Opérateurs, constantes et symboles du langage de script de transaction

2022-03-13

Tableaux et descriptions provenant de <https://en.bitcoin.it/wiki/Script>.

<<tx\_script\_ops\_table\_pushdata> > affiche les opérateurs pour pousser les valeurs sur la pile.

Pousser la valeur sur la pile

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_0 ou OP\_FALSE | 0x00 | Un tableau vide est poussé sur la pile |
| 1 à 75 | 0x01 à 0x4b | Poussez les N octets suivants sur la pile, où N est compris entre 1 et 75 octets |
| OP\_PUSHDATA1 | 0x4c | L'octet de script suivant contient N, poussez les N octets suivants sur la pile |
| OP\_PUSHDATA2 | 0x4d | Les deux octets de script suivants contiennent N, poussez les N octets suivants sur la pile |
| OP\_PUSHDATA4 | 0x4e | Les quatre octets de script suivants contiennent N, poussez les N octets suivants sur la pile |
| OP\_1NEGATE | 0x4f | Poussez la valeur "–1" sur la pile |
| OP\_RESERVED | 0x50 | Halt - Transaction non valide à moins qu'elle ne se trouve dans une clause OP\_IF non exécutée |
| OP\_1 ou OP\_TRUE | 0x51 | Poussez la valeur "1" sur la pile |
| OP\_2 à OP\_16 | 0x52 à 0x60 | Pour OP\_N, poussez la valeur "N" sur la pile, par exemple, OP\_2 pousse "2" |

<<tx\_script\_ops\_table\_control> > affiche les opérateurs de contrôle de flux conditionnel.

Contrôle de flux conditionnel

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_NOP | 0x61 | Ne fais rien |
| OP\_VER | 0x62 | Halt - Transaction non valide à moins qu'elle ne se trouve dans une clause OP\_IF non exécutée |
| OP\_IF | 0x63 | Exécute les instructions suivantes si le haut de la pile n'est pas 0 |
| OP\_NOTIF | 0x64 | Exécute les instructions suivantes si le haut de la pile est 0 |
| OP\_VERIF | 0x65 | Interrompre—Transaction invalide |
| OP\_VERNOTIF | 0x66 | Interrompre—Transaction invalide |
| OP\_ELSE | 0x67 | Exécuter uniquement si les instructions précédentes n'ont pas été exécutées |
| OP\_ENDIF | 0x68 | Terminer le bloc OP\_IF, OP\_NOTIF, OP\_ELSE |
| OP\_VERIFY | 0x69 | Vérifiez le haut de la pile, arrêtez et invalidez la transaction si elle n'est pas VRAIE |
| OP\_RETURN | 0x6a | Arrêter et invalider la transaction |

<<tx\_script\_ops\_table\_timelock> > affiche les opérateurs utilisés pour les timelocks.

Opérations Timelock

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_CHECKLOCKTIMEVERIFY (précédemment OP\_NOP2) | 0xb1 | Marque la transaction comme non valide si l'élément supérieur de la pile est supérieur au champ nLockTime de la transaction, sinon l'évaluation du script continue comme si un OP\_NOP avait été exécuté. La transaction est également invalide si 1. la pile est vide ; ou 2. l'élément de la pile supérieure est négatif ; ou 3. l'élément de la pile supérieure est supérieur ou égal à 500000000 alors que le champ nLockTime de la transaction est inférieur à 500000000, ou vice versa ; ou 4. le champ nSequence de l'entrée est égal à 0xffffffff. La sémantique précise est décrite dans la BIP-65 |
| OP\_CHECKSEQUENCEVERIFY (précédemment OP\_NOP3) | 0xb2 | Marque la transaction comme non valide si le temps de verrouillage relatif de l'entrée (appliqué par BIP 0068 avec nSequence) n'est pas égal ou supérieur à la valeur de l'élément de la pile supérieure. La sémantique précise est décrite dans BIP-112 |

<<tx\_script\_ops\_table\_stack> > montre les opérateurs utilisés pour manipuler la pile.

Opérations de pile

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_TOALTSTACK | 0x6b | Sortez l'élément supérieur de la pile et poussez vers une autre pile |
| OP\_FROMALTSTACK | 0x6c | Sortez l'élément supérieur de la pile alternative et poussez pour empiler |
| OP\_2DROP | 0x6d | Pop top deux éléments de la pile |
| OP\_2DUP | 0x6e | Dupliquer les deux éléments du haut de la pile |
| OP\_3DUP | 0x6f | Dupliquer les trois premiers éléments de la pile |
| OP\_2OVER | 0x70 | Copiez les troisième et quatrième éléments de la pile vers le haut |
| OP\_2ROT | 0x71 | Déplacez les cinquième et sixième éléments de la pile vers le haut |
| OP\_2SWAP | 0x72 | Échangez les deux meilleures paires d'objets dans la pile |
| OP\_IFDUP | 0x73 | Dupliquer l'élément du haut de la pile s'il n'est pas 0 |
| OP\_DEPTH | 0x74 | Comptez les éléments sur la pile et poussez le décompte résultant |
| OP\_DROP | 0x75 | Déposez l'élément du haut de la pile |
| OP\_DUP | 0x76 | Dupliquer l'élément supérieur de la pile |
| OP\_NIP | 0x77 | Déposez le deuxième élément de la pile |
| OP\_OVER | 0x78 | Copiez le deuxième élément de la pile et poussez-le vers le haut |
| OP\_PICK | 0x79 | Extraire la valeur N du haut, puis copier le Nième élément en haut de la pile |
| OP\_ROLL | 0x7a | Extraire la valeur N du haut, puis déplacer le Nième élément vers le haut de la pile |
| OP\_ROT | 0x7b | Faire pivoter les trois premiers éléments de la pile |
| OP\_SWAP | 0x7c | Échangez les deux premiers éléments de la pile |
| OP\_TUCK | 0x7d | Copiez l'élément supérieur et insérez-le entre l'élément supérieur et le deuxième élément |

<<tx\_script\_ops\_table\_splice> > affiche les opérateurs de chaîne.

Opérations d'épissure de chaînes

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| *OP\_CAT* | 0x7e | Désactivé (concatène les deux premiers éléments) |
| *OP\_SUBSTR* | 0x7f | Désactivé (renvoie la sous-chaîne) |
| *OP\_LEFT* | 0x80 | Désactivé (renvoie la sous-chaîne gauche) |
| *OP\_RIGHT* | 0x81 | Désactivé (renvoie la sous-chaîne droite) |
| OP\_SIZE | 0x82 | Calculer la longueur de la chaîne de l'élément supérieur et pousser le résultat |

<<tx\_script\_ops\_table\_binmath> > affiche les opérateurs logiques arithmétiques et booléens binaires.

Arithmétique binaire et conditionnelles

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| *OP\_INVERT* | 0x83 | Désactivé (Inverser les bits de l'élément supérieur) |
| *OP\_AND* | 0x84 | Désactivé (booléen ET des deux premiers éléments) |
| *OP\_OU* | 0x85 | Désactivé (OU booléen des deux éléments supérieurs) |
| *OP\_XOR* | 0x86 | Désactivé (XOR booléen des deux principaux éléments) |
| OP\_EQUAL | 0x87 | Appuyez sur TRUE (1) si les deux premiers éléments sont exactement égaux, appuyez sur FALSE (0) sinon |
| OP\_EQUALVERIFY | 0x88 | Identique à OP\_EQUAL, mais exécutez OP\_VERIFY après pour arrêter si ce n'est pas TRUE |
| OP\_RESERVED1 | 0x89 | Halt - Transaction non valide à moins qu'elle ne se trouve dans une clause OP\_IF non exécutée |
| OP\_RESERVED2 | 0x8a | Halt - Transaction non valide à moins qu'elle ne se trouve dans une clause OP\_IF non exécutée |

<<tx\_script\_ops\_table\_numbers> > affiche les opérateurs numériques (arithmétiques).

Opérateurs numériques

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_1AJOUT | 0x8b | Ajouter 1 à l'élément supérieur |
| OP\_1SUB | 0x8c | Soustraire 1 de l'élément supérieur |
| *OP\_2MUL* | 0x8d | Désactivé (multiplier l'élément supérieur par 2) |
| *OP\_2DIV* | 0x8e | Désactivé (diviser l'élément supérieur par 2) |
| OP\_NEGATE | 0x8f | Retournez le signe de l'élément supérieur |
| OP\_ABS | 0x90 | Changer le signe de l'élément supérieur en positif |
| OP\_NOT | 0x91 | Si l'élément supérieur est 0 ou 1 booléen retournez-le, sinon retournez 0 |
| OP\_0NOTEQUAL | 0x92 | Si l'élément supérieur est 0, renvoie 0, sinon renvoie 1 |
| OP\_ADD | 0x93 | Pop top deux éléments, ajoutez-les et poussez le résultat |
| OP\_SUB | 0x94 | Faites apparaître les deux premiers éléments, soustrayez le premier du second, poussez le résultat |
| *OP\_MUL* | 0x95 | Désactivé (multiplier les deux premiers éléments) |
| *OP\_DIV* | 0x96 | Désactivé (diviser le deuxième élément par le premier élément) |
| *OP\_MOD* | 0x97 | Désactivé (le reste divise le deuxième élément par le premier élément) |
| *OP\_LSHIFT* | 0x98 | Désactivé (décaler le deuxième élément vers la gauche du nombre de bits du premier élément) |
| *OP\_RSHIFT* | 0x99 | Désactivé (décaler le deuxième élément vers la droite du nombre de bits du premier élément) |
| OP\_BOLAND | 0x9a | ET booléen des deux premiers éléments |
| OP\_BOOLOR | 0x9b | OU booléen des deux premiers éléments |
| OP\_NUMEQUAL | 0x9c | Renvoie TRUE si les deux premiers éléments sont des nombres égaux |
| OP\_NUMEQUALVERIFY | 0x9d | Identique à NUMEQUAL, puis OP\_VERIFY pour arrêter si ce n'est pas TRUE |
| OP\_NUMNOTEQUAL | 0x9e | Renvoie VRAI si les deux premiers éléments ne sont pas des nombres égaux |
| OP\_LESSTHAN | 0x9f | Renvoie VRAI si le deuxième élément est inférieur à l'élément supérieur |
| OP\_GREATERTHAN | 0xa0 | Renvoie VRAI si le deuxième élément est supérieur à l'élément supérieur |
| OP\_LESSTHANOREQUAL | 0xa1 | Renvoie VRAI si le deuxième élément est inférieur ou égal à l'élément supérieur |
| OP\_GREATERTHANOREQUAL | 0xa2 | Renvoie VRAI si le deuxième élément est supérieur ou égal à l'élément supérieur |
| OP\_MIN | 0xa3 | Renvoie le plus petit des deux premiers éléments |
| OP\_MAX | 0xa4 | Renvoie le plus grand des deux premiers éléments |
| OP\_WITHIN | 0xa5 | Renvoie VRAI si le troisième élément est compris entre le deuxième élément (ou égal) et le premier élément |

<<tx\_script\_ops\_table\_crypto> > affiche les opérateurs de fonction cryptographique.

Opérations cryptographiques et de hachage

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_RIPEMD160 | 0xa6 | Renvoyer le hachage RIPEMD160 de l'élément supérieur |
| OP\_SHA1 | 0xa7 | Renvoie le hachage SHA1 de l'élément supérieur |
| OP\_SHA256 | 0xa8 | Renvoie le hachage SHA256 de l'élément supérieur |
| OP\_HASH160 | 0xa9 | Renvoie le hachage RIPEMD160(SHA256(x)) de l'élément supérieur |
| OP\_HASH256 | 0xaa | Renvoie le hachage SHA256(SHA256(x)) de l'élément supérieur |
| OP\_CODESEPARATOR | 0xab | Marquer le début des données vérifiées par signature |
| OP\_CHECKSIG | 0xac | Popez une clé publique et une signature et validez la signature pour les données hachées de la transaction, renvoyez TRUE si cela correspond |
| OP\_CHECKSIGNVERIFY | 0xad | Identique à CHECKSIG, puis OP\_VERIFY pour arrêter si ce n'est pas TRUE |
| OP\_CHECKMULTISIG | 0xae | Exécutez CHECKSIG pour chaque paire de signature et de clé publique fournie. Tous doivent correspondre. Un bogue dans la mise en œuvre affiche une valeur supplémentaire, préfixée par OP\_0 comme solution de contournement |
| OP\_CHECKMULTISIGVERIFY | 0xaf | Identique à CHECKMULTISIG, puis OP\_VERIFY pour arrêter si ce n'est pas TRUE |

<<tx\_script\_ops\_table\_nop> > affiche les symboles non opérateur.

Nonopérateurs

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_NOP1 à OP\_NOP10 | 0xb0 à 0xb9 | Ne fait rien, ignoré |

<<tx\_script\_ops\_table\_internal> > affiche les codes opérateur réservés à l'analyseur de script interne.

Codes OP réservés à un usage interne par l'analyseur

|  |  |  |
| --- | --- | --- |
| Symbole | Valeur (hex) | La description |
| OP\_SMALLDATA | 0xf9 | Représente un petit champ de données |
| OP\_SMALLINTEGER | 0xfa | Représente un petit champ de données entier |
| OP\_PUBKEYS | 0xfb | Représente les champs de clé publique |
| OP\_PUBKEYHASH | 0xfd | Représente un champ de hachage de clé publique |
| OP\_PUBKEY | 0xfe | Représente un champ de clé publique |
| OP\_INVALIDOPCODE | 0xff | Représente tout code OP non attribué actuellement |