The Major System for Remembering Seed Word Numbers

November 17, 2026 · Document #431 · bip39-phrase.com

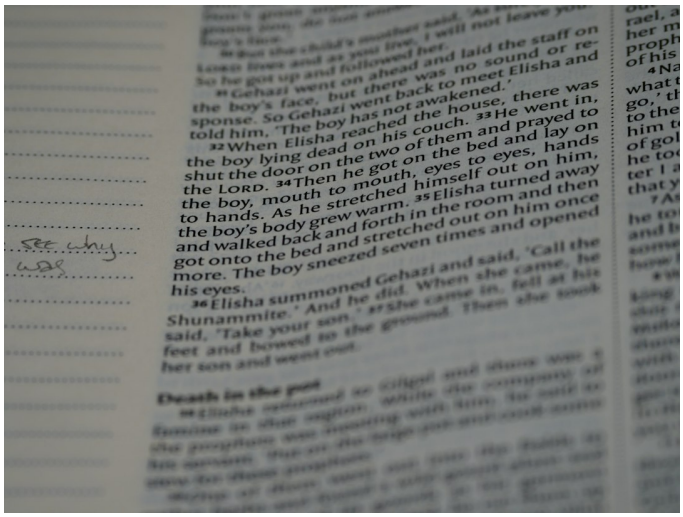


Photo by Brett Jordan on Unsplash

Learn how the Major System turns seed word numbers into images, and why numbering, padding and word order must stay consistent.

Contents

Fix the numbering convention first	1
Turn digits into consonant sounds	2
Encode one- to four-digit numbers	2
Preserve order and test recall	3
FAQ	3
Sources	3
Sample sequences of BIP39 words	4
Reading a BIP39 phrase	4
See also	5

BIP39 assigns each word in a 2,048-entry wordlist an 11-bit index from `0` to `2047`. Remembering those values requires preserving every digit and the order of the words. The Major System offers a way to turn decimal numbers into pronounceable words and mental images, but first you must fix the numbering convention. ([github.com](https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki))

Fix the numbering convention first

A word’s list number and its position in your recovery phrase are separate facts. Phrase position tells you where a word belongs; its list number identifies which word it is.

BIP39 uses indices `0–2047`. The same list displayed with row numbers starting at one instead uses `1–2048`, adding one to every index. In the official English list, `abandon` is the first entry: index `0`, or row `1`. Mixing these conventions changes which word a remembered number identifies. ([github.com](https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt))

Before building images, record the language, exact BIP39 list, numbering base and padding rule. For example, choose zero-based indices written as ordinary decimal integers, without leading zeros. Keep that convention with your recovery instructions.

If you prefer four-digit groups, define padding explicitly: `7` becomes `0007`. Those added zeros must either be encoded or restored by a fixed rule. Never alternate silently between padded and unpadded numbers.

Turn digits into consonant sounds

The classic Major System assigns digits to consonant sounds. Add vowels to form words you can picture. A common English mapping is: ([artofmemory.com](https://artofmemory.com/blog/major-system/))

- `0`: s, z, soft c.
- `1`: t, d, th.
- `2`: n.
- `3`: m.
- `4`: r.
- `5`: l.
- `6`: sh, ch, j, soft g, zh.
- `7`: k, hard c, hard g.
- `8`: f, v.
- `9`: p, b.

Vowels and the sounds w, h and y act as fillers. Work from pronunciation: `phone` contains an f sound; `cat` begins with k. Doubled letters count once when they represent one sound. Some versions differ, so choose one mapping and keep it fixed. ([artofmemory.com](https://artofmemory.com/blog/major-system/))

For the method here, count every mapped consonant sound in your chosen image word. Avoid a loose “first two consonants” rule: an unexpected extra sound could become an extra digit during recovery. Choose words whose pronunciation is unambiguous in your accent.

Encode one- to four-digit numbers

Use these standalone exercises to build images; they do not form a recovery phrase:

- One digit: `2` becomes “hen.” Ignore h and the vowel; n gives `2`.
- Two digits: `21` becomes “net,” from n–t.
- Three digits: `710` becomes “cats,” from k–t–s.
- Four digits: `1210` becomes “dentist,” from d–n–t–s–t? Reject it: that actually gives `12101`.

That rejected example shows why decoding matters. For `1210`, split the number into `12 | 10`: “tin” gives t–n, and “toes” gives t–z. Picture a tin crushing oversized toes, with the tin encountered first.

A useful working rule is to encode one- and two-digit numbers with one image, three-digit numbers as one digit followed by two, and four-digit numbers as two pairs. Define separate images for single digits and pairs with leading zeros: `2` and `02` must remain distinguishable. Preserve the order of images within each number.

For each number, decode the finished image back into digits before using it. If an image needs an explanation you repeatedly forget, replace it during practice.

Preserve order and test recall

Try this workflow with public practice material before applying it to a wallet backup:

1. Select the official wordlist and write down your numbering convention.
2. Convert each practice word to its list number, checking the lookup twice.
3. Encode each number using your fixed sound and grouping rules.
4. Assign one complete word number to each location on a familiar mental route.
5. Reconstruct the digits from each location, then look up the corresponding word.
6. Compare every recovered word and its position against the practice original.

Keep all images for one word number together. Give the next word a separate location, including when the same word repeats. Practice again after a day and a week, checking against the original each time.

Treat memorization as an additional recovery aid. Maintain a durable offline backup. A complete sequence of word numbers is another representation of the recovery phrase; the sound code provides no secret key or cryptographic protection. BIP39’s checksum detects most accidental errors, but not all, and a valid checksum does not establish that you recovered your intended wallet. ([github.com](https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki))

FAQ

Must I memorize all 2,048 word numbers?

No. Build images for the numbers you need and retain access to the exact wordlist for decoding.

Can I enter the image words into my wallet?

Decode them into numbers, then recover the original BIP39 words in their original order. The image words are your personal memory aids.

Should I choose seed words that make easier images?

Use the method to remember an already generated phrase. BIP39 specifies encoding computer-generated randomness and explicitly distinguishes that process from inventing your own sentence. ([github.com](https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki))

Sources

- BIP 39: Mnemonic code for generating deterministic keys (github.com)

- Official BIP39 English wordlist (github.com)

> Continue reading on bip39-phrase.com: How to Memorize a Seed Phrase: Color, Story and Palace Methods
> More on this topic: Seed Phrases

Sample sequences of BIP39 words

Randomly generated sequences of words from the BIP39 English wordlist. None of them passes the BIP39 checksum, so no wallet will accept them: they show the format only. Never store funds with a phrase you found in a document.

Example 1

1	2	3	4	5	6
fiber	elephant	mandate	unhappy	maid	toy
7	8	9	10	11	12
gap	hammer	they	actress	video	author

Example 2

1	2	3	4	5	6
garden	segment	volcano	rose	access	naive
7	8	9	10	11	12
breeze	slide	enlist	choice	cinnamon	october

Example 3

1	2	3	4	5	6
link	spider	wage	series	away	equip
7	8	9	10	11	12
order	decorate	crucial	divert	permit	plug

Example 4

1	2	3	4	5	6
jealous	advice	glance	spoon	sad	zoo
7	8	9	10	11	12
topple	observe	alcohol	quick	plug	curve

Example 5

1	2	3	4	5	6
hole	venture	nuclear	lecture	scatter	walnut
7	8	9	10	11	12
bracket	fetch	task	symbol	achieve	panic

Example 6

1	2	3	4	5	6
very	vessel	stove	vendor	retreat	patient
7	8	9	10	11	12
donkey	wolf	horse	suggest	economy	unique

Reading a BIP39 phrase

Every word stands for an 11-bit number: its position in the 2048-word list, counted from 0. Here is Example 1 decoded word by word. Each word links to its page on bip39-phrase.com, where the list is numbered from 1.

#	Word	Index (0-2047)	Word no.	11 bits
1	fiber	685	686	01010101101
2	elephant	574	575	01000111110

#	Word	Index (0-2047)	Word no.	11 bits
3	mandate	1080	1081	10000111000
4	unhappy	1898	1899	11101101010
5	maid	1072	1073	10000110000
6	toy	1843	1844	11100110011
7	gap	762	763	01011111010
8	hammer	836	837	01101000100
9	they	1796	1797	11100000100
10	actress	22	23	00000010110
11	video	1950	1951	11110011110
12	author	123	124	00001111011

The 12 words carry 132 bits: 128 bits of entropy followed by a 4-bit checksum. The checksum must equal the first 4 bits of the SHA-256 hash of the entropy.

`Entropy (hex): 55a8fa1c76a861ccd7d344e0805bcf07`

This sequence ends in the checksum bits 1011, but SHA-256 of its entropy starts with 1100, so wallets reject it as an invalid phrase. That is the point of the checksum: about 15 of every 16 random or mistyped sequences fail it.

See also

- Playing Cards as Seed Phrase Entropy · November 16, 2026
- Seed Access If You and Your Partner Die Together · November 15, 2026
- Why Song Lyrics and Famous Quotes Make Weak Passphrases · November 14, 2026
- Seed Phrase Planning for Incapacity and Recovery · November 13, 2026

Educational document. The example sequences are random and fail the BIP39 checksum: they are not wallets and must never hold funds. Security notes: bip39-phrase.com/security/