SPADE-The world's first carbon neutral, zero-emissions language.
NOW EVERYONE CAN CULTIVATE THEIR CODE
Language Specification

Data types

- i32 -> 32-bit signed integer
- u32 -> 32-bit unsigned integer
- f32 -> 32-bit floating point [IEEE] number
- b8 -> 8-bit unsigned integer [byte]
- b1 -> 1-bit unsigned integer [boolean]
- c32 -> “character” unicode code point
- c∞ -> string

Constants

- +∞=plus infinite
- -∞=minus infinite
- π=pi
- Є=e

Comments

```
⚠ this is a comment
```

Function

Function init/definition:

```
<function name> with (vars[])
        ⚠ this is a comment
        ⚠ this is a comment
plant
```

supply -> return

```
fibonacci with (i32 x, u32# y, c∞ z)
        supply x
plant
```

Function call:
reap <function name> with (vars[])

reap fibonacci with (i1, 2.0, "aedhaw")

Struct

here lies [struct name]

```
___________
|
|        sow <var name> of i32
|        sow <var name> of u32
|        sow <var name> of c32
|___________
```

[struct name]→lmao is 64.

If-else

```
a==1 fresh?
        do something
rotten a==2 fresh?
        do something else
rotten
        do something else finally
dispose
```

For-loops

harvest [from <loop_var> is <init_val>] (until <condition> | eternally) [every <value> <loop_var>]
sell

skip -> continue
kill -> break

harvest until x==0 every 1 x

```
        x is x-2
sell

harvest from apple is 0 until day = 25 every 5 apple
        cow is sad
sell
harvest from i is 0 until i =25 every 5 i
        cow is happy
sell
```

Variable initialization

```
sow <name> of <data-type>
sow apple of i32
```

Assignment

```
<name> is <literal>
apple is “red”
oranges#3 is 444.33
```

Arrays(0-indexed)

```
sow <number> <var name> of <data type>
sow 6 oranges of i32

oranges#2
```

Output

```
engrave <statement> [on <file handle>]
engrave "kiss me" on stdout
engrave “miss me clooooooooooooooooooose your eyes” on stderr
engrave "hello" on C:/dir/file.txt
```

Input

```
gather from stdin
gather from <file handle>
apple#45 is gather from stdin
```

File I/O

[variable] is unearth [filename] ← [modes]

bury [variable]

Operators

- + add
- - subtract
- × multiply
- ÷ divide
- / fractions
- 3^{12}
- √46 square root
- & bitwise and
- | bitwise or
- ^ bitwise xor
- ~ bitwise not
- =,≠ equals
- >,<, ≥,≤ comparisons
- true: ⭕
- false: ❌
- && logical and
- || logical or
- ! logical not