

We recently received this information from Professor Joseph Schnarchhund. The content is quite surprising, as it doesn't match earlier published fragments of Iemy script. His explanation for that discrepancy is included in the text, below. In fact, much of the following text came from the Professor, though we have edited it in an effort to make the style a bit more even.

It's unclear how Professor Schnarchhund obtained the information; his claim that his daughter Taurina received it from Isis during the Erfout Eetjney's layover in Russia is not terribly plausible, as other reports indicate that they were far too busy to send anybody anything while they were grounded there. None the less, he claims it is fully authentic, and it's hard to see where it came from if it's not, so we are willing to take him at his word, at least for the time being.

In this document, we'll be describing modern Iemy script. The background information was mostly obtained from the records in Erfout Eetjney's computer, though some was gleaned from Nim-nim, who knows as much about Iemy as any Earth person.

Early History of Iemyscript

Old Iemy was written in old Iemy script (as you might expect), which was a complex ideograph-based system with, we are led to believe, some thousands of difficult to draw characters. Published examples of Iemy characters – notably, in Felix's article in d'Ovni – were of Old Iemy ideographs.

Though relatively easy to read (*after* one learned the ideographs, of course), it was extremely cumbersome to write. As modern Iemy supplanted old Iemy, and the pronunciation of many words was gradually simplified, the characters used to write it seemed more and more out of step with the language. This situation was finally addressed by Iyřuhear {·**Λ**□**∨**×□} some number of centuries ago (dates are rather hazy in most of the history of the people of Iem; apparently none of them kept diaries). “Iyřuhear” is how he (or she) is remembered, but it is probably not a real name. It is a Iemy word meaning “write quickly and sloppily”; the English equivalent of the name Iyřuhear would be “Scribbles”.

Scribbles was supposedly one of the authors of the Book of Iem, and is sometimes referred to as Scribbles the Manyworded. Scribbles apparently found the situation with the Iemy ideographs extremely annoying, as did many; Scribbles, however, did something about it: She (or he) invented a whole new script. It's “new” Iemy script. It is far simpler than the old ideograph system, with just 96 basic characters. Furthermore, it's firmly oriented toward writing new Iemy: Representing a number of the sounds of old Iemy is quite awkward in the new script.

Parts of the Book of Iem, along with certain honorific terms, are written in old Iemy script. Almost everything else we are aware of uses the “new” (but still centuries-old) script.

One might think that someone inventing a script from scratch would either use some sort of simplified ideographs (which seem like an obvious notion), or would try to break down the language into its simplest parts, and assign one symbol to each phoneme. In point of fact, however, Scribbles did neither: The script is a syllabary, rather than an actual alphabet. What's more, it's a syllabary which doesn't include all phoneme sequences in Iemy, which resulted in widespread difficulty with the spelling of Iemyscript when it first came into wide use. This leads us to speculate that Scribbles actually obtained the basic characters from *some other language* and adapted them for use in Iemy.

Either that, or she (or he) was just lazy. (Cats, what can you do with them, they refuse to think like humans, and they all have ADHD...)

As we said, spelling was rather difficult when Iemyscript came into use. To allow the “missing” phoneme sequences to be written down, certain words, which couldn't be spelled “naturally”, were spelled using character *pairs*, where the first sound from the first character would be combined with the last sound from the second character. Typically a character pair was written just as one character followed by another, and one had to “just know” that the characters were being used that way when trying to read the word. There was, however, also a convention that the characters could be written *very close together* or even slightly *overlapping* when a character pair was intended.

That still wasn't sufficient to represent *all* sequences, however. A number of words simply had to be written using characters which included extra sounds. This was particularly common at the beginnings of certain words. In those cases, the initial phoneme represented by the first character just wasn't pronounced.

At any rate, that's what's believed at this time. Since there are no recordings of spoken Iemy from the period in question, it's quite possible that, at that time, the initial sounds of the words in question *were* pronounced, and pronunciation of the language has simply “drifted” since that time. But that's just speculation; what's known for sure is that examples of writing from that period can be very difficult to decipher, and are typically peppered with highly irregular spelling and many smudgy botched attempts at writing “overlapping” characters.

The Join Characters

Be that as it may, that rather awkward situation persisted for some decades after Scribbles invented the script. Things finally changed when a cat, whose name is not known (to us, at least), decided that the vague and confusing way of writing “joined” symbols was unacceptable. And that unknown person invented the *join sign*. When it's written between two characters, the join sign indicates that the two characters should be joined: The last sound of the symbol to the left should be removed, and the first sound of the symbol on the right should be removed, thus replacing the “head” of the second symbol with the “head” of the first.

This solved three problems at once. First, joined symbols were now easy to read (and write). Second, cases in which the symbol at the start of a word had an “extraneous” unpronounced phoneme could now be handled: A join sign placed *before* the first character of the word would join it to *nothing* – in short, it would just *remove* the first sound of the first symbol. Third, cases where the *last* character of a word had an unpronounced phoneme could also be handled (sometimes at a cost of spelling the word differently): A join sign placed *after* the word would join the last character to nothing, thus simply removing its final phoneme.

It was probably inevitable that once the “join” character was invented, someone would improve on it so that *all* words in Iemy could be properly represented. However, that didn't happen immediately; the new “spelling device”, alone, was in general use for quite a while, as we can tell from samples of writing from the years immediately after its invention. This was despite the fact that there were still some sound sequences which occurred in the middle of words where there was no appropriate symbol sequence. Eventually, two additional improvements came in rapid succession. The first was the

reverse join symbol, which removed the *head* of the character to the left, and the *tail* of the character to the right, and joined them, tail-to-head. That added a number of new combinations to the sound sequences which could be represented by the script. And finally, a device so simple it's surprising it wasn't discovered as soon as the “join” character came into use: A special “empty” character was introduced, which, when joined to another character, contributed nothing. Thus, joining something to the empty character was just like joining to nothing at the beginning or end of a word: It removed one sound. With that addition, it became possible to represent any sound sequence which occurred in Iemy. In fact, the “empty” character wasn't actually new. It was the character “𐌆”, or “en”, which is also the word for “none-of”. It had rarely been used in combinations, and after this usage was introduced, of course, it was never used in combinations, save when the join was being used to delete a sound from the character to which it was joined.

Character Formation

The characters of Iemyscript are very angular, with few curved lines, and almost remind us of cuneiform. A typical character consists of between one and four individual strokes. The strokes are not uniform, straight lines, but are typically slightly tapered. The thick end of each stroke is toward the top in vertical strokes, and toward the right in horizontal strokes. When “hand” writing the script, the strokes are intended to be drawn from top to bottom, and from right to left; thus, the thick end of each stroke is drawn first.

It's not known what Scribbles used for a medium when she (or he) was developing the script, but the strokes look an awful lot like the patterns which are formed by scratching a soft material, using a cat claw for a stylus. You dig one of your claws in at the wide start of the stroke, and then pull it out to form the tapered end of the stroke. (If you're a human, and don't have claws, you might achieve the same effect using a brush on paper. In point of fact, though, with a brush you would probably find it easier to form the wide part of the stroke *last*.)

The Digit Symbols

The number symbols also presented us with a surprise. There's no symbol for zero, which we already knew. However, much more surprising is that there are *eighteen* digit symbols. Since the numbering system uses base 12, this seems a bit excessive. We will have more to say about the numbering system(s) later on, but now we'd like to say a little more regarding the digit symbols themselves.

The digits are believed to be much older than Iemyscript in general. This is interesting since, though they are typically drawn a bit more heavily than the “alphabetic” characters, their style is inarguably similar to the characters of Iemyscript. Old Iemy script, on the other hand, doesn't look anything like the digit symbols. In consequence, we might speculate that the digit symbols may have had the same original source as the rest of new Iemyscript; alternatively, Scribbles may have taken the digit symbols as her (or his) inspiration when creating the characters of Iemyscript.

The Structure of the Syllabary

The syllabary contains 96 characters, as mentioned previously. That's eight twelves, which may have some significance to the cats; multiples of 12 seem to be very important to them.

The script may be conveniently broken down into several sections. There are 26 vowel symbols, 26

consonant symbols, 36 “mixed” symbols, and 8 “special” symbols. The “specials” are actually symbols which stand for entire words. Presumably these were words which Scribbles felt were exceptionally useful. (On the other hand, the “special” symbols could just have been added to the syllabary to bring the total to a multiple of 12. It's hard to say.)

In Felix's Romanization, the vowel, consonant, and mixed symbols are nearly all digraphs, with some trigraphs and longer strings used as well. The letters in the Romanization are supposed to be given something very close to the pronunciation of the equivalent IPA symbols (save for the 'ř', which is a trilled or tapped 'r'). Thus, the vowel symbols all represent diphthongs (or longer sound sequences). Similarly, the characters Romanized as consonant and mixed strings also represent sequences of multiple sounds.

Unfortunately, as we already noted, Iemy includes a number of words which can't be represented using the basic characters, due not to the presence of additional sounds, but rather to the presence of sound sequences which are not represented in the syllabary. We've already discussed the *join* and *reverse-join* characters which are used to enlarge the set of representable sequences.

Finally, the *suffix* symbols are rarely used, and are a holdover from old Iemy. Old Iemy had “part of speech” endings, which can still be used in new Iemy if the writer feels so inclined. In the script, however, they're not actually suffixes. Instead, to apply a part of speech suffix to a word, you place the “suffix symbol” directly *under* the last character of the word. The additional 'y' character which Felix specified in his Romanization, which was used to split the suffix off from the word in cases where the word ended with the starting character for the suffix, is not present in Iemy-script (but it correctly represents how the cats pronounce such a word, or so we're told).

And now, we present ***Iemyscript***. Examples, further discussion, and information regarding Iemy numbers will be provided below, after the table of characters.

Each entry in the table is shown with Felix's Romanization, followed by the corresponding Iemy-script character; the entries are sorted by the Romanization value. (The “native” lexicographic order for Iemy symbols, if any, is not known at this time.)

Vowels		Consonants		Mixed Strings		Special Strings	
aa	≡	řf	//	ař	∕	ařsit	↳
ai	≠	řh	≠	ařf	≡	arv	▷
aio	▷	řl	≠	ařt	∕	fey	▷
ao	≠	řn	△	aj	↳	let	↳
aou	↳	řs	△	ar	□	no	↳
au	△	řt	▽	arf	△	yřfachiř	↳
ea	≠	řv	↳	arn	△	yn	△
ee	≠	řvh	∅	art	>	yřv	↳
eeu	◁	řz	△	at	◁		
ei	≡	nh	┌	eet	↳	Connectives	
eo	≠	nz	┌	ef	┌	Join	·
eou	◁	rd	┌	eh	↳	Reverse	:
eu	↳	rf	↳	em	↳	Join	
ey	≠	rh	↳	en	∅		
ia	≠	rn	≠	et	┌	Digits	
ii	△	rt	∕	iř	∕	0	
io	≠	rtf	↳	iřt	↳	1	↳
ou	≠	rv	↳	ib	π	2	↳
oua	┌	tf	↳	ig	π	3	↳
ouu	┌	tl	≡	ik	◁	4	┌
ua	↳	tv	◇	ir	≡	5	┌
ue	↳	tvr	↳	irh	▽	6	▽
ui	△	vh	┌	it	┌	7	↳
uo	△	vl	≠	oř	┌	8	┌
uu	↳	vr	◇	ořt	▷	9	┌
uy	△	zf	×	op	◁	10	┌
				ouar	┌	11	△
				out	∕	12	≡
				uf	┌	13	↳
Suffix Symbols				uh	↳	14	↳
o	„			un	π	15	↳
en	„			ut	↳	16	≡
ol	„			yř	□	17	↳
or	„			yřf	▷	18	□
ors	„			yřn	×		
				yt	π		

The Join Rules

The first thing to note about Iemyscript is that use of the join signs results in a plethora of ways to spell many words. If the script had evolved “naturally”, we would expect this to have resulted in many homonyms, where words with identical pronunciation meant different things, and were spelled different ways. However, that's not the case with Iemy. The first reason is that the basic script was created all in one “go” by Scribbles, who developed a single system for spelling all sound sequences, rather than working out the spelling of each word individually. The join symbols were introduced by one (or a few) cats, some decades later, along with rules for using them. Consequently, the spelling is entirely phonetic, and words which are pronounced alike are spelled alike.

The term “phonetic spelling” tends to pre-assume that a word which sounds a particular way *cannot* be spelled in more than one way, and we've just said that's not the case. So, if not, then what case is it?

What *is* the case is that there are rules which determine how to spell any word, based solely on how it sounds. Unfortunately the rules are rather complicated and somewhat arbitrary. And this brings us to the second reason that the spelling is regular: Most cats are not very patient pupils, and have little interest in learning subtle variations in spelling. In fact, most Iemy cats have a lot of trouble with the system as it stands, and we're told that “invented spelling” is very common. The dictionary may list just one spelling for each word, but “out on the street” you can find nearly as many ways to spell things as there are cats who spell. So, while they grasp the connection between how things sound and how they're spelled, the idea that *certain* words which sound a particular way are spelled one way, while *other* words which sound exactly the same are none the less spelled *differently* would most likely never have caught on.

An example may help to clarify this. Consider the name “Scribbles”, or “Iyřuhear”, which we spelled ·A□vX□. We could also write it as ·^□vX□, or as †:ll̄x̄·□vX□, or †:ll̄x̄·□v<|·††|, or ... the list of alternatives is nearly endless.

In fact, even without join characters, phonetics alone may not be enough to determine the spelling of a word. As another example, the word *arfey* (ground or land) could be spelled either □↗ or △✱.

There are five rules used to determine the correct spelling when more than one possibility exists. They are to be applied in order, until there is just one candidate left:

1. A spelling with fewer characters (*not* counting join signs) is preferred over one with more characters. (Note that the “none” character, if present, *is* included in the count.)
2. A spelling with fewer join signs is preferred over one more join signs.
3. When a join is needed, a plain join is preferred over a reverse join.
4. If a join sign is used to obtain a particular head or tail, and there is more than one possible “donor” character, the choice is determined by consulting the head or tail donor list.

These two lists each contain all 96 Iemyscript characters. To choose a “head” donor character, you find the first character on the “head donor” list which provides the necessary sound; to find

a “tail” donor character, you find the first character on the “tail donor” list which fills the bill.

In this case, working with the Romanization actually gives us a significant advantage over a native Iemy speaker, because we can see, looking over the two donor lists, that it's the *suppressed sound* which determines the place on the list. And the sequence of possible suppressed sounds is far shorter than the full set of characters. It's:

'y', 'o', 'u', 'e', 'i', 'a', 'j', 'b', 'p', 'g', 'm', 'k', 'd', 's', 'z', 'v', 'l', 'h', 'n', 'f', 'r', 'ř', 't'

Thus, for example, suppose you need to form the sound “af”. There's no single character which provides it, but you can find characters which *start* with “a”, and characters which *end* with “f”. All you need to do is join two of them. How do you select the ones to use?

The ones starting with “a” are (in Romanized form) “aa”, “ai”, “ao”, “au”, “ar”, “aj”, “ař”, and “at” (where we've considered only *digraphs*, because we need just one phoneme in the head, and the join character only removes one phoneme). Looking at the collating sequence shown above, we see that there is no “ay”, but there is an “ao”. So, the 'head' character should be “ao”, or '𐌰'.

Characters which *end* with an “f” sound include “rf”, “tf”, “zf”, “ef”, and “uf”. The earliest one of those in our collating sequence is “uf”, with a 'u' in the suppressed position, so the symbol we want for the tail is “uf”, or '𐌱'. And so, to form “af”, we'd use the sequence “𐌰𐌱”.

In practice, neither the full 96 character list nor the 23 Roman character collating sequence is needed for choosing characters to join. There will (nearly) always be a character available which has a vowel phoneme which can be trimmed off, and the vowels all come at the start of the Romanized collating sequence. So, all one really needs to remember to get it right most of the time is “youeia”.

5. Characters representing more complex sound sequences should be placed *later* in the word.

Using the five rules on our two examples, we can see that the correct spelling of “Iyřuhear” must be “·𐌶𐌰𐌶𐌰𐌶𐌰”. It is the shortest spelling of the ones we exhibited, it has just one join sign, the join is *forward*, and we can't find anything in Iemyscript which can eliminate that join, because no symbol's phoneme string starts with “iy”. The character used in the join, “ui”, is preferred for joining over any other digraph save “yi” or “oi”, and neither of those exists. Finally, the last four characters in the word are all digraphs, so there is no ambiguity in the character choice due to the presence of longer sound sequences.

When we look at *arfey*, we see that “𐌱𐌶” puts the 3-phoneme sequence “𐌶” later in the word than the 3-phoneme sequence “𐌶” which occurs in “𐌶𐌶”. So, the correct spelling must be “𐌱𐌶”.

As we said to start with, this process is rather complex and cumbersome, and most cats just learn the spelling of the words they know by rote, much as we do in English. (After all, how many of us consider the etymology of the word *bright* when we're deciding how to spell it, and conclude that there should be a silent 'gh' group inserted in it to indicate the presence of a suppressed “ch” sound? We just remember that the “eye” sound in “bright” is spelled “the funny way” rather than the usual way.)

Actually that's not entirely accurate. What we've been told is that most cats *who can spell* learn the spelling of most words by rote rather than applying the rules. But most cats don't do either, and just spell really, really badly.

The Number Systems

We've explained elsewhere that the cats use a base 12 numbering system, and Felix's Romanization of it uses the 0 character along with the Greek letters α through κ to represent the digits 0 through 11. (The Romanization would have been a *lot* easier to read if Felix had chosen the more traditional rendering of base 12 as the digits 0 through 9 along with the Latin letters 'a' and 'b', but we're not going to try to explain that here. It's hard enough to explain what the cats did; explaining Felix is entirely beyond our abilities.)

However, the cats use a rather different system for representing their numbers. As we already knew, they have no symbol for the digit '0'; where a zero occurs in a place, they just leave it blank. More surprising, perhaps, is that they have *eighteen* digit symbols (which doesn't count 0). (This is particularly surprising in light of the fact that Isis stated in an earlier communication that they used *eleven* digit symbols. But she was still learning the language at the time, and may not have been aware of the extra six digits.) Since base 12 only requires 12 symbols (including 0), the result is that there is more than one way to write most numbers, including a couple of completely unreadable ones.

The most common way of representing numbers in printed media, including computer screens, is to use the 11 digits from \uparrow to \wedge , along with a blank for zero. As long as the font used is *unit width* the result is relatively readable. This system is also often used when writing values by hand, where the order of magnitude is already known. In a lot of cases, only the mantissa matters, so the fact that there's no way to tell how many zeros are missing is not important. (It also appears that it's sometimes used in cases where it *does* matter how many zeros are missing, but the cat doing the writing is just lazy. That may have been the case with the room numbers on the Erfout Eetjney, which Isis found so confusing when she was first finding her way around the ship.)

The first variation on this is used when it's important to know the number of “missing” zeros. The zero count can be written small, and placed just after the last nonzero digit. (It's a subscript, but often not placed all the way down in what we would call “subscript position”.) So, for instance, the number 1000_{12} can be written “ \uparrow ”, but if the exact number of elided zeros is important, it can also be written “ \uparrow_{12} ”. When written this way, the subscript takes the place of one zero; the rest of the “missing” zeros may or may not be written (or, rather, left blank) depending on the taste of the writer.

In either of these two schemes, the “extra” digits, from $\bar{_}$ to \square , are often used to write the *first* digit of the number. In particular, numbers up to 16_{12} (in which the first digit is the only digit) are commonly written using the extra digits. Thus, for instance, the value 130013_{12} would commonly be written “ $\bar{_}_{12} \uparrow \wedge$ ”.

This is not the only approach which is taken to reduce the confusion over the missing zeros, however. When hand writing numbers, it's common to use a *zero avoidance* scheme. Where a zero digit would appear in a number, it's common to *borrow* from the next digit to the left, and replace the zero with a 12. Thus, 1000_{12} is often written “ $\wedge \wedge \bar{_}$ ”. This is typically combined with use of the digits from \mathcal{N} through \square in the first place of the number. So, we would often see our second example, 130013_{12} , written as “ $\bar{_} \wedge \bar{_} \uparrow \wedge$ ”. This is probably the most common way to hand-write numbers in Iemyscript.

In the three schemes we've described so far, the characters \mathcal{N} through \square are only used in the first digit

of a numeral. There is a fourth scheme which is used occasionally, and which is rather like the reverse of the preceding scheme. The “extra” characters are used anywhere, zero (blank) isn't used at all, and the characters for values 1 through 6 (that is, **!** through **∇**) are restricted to the first digit of the numeral. We might call this “compact base 12”, since the digit strings are as short as they can be made using digits with values from 0 to 18 and a base 12 place value system (though, in fact, they're still no more than one digit shorter than “simple” base twelve).

To convert from “ordinary” base 12 to this “compact” base 12, we can start at the right end of the number, and replace all the 0 through 6 digits by borrowing from the place to the left. (When we get to the left end, there's nothing to borrow from, so the highest digit may be in the range 1 through 6.) Thus, our earlier example 130013_{12} would be converted as:

$$130013 \rightarrow 13000d \rightarrow 12bbcd \rightarrow ebbcd \Rightarrow \nabla \wedge \wedge \sqsubset N$$

Another and perhaps more interesting example of “compact” base 12 is provided by 17776_{12} . Converting it, once again, in Romanized form, and working from right to left, we get the sequence:

$$17776 \rightarrow 1776i \rightarrow 176ii \rightarrow 16iii \rightarrow iiii \Rightarrow \square \square \square \square$$

This all may seem very cumbersome, but it's only the conversions that are a problem (or so we're told). Arithmetic in any of the four systems is straightforward, as long as all the values are in the same form and you're happy to get the result in that form as well. It's not nearly as troublesome as, for example, the mixing of Imperial and metric units which North American scientists must deal with.

What About Zero?

Finally, suppose you have zero of something. How do you say that? It's certainly not possible to do meaningful mathematics without ever referring to zero! Do the People of Iem always just write it out, as “**✂✂□□✂**”? That would be terribly cumbersome!

And, in fact, they don't. It's only the *digit* zero which is missing. When the *value* zero is needed, they typically use the symbol for the word “none”, which is written “**∅**”.