVERIFY

- まとめ
  - ブロックに入ってから、使えるまでの期間

    （時間（秒）orブロック高）の下限値を決める事ができる
  - 昔のopcodeのなごりが残っている為、

    OP_DROPを付加しなければならない

まとめ

まとめ

Transactionに設定するlocktimeは、送信側が自由に設定出来る

「OP_CHECKLOCKTIMEVERIFY」

「OP_CHECKSEQUENCEVERIFY」

は、使う時に下限値の制限を付けることが出来る

利用例

利用例

・Tumblebit
・Lightning
などで使われるので、
今後のセッションで説明予定！

演習

演習

URLを開いてください。

http://www.bc-2.jp/tools/index.html

すでに開いている人は、SegwitがONになっているのでF5などでリセットしてください。

演習

$ ./bitcoin-cli sendrawtransaction <data>

上記コマンドを送った場合、
「OP_CHECKSEQUENCEVERIFY」で失敗する事を確かめよう！

演習（流れ）

・P2SH（OP_CHECKSEQUENCEVERIFYを含む）に送信
　→成功
・P2SHから、P2PKHに送信
　→失敗（指定したブロック数以下の場合）
　→成功（指定したブロック数以上の場合）

演習（redeem）

このpk_script,locking scriptを持つTransactionが入っているBlockから5Block後でないと使用出来ない

P2SH - redeem

OP_5

OP_CHECKSEQUENCEVERIFY

OP_DROP

<random 32byte>

OP_EQUAL

※今回は、OP_CSV検証の為、署名を使わないP2SHにしています

演習

・準備（ランダム数取得）

$ head -c 512 /dev/urandom | shasum -a 256

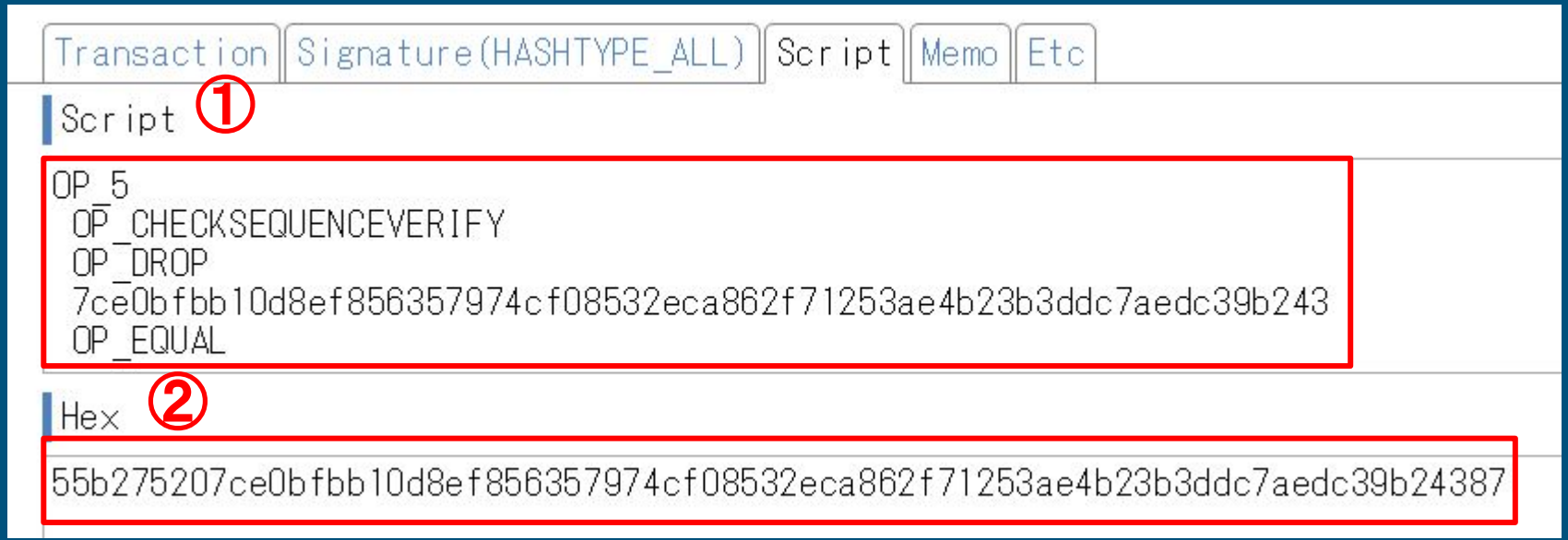
7ce0bfbb10d8ef856357...4b23b3ddc7aedc39b243 -

＜random 32byte>（先頭64文字分）

演習

・Script作成

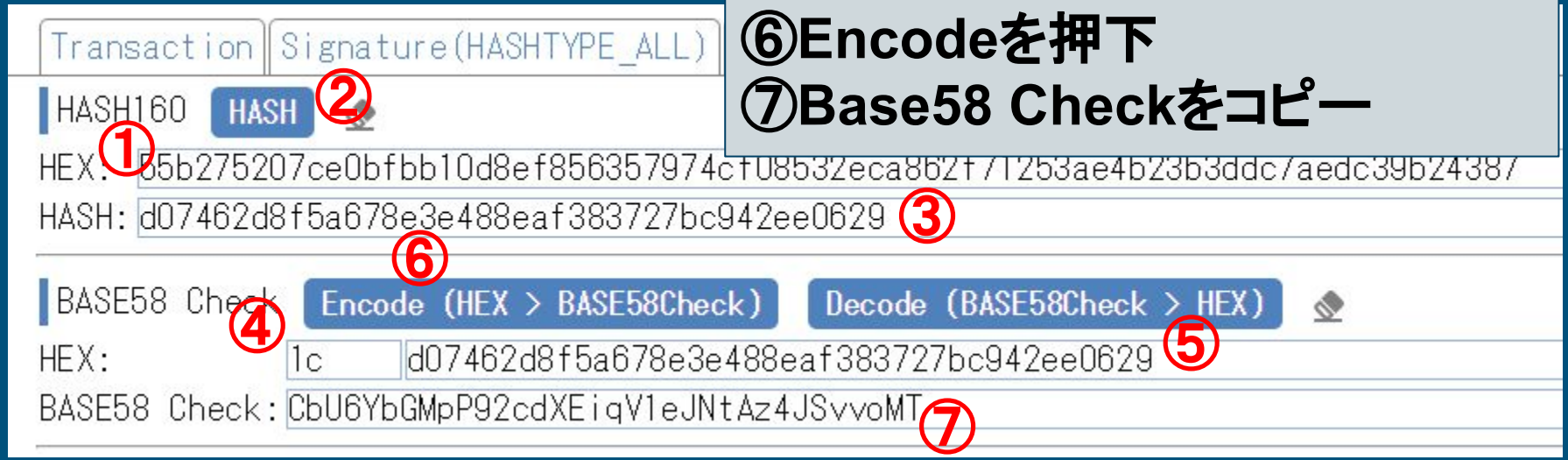
①Scriptを設定
②作成したHEXをコピー



演習

・P2SHアドレス作成

①コピーしたHEXを貼り付ける
②HASHボタン押下
③HASH値をコピー
④1cを設定（P2SH）
⑤「③」を貼り付ける
⑥Encodeを押下
⑦Base58 Checkをコピー



演習

<P2SH address> 「C」から始まるアドレス
<value> 自身のUTXOから少し

・P2SHに送信

$ ./bitcoin-cli sendtoaddress <P2SH address> <value>
<txid>

例

$ ./bitcoin-cli sendtoaddress
CbU6YbGMpP92cdXEiqV1eJNtAz4JSvvoMT 0.1
<txid>

成功！<txid>をコピー！

演習

- 確認

$ ./bitcoin-cli getrawtransaction <txid> 1

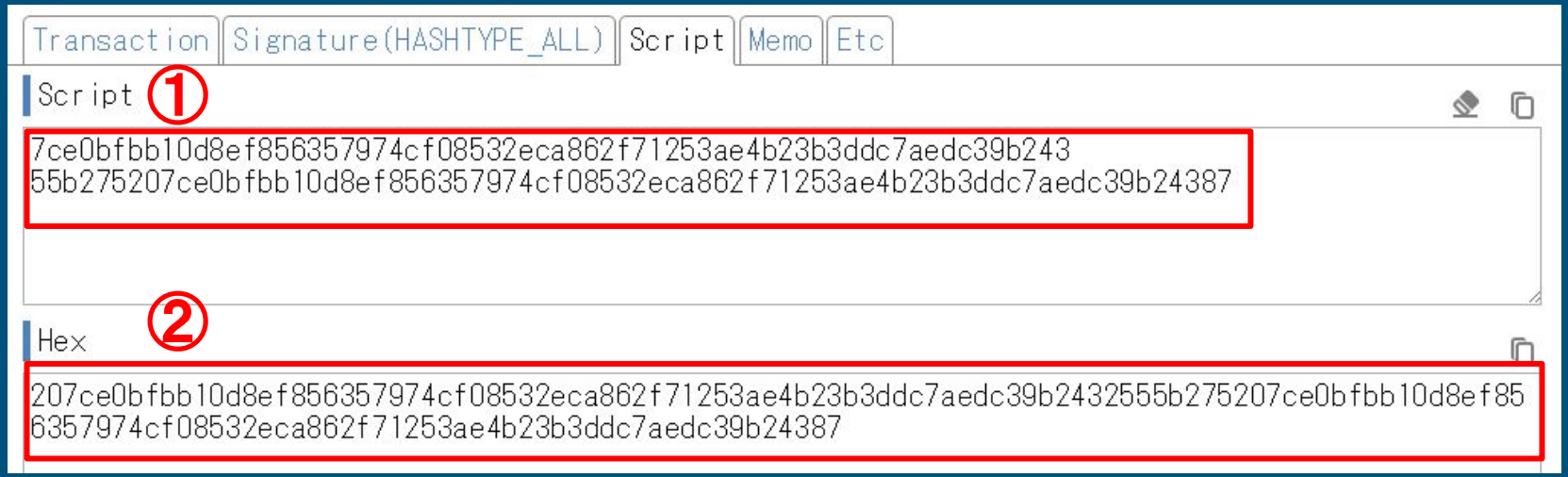
以下のような、情報が付与されればBlock化している

```
"blockhash": "00000032bf881….dbfb960a84e1a2da",
  "confirmations": 1,
  "time": 1484095054,
  "blocktime": 1484095054
```

演習

①<random 32byte>
　<redeem>を設定
②作成したHEXをコピー

・signature script（unlocking script）生成
　<random 32byte>
　<redeem>



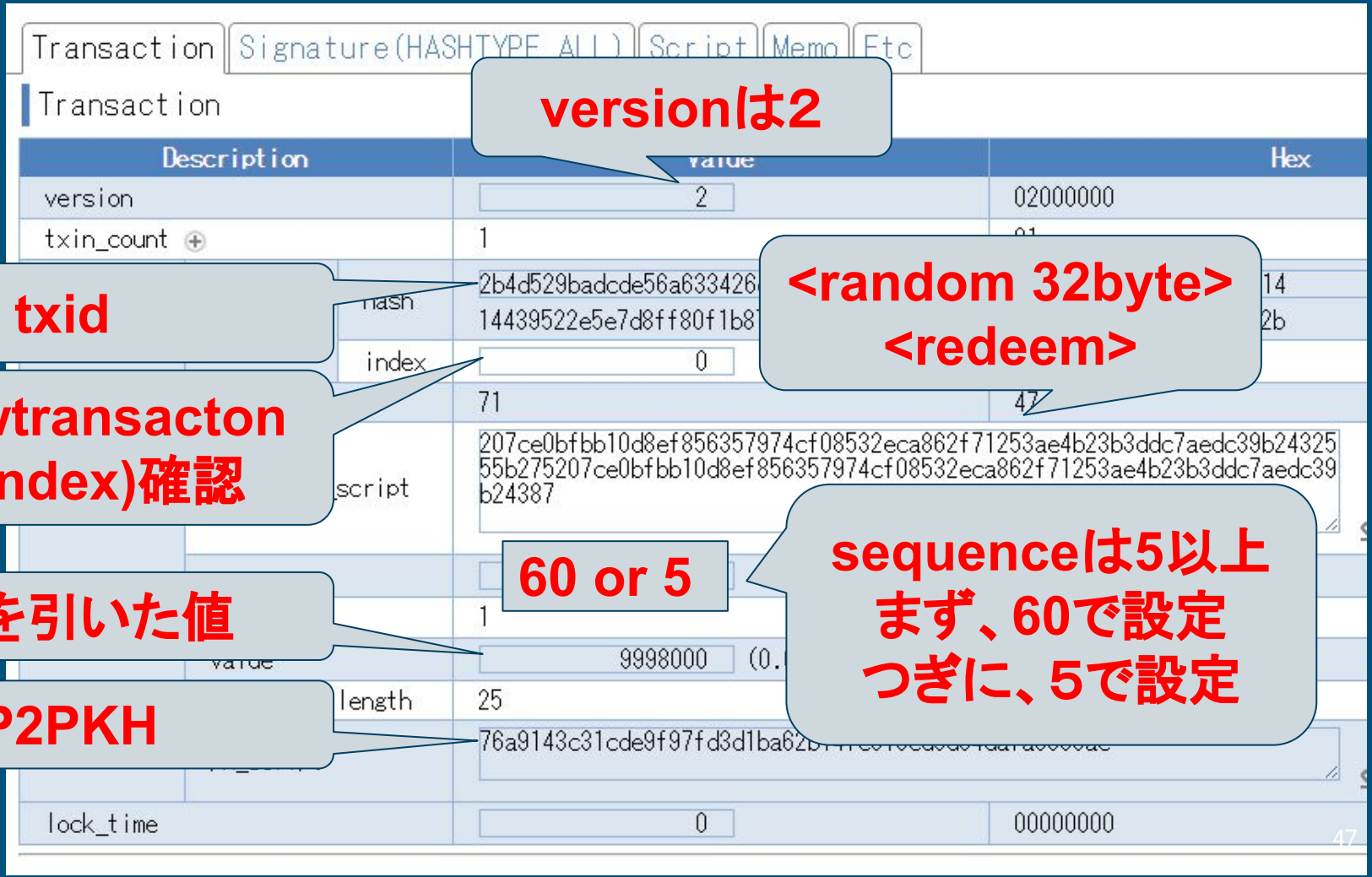
演習

・送信先取得
$ ./bitcoin-cli getnewaddress
<new address>

$ ./bitcoin-cli validateaddress <new address>
<address infos>

の「scriptPubKey」をコピー

演習


Transaction
Signature(HASHTYPE_ALL)
Script
Memo
Etc
Transaction
versionは2
Description
Value
Hex
version
2
02000000
txin_count
1
txid
<random 32byte>
<redeem>
index
0
getrawtransacton
でn(index)確認
71
47
207ce0bfbb10d8ef856357974cf08532eca862f71253ae4b23b3ddc7aedc39b24325
55b275207ce0bfbb10d8ef856357974cf08532eca862f71253ae4b23b3ddc7aedc39
b24387
60 or 5
sequenceは5以上
まず、60で設定
つぎに、5で設定
1
feeを引いた値
9998000
length
25
P2PKH
76a9143c31cde9f97fd3d1ba62
lock_time
0
00000000

演習

・P2PKHに送信

$ ./bitcoin-cli sendrawtransaction <data>

error code: -26

error message:

64: non-BIP68-final

失敗

※：60の場合は上記のようなエラーがでます。
5の場合は正常に送信され、txidが出来ると思います。

参考資料

- BIP: 65 OP_CHECKLOCKTIMEVERIFY

https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki

- BIP: 68 Relative lock-time using consensus-enforced sequence numbers

https://github.com/bitcoin/bips/blob/master/bip-0068.mediawiki

- BIP: 112 CHECKSEQUENCEVERIFY

https://github.com/bitcoin/bips/blob/master/bip-0112.mediawiki

takatoshi@dglab.com