

Purpose: Clank exists to make debugging simpler and more streamlined. A simple change in a statement's terminator can yield a different output or result. This makes Clank easy to read, and allows for control flow, functions, and user-defined variables with a few basic data structures. Clank offers a robust, dynamic system for testing statements quickly.

Sample Programs

```

var x = "Hello, world!";
x?

Array a = [1, 2, 3];
a?

var num = 13;
num?

x\
a\
num\

func ouchieDoodle(x, y) {
    var hello = x + y?
    return hello?
}

ouchieDoodle(1,2)@

```

Sahba Ostovaravary

Clank is a language designed around using terminator characters to modify any line's output to a user's needs, along with other syntactic quirks.



<https://github.com/zaagba/clank>

File extension



Terminators and What They Do:

- ;
Normal terminator, evaluates the statement and discards the value
- ?
Prints the output of a statement/declared variable
- @
Asserts truthiness of a statement, and exits the program if False
- \
Prints the type of evaluated statement and its value

Output	
Hello, world!	
[1, 2, 3]	
13	
<String "Hello, World!">	
<Array [1, 2, 3]>	
<Integer 13>	
3	

Clank allows for indexing 'arrays' with floats that can place values between indexes and can also pass indexes with negative values in order to place values from the right!

```

Array x = [1, 2, 3, 4];
x?

x[1.5]?
x[1.75]?
x[-1.5]?
x[-1.5] = 1.5;

x?

for i in 1...5 {
    i?
}

```

Output	
[1, 2, 3, 4]	
nil	
nil	
nil	
[1, 2, 3, 1.5, 4]	
1	
2	
3	
4	
5	