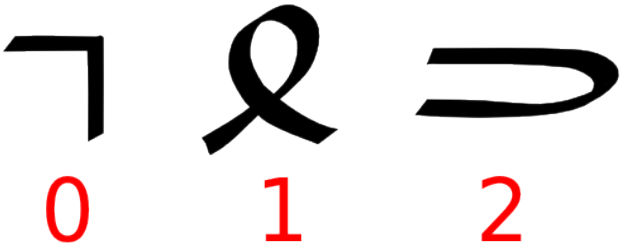
WireScript

a.k.a Triple Trit Trinary
a.k.a Ternary Dscript



WireScript is is a version of Dscript where each letter is represented by 3 trits(a trit has 3 values, like a bit has 2)

Each trit can be 0,1,or 2. Each value is represented by either a 90 degree corner, a 180 degree curve, or a 270 degree loop.

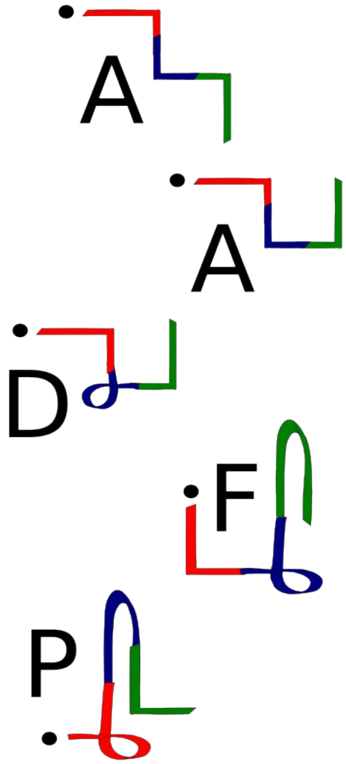
3 trits yields 27 possibilities. Full English Alphabet + Space.

0 degrees is meaningless as it can not be identified.
360 degrees is hard to distinguish from 270.
So neither is used.

Absolute Orientation is relative, so it not used either. Also, relative orientation (like left<->right) causes far to many unintended line crossings that might end up being confused as loops.
ie. 3 left turns is a loop with corners

Corners and loops provide a choice of direction. (left,right)
180 degree curves have no such choice

The "entry point" is denoted by a dot.



INDEX

First Trit **Second Trit** **Third Trit**
*colors for learning examples only
not needed for usage

¬

	¬	ℓ	⊃
¬	a	b	c
ℓ	d	e	f
⊃	g	h	i

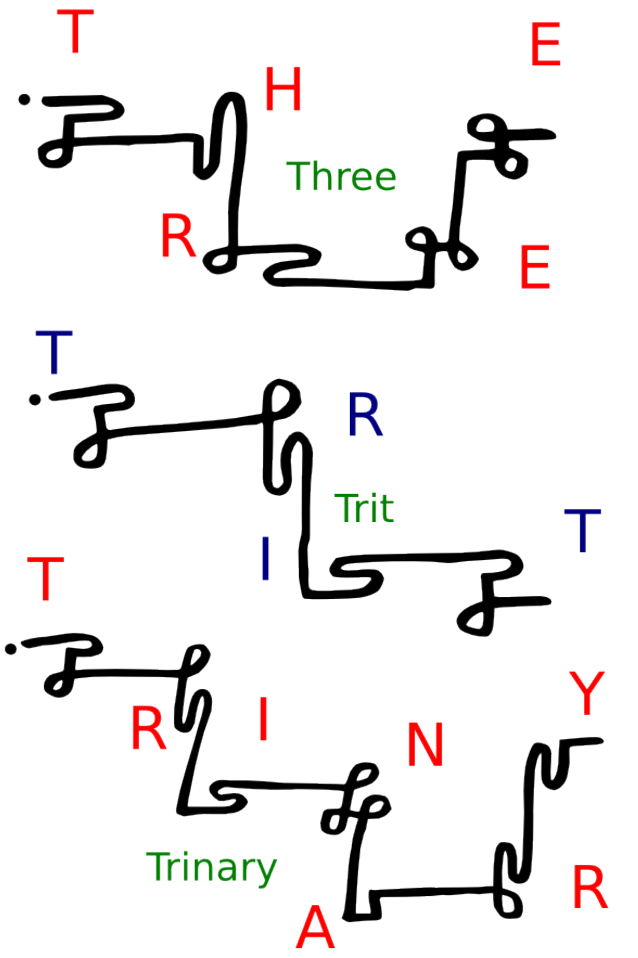
ℓ

	¬	ℓ	⊃
¬	j	k	l
ℓ	m	n	o
⊃	p	q	r

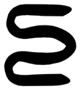
⊃

	¬	ℓ	⊃
¬	s	t	u
ℓ	v	w	x
⊃	y	z	Space

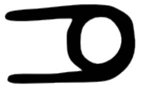
Here you can see a quick example using the words “three” “trit” and “trinary”.

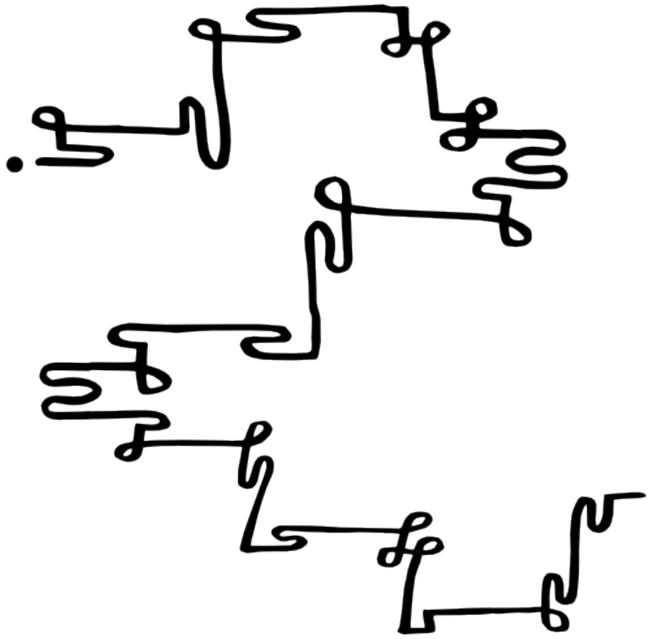


At first it is easier to draw “gaps”(straight lines) between letters to make reading easier.

The “space” character is three 180s 

An alternate form of the space is to merge all three 180 degree curves into a single shape.

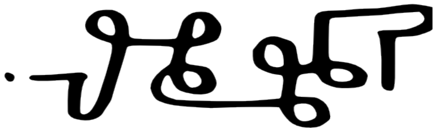




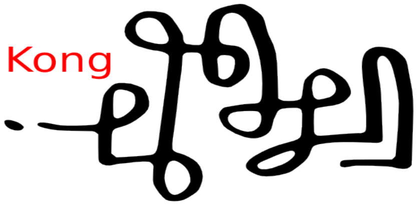
"three trit trinary"

With the space character it possible to write entire sentences as a single line without lifting the pen. I use 2 spaces to indicate a “.” allowing entire paragraphs. 3 spaces for “line break”, allowing multiple paragraphs.

Hong

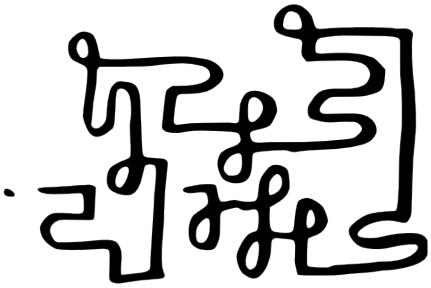


Kong



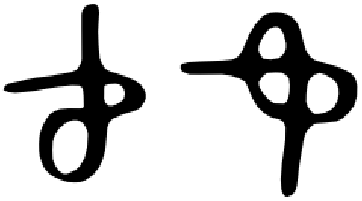
With a little practice the letters can be tightened up. It is a little harder to read, but still legible. Just be careful not to miss a trit when writing or skip a trit when reading.

Phrases can be tightened into a nice glyph as well.

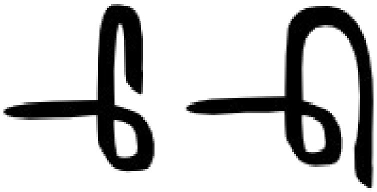


Carpe Diem

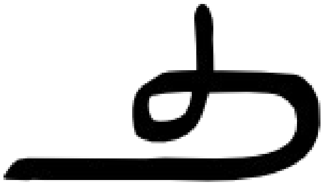
The trickiest combination is the “180 curve + 270 loop” as this will cause lines to cross. It might look a little confusing, but it is easily identified and very unique, when you see these read them a “180, then 270”(or 2,1).



The reverse of 270 + 180, does not present such a problem. It does often risk colliding with the loop. This is easily avoided by extending the curve wide enough to avoid the loop.



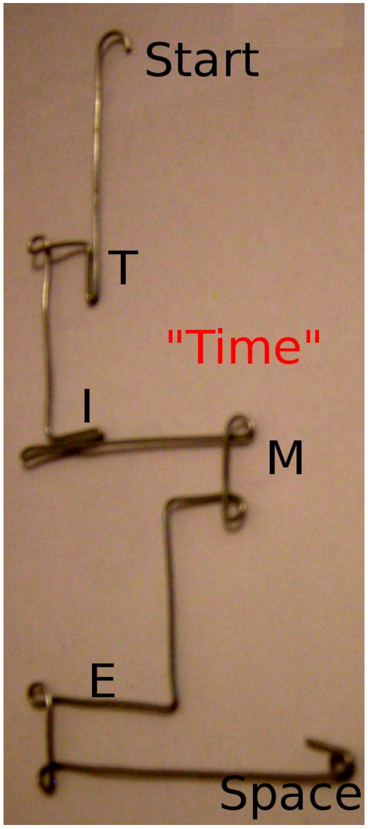
Technically the same principle could be applied to the “180 + 270”. But this requires more planning and it is easy to accidentally draw lines so close they touch, especially if the “180+270” is followed by another “180”. It is slightly tricker to implement, but no less valid.

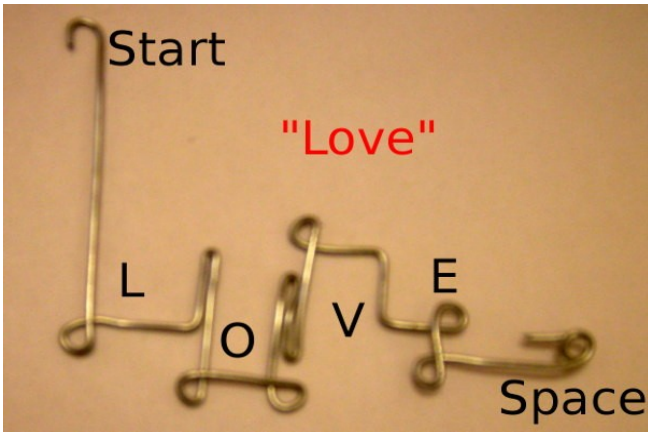


NOW...... The real fun begins!

Because WireScript requires no pen lifts, and uses very simple angles, it can be written.... you guessed it..
with wires!!!

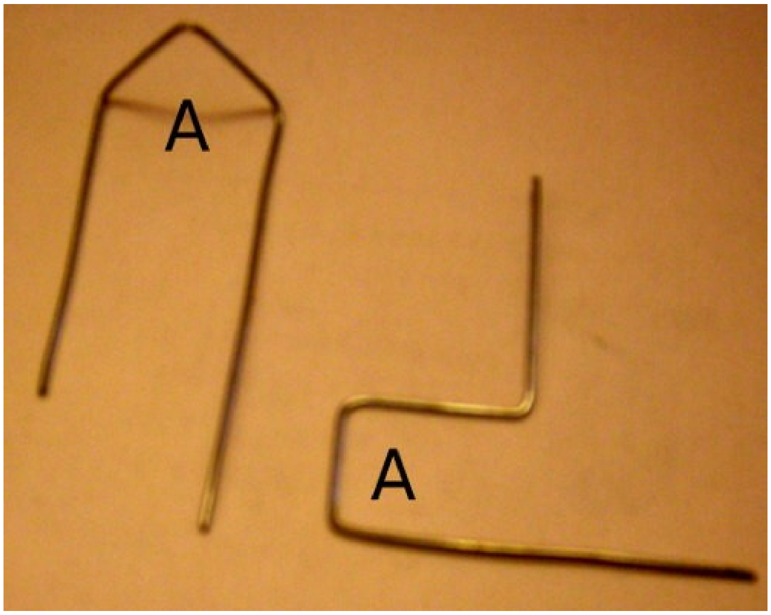
In this example of the word “time” the letters are spaced out for easy reading as before. Below is the word “love” tightened up together.





The 180+270 problem even disappears because crossing the lines is much less of a problem.

Now... Even more fun... 3 DIMESIONS!!!



This is where the “real fun” begins. Because WireScript ignores relative orientation, when we use wires to write in WireScript we now have 4 choices of direction instead of 2.(except for 180)

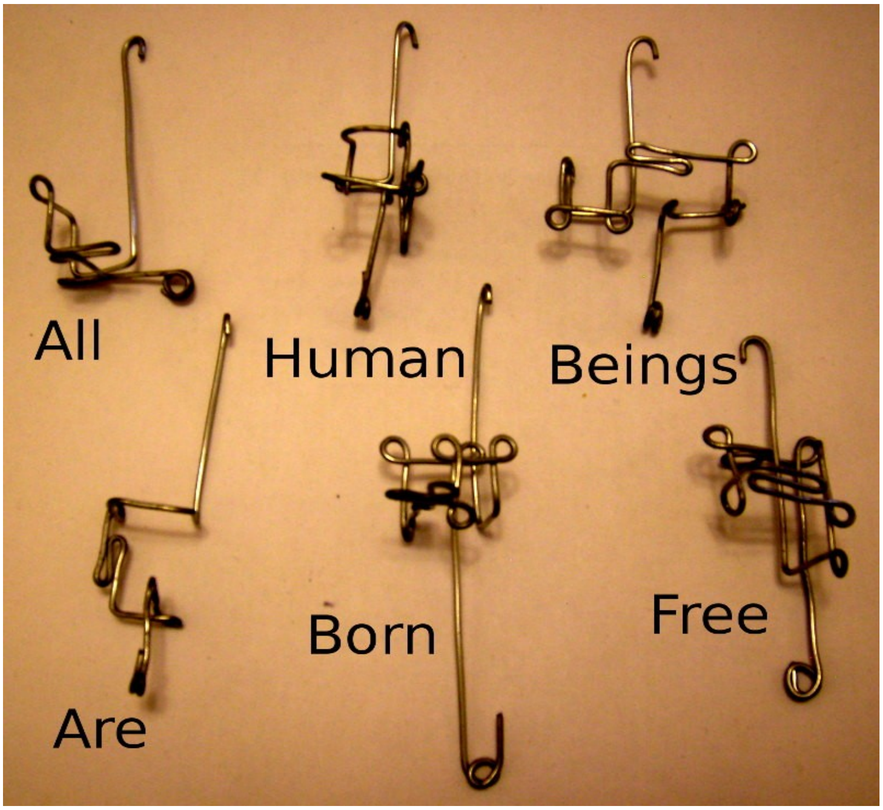
In 2D the 90 and 270 can go “left or right”.

In 3D a 90 or 270 can choose between 2 axis instead of 1. Creating three dimensional shapes.

Here we see the letter “A” in 2 of its possible forms. Both are composed of 3 separate 90 degree corners, but the one on the left only uses 1 axis, while the one on the right uses both axis.

Here are the words “all human beings are born free”. (The wire I used is a bit thick, and my pliers are a bit wide so some of my 90 degree corners are a bit round. I hope to improve this with practice and better tools/materials)

It is hard to depict a 2D picture of a 3D script. Check out the WireScript video to get a better idea.



[WireScript Video](http://youtu.be/vujACDUNn1k)

http://youtu.be/vujACDUNn1k

*folding 3D WireScript has given me a new appreciations of the complexity of protein folding ;)

You may have noticed that the “start” is an “open hook” and the “end”/”space” is a “540 loop”.

If you have not guessed why yet... its so they can hang and string together :)

On the right you have the exact same 6 words “All human beings are born free” strung together into a hanging chain.

Words need not be broken into individual modules, with the space character you can put as many words as you like into a single “ornament”.

Breaking them into individual “ornaments” just makes it easier, and allows you to re-use words later.

These ornaments have rather long “starts and tails”. Ideally they can be tightened up as well.

A string could start at the top and hang down, or could start at the base and form a standing structure. Just about anything could be made out of WireScript.

A touch of glue/solder or a small weld could hold them in place if you want to create more complex structures that do not “bend to the will of gravity”.

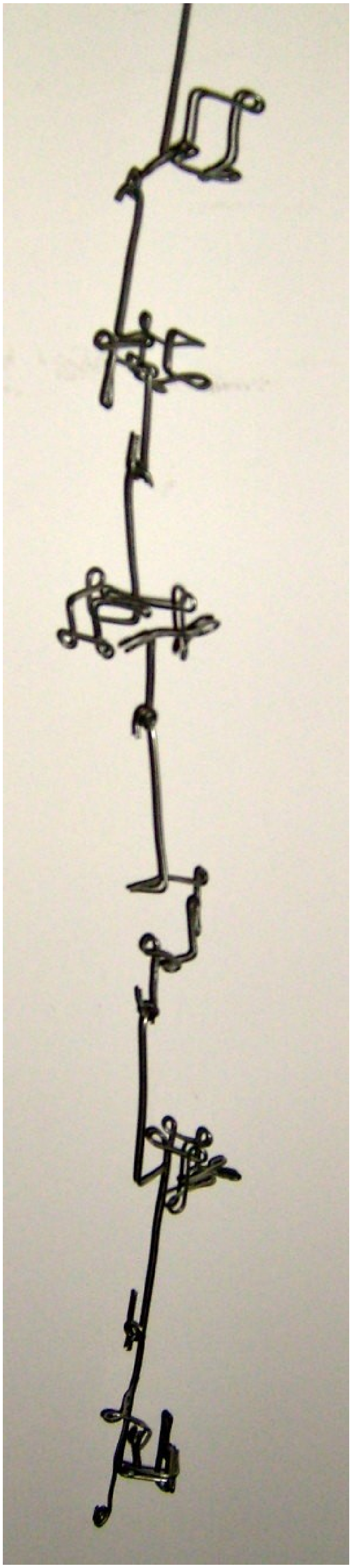
As the structure becomes bigger eventually gravity may outway the structural strength of the wire. To get around this just have the wire come back and provide “support” to other segments of the wire so that load is shared.

The best way to add structural integrity is by passing back through the middle of an earlier “270 loop”, but this is tricky.

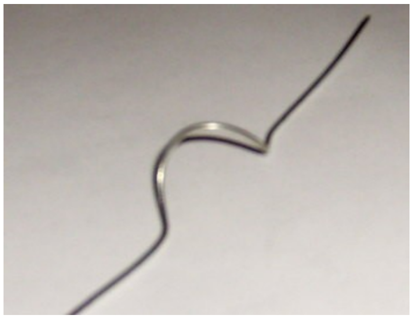
WARNING:

WEAR SAFETY GLASSES!!!!!

The wire bounces around as you work with it, I almost took out an eye before I realized the level of danger.



Interesting Points / Other Options



Helix: When working in 3D there is a new option available. The 3D helix. This would allow base 4 instead of base 3. It is harder to read, but I have contemplated using it as the space character or punctuation. If you need more than 26 letters than this is a good way to increase the size of your index from 27 to 64.

More Trits: If you have an alphabet larger than 26, you may want to use a "4 trit system", this increases the index to 81 characters.
*if you use both above methods you can reach an index size of 256 (4x4x4x4=256). This should be more than enough for any language.

Computers/Electronics: Trinary (or more correctly referred to as Ternary) has been use din computing. The Russians built ternary computers back in the "tech race" era. Normal bits are 1/0. Ternary can easily be described as 0/1/2 but more importantly, ternary can take advantage of electrical polarity and employ -1/0/1. This allows for some very cool logical operations, many of which have a great advantage over binary. But the increased complexity of design, and the fact that binary has become the standard, has lead to ternary being all but forgotten. (actually there are a few special applications where ternary is used, but not in microprocessors as far as I know). It is probably more likely that as computers develop we will "skip ternary" and "jump to quaternary"(base 4). Ternary is a real pain to convert to binary, whereas quaternary converts easily, so quaternary is much easier to integrate with current technologies and standards.

Resources:

WireScript Tutorial Video : http://youtu.be/vujACDUNn1k
Wirescript PDF download link : http://dscript.org/wirescript.pdf

Wirescript is a derivative of Dscript 2D writing system
Dscript Introduction PDF : http://dscript.org/dscript/pdf
Dscript Sites : http://dscript.org   http://dscript.ca