# cell

## Summary

**draft version: 0.2**
**rev date: 180421**

audio visual poetic language

## Summary

cell is a two-dimensional esoteric programming language made to create short audio visual poems.

cell programs are written by placing letters on a 10x10 grid with each cell of the grid containing one letter only. One can write a letter to a specific cell by specifying its (x,y) coordinates using predetermined unicode symbols in place of numbers. This way, a letter can be selected by its corresponding decimal code.

cell uses a base 10 numbering system, counting starts from 0. Decimal values can also be calculated from basic arithmetic operations using postfix notation.

order of operators doesn't matter inside of the instruction.
space characters are escaped.

audio can also be generated by writing dedicated symbols on the same grid, each cell holding a single oscillator pulse with a specific frequency and ADSR envelope parameters.

# OP

## Coordinates

| COORD | X | Y |
|---|---|---|
| **OP** | ⋯ | ⋮ |

## Numbers

| NUM | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **OP** | ● | ○ | ∟ | ▶ | ⊹ | ▣ | ∈ | ◆ | ⚯ | ✕ |

## Arithmetic

| CMD | + | - | * | / |
|---|---|---|---|---|
| **OP** | ⊷ | ⊸ | ꝏ | ≎ |

/: floor integer division

## Commands

| CMD | DEF | EOL | ESC | DOT | JMP |
|---|---|---|---|---|---|
| **OP** | : | ⁘ | ∸ | • | ↷ |

DOT operator is used to mark the end of a decimal sequence

Oscillators

| **OSC** | sine | square | sawtooth | triangle |
|---|---|---|---|---|
| **OP** | ∼ | ■ | ⥧ | ▲ |

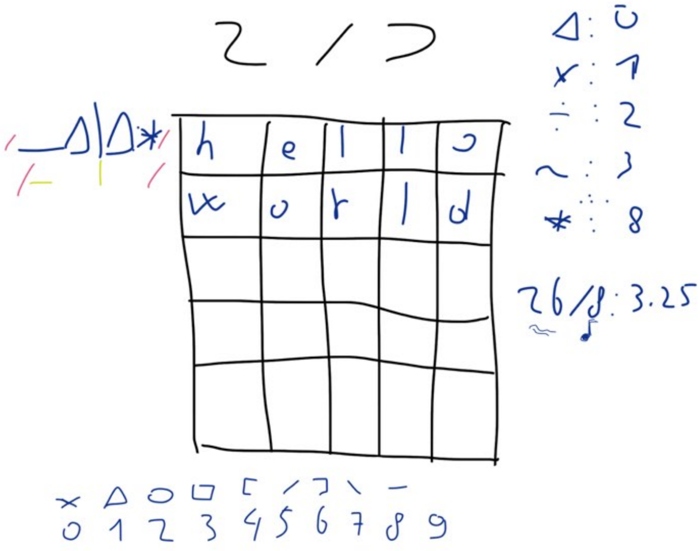
Sequencer

| **CMD** | START | END |
|---|---|---|
| **OP** | ∴ | ∵ |

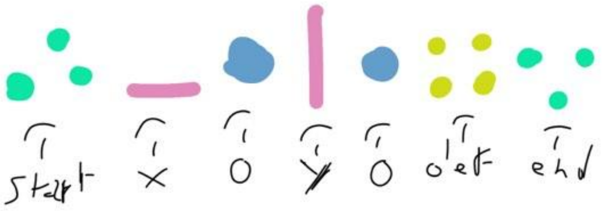
## Examples

### ver. 0.1

*letter are selected by alphabetical order instead of unicode



| | | | | |
|---|---|---|---|---|
| h | e | l | l | o |
| w | o | r | l | d |
| | | | | |
| | | | | |
| | | | | |



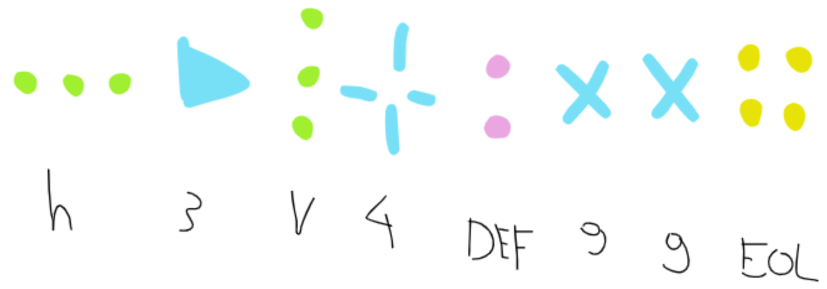
∴_●|●∷◆∵
∴_○|●∷▶∵
∴_⌓|●∷○⌓∵
∴_⊹|●∷○∾⊸∵
∴_▣|●∷○▶∵
∴_●|○∷⌓⊹∵
∴_○|○∷○▶∵
∴_⌓|○∷○◆∵
∴_⊹|○∷○⌓∵
∴_▣|○∷▣∵

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| h | e | l | l | o | | | | | | |
| w | o | r | l | d | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |

### ver. 0.2

⋯▶⁝⊹:`××`∷⋯⊹⁝⊹:○●○∷⋯▣⁝⊹:○●☍∷⋯∈⁝⊹:▣⊹•∟∾∷⋯◆⁝⊹:☍◆▣`×`∷

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | c | e | l | l | ∷ | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |



⋯▶⁝⊹:`××`∷
⋯⊹⁝⊹:○●○∷
⋯▣⁝⊹:○●☍∷
⋯∈⁝⊹:▣⊹•∟∾∷
⋯◆⁝⊹:☍◆▣`×`∷

```
h3v4DEF99EOL
h4v4DEF101EOL
h5v4DEF108EOL
```

```
h6v4DEF54DOT2*EOL
h7v4DEF8759EOL
```

## Ideas

- add color operator followed by 3 chars of color hex code

- possible to create variables of types int char ascii hex

- possible to create variables that acts as placeholder for sub instruction using a custom symbol for the whole subinstruction

- audio and text can be used as modulation sources both ways e.g. to change a group of letters style from gain or to transpose a note by “patching” a letter from another cell.