Burleseque - Moonpage

Roman Müntener

Contents

ABOUT **1**

SYNOPSIS **2**

LANGUAGE **2**

SYNTAX 2

BUILT-INS 2

Add `.+` 2

AddX `_+` 4

Div `./` 5

Duplicate `J ^^` 5

Mul `.*` 6

Sub `.-` 7

Swap `j \/` 8

ABOUT

An interpreter for the esoteric programming language *The Burlesque Programming Language*.

Author: Roman Müntener, 2012-?

Useful Weblinks:

- Burlesque on RosettaCode
- Source code

- Language Reference

Until this moonpage is complete, please consult the Language Reference. Once compelete, the moonpage will superseed the Language Reference.

SYNOPSIS

```
blsq <options>
```

LANGUAGE

SYNTAX

BUILT-INS

Add `.+`

`Int a, Int b:` Integer addition.

```
blsq ) 5 5.+
10
```

`Double a, Double b:` Double addition.

```
blsq ) 5.1 0.9.+
6.0
```

`String a, String b:` Concatenates two strings.

```
blsq ) "ab" "cd" .+
"abcd"
```

`Int a, String b:` Returns the first `a` characters of `b` as a String.

```
blsq ) 3 "abcdef" .+
"abc"
```

`Block a, Block b:` Concatenates two blocks.

```
blsq ) {1 2}{3 4}.+
{1 2 3 4}
```

`Char a, Char b:` Creates a string with the two characters `a` and `b` in it (in that exact order).

```
blsq ) 'a'b.+
"ab"
```

`String a, Char b:` Append `b` to `a`.

```
blsq ) "ab"'c.+
"abc"
```

`Int a, Block b:` Returns the first `a` elements of `b`.

```
blsq ) 2{1 2 3}.+
{1 2}
```

`Block a, Int b:` Returns the first `b` elements of `a`.

```
blsq ) {1 2 3}2.+
{1 2}
```

`String a, Int b:` Returns the first `b` characters of `a` as a String.

```
blsq ) "abc"2.+
"ab"
```

`Double a, Int b:` Convert `b` to Double, then perform addition.

```
blsq ) 1.0 2.+
3.0
```

`Int a, Double b:` Convert `a` to Double, then perform addition.

```
blsq ) 2 1.0.+
3.0
```

AddX _+

`Int a, Int b:` Creates a Block with the two Integers `a` and `b` as elements (in this exact order).

```
blsq ) 1 2_+
{1 2}
```

`Double a, Double b:` Creates a Block with the two Doubles `a` and `b` as elements (in this exact order).

```
blsq ) 1.0 2.0_+
{1.0 2.0}
```

`String a, String b:` Concatenates the two Strings.

```
blsq ) "ab""cd"_+
"abcd"
```

`Block a, Block b:` Concatenates the two Blocks.

```
blsq ) {1}{2}_+
{1 2}
```

`Char a, Char b:` Converts both arguments two string and concatenates them.

```
blsq ) 'a'b_+
"ab"
```

`String a, Char b:` Converts `b` to String, then concatenates.

```
blsq ) "a"'b_+
"ab"
```

`Char a, String b:` Converts `a` to String, then appends it to `b`.

```
blsq ) 'a"b"_+
"ba"
```

`Int a, String b:` Converts `a` to String, then appends it to `b`.

```
blsq ) 1"b"_+
"b1"
```

`String a, Int b:` Converts `b` to String, then concatenates.

```
blsq ) "b"1_+
"b1"
```

Div `./`

`Int a, Int b`: Integer division.

```
blsq ) 10 3./
3
```

`Double a, Double b`: Double division.

```
blsq ) 10.0 3.0./
3.3333333333333335
```

`String a, String b`: Removes `b` from the beginning of `a` iff `b` is a prefix of `a`.

```
blsq ) "README.md" "README" ./
".md"
blsq ) "README.md" "REDME" ./
"README.md"
```

`Block a, Block b`: Removes `b` from the beginning of `a` iff `b` is a prefix of `a`.

```
blsq ) {1 2 3}{1 2}./
{3}
blsq ) {1 2 3}{2 2}./
{1 2 3}
```

`Int a, Double b`: Converts `a` to Double, then divides.

```
blsq ) 10 3.0./
3.3333333333333335
```

`Double a, Int b`: Converts `b` to Double, then divides.

```
blsq ) 10.0 3./
3.3333333333333335
```

Duplicate `J ^^`

Duplicates the top most element.

```
blsq ) 5
5
blsq ) 5J
5
5
```

Mul .*

`Int a, Int b:` Integer multiplication.

```
blsq ) 2 3.*
6
```

`Double a, Double b:` Double multiplication.

```
blsq ) 2.0 3.0.*
6.0
```

`String a, Int b:` Creates a Block containing `a` exactly `b` times.

```
blsq ) "ab"3.*
{"ab" "ab" "ab"}
```

`Char a, Int b:` Creates a String containing `a` exactly `b` times.

```
blsq ) 'a 3.*
"aaa"
```

`Block a, Int b:` Creates a Block containing `a` exactly `b` times.

```
blsq ) {1 2}3.*
{{1 2} {1 2} {1 2}}
```

`String a, String b:` Appends `a` to `b` then reverses.

```
blsq ) "123""456".*
"321654"
```

`Int a, Double b:` Converts `a` to Double, then multiplies.

```
blsq ) 2 3.0.*
6.0
```

`Double a, Int b:` Converts 'b' to Double, then multiplies.

```
blsq ) 2.0 3.*
6.0
```

Sub .-

`Int a, Int b:` Integer subtraction.

```
blsq ) 1 5.-
-4
```

`Double a, Double b:` Double subtraction.

```
blsq ) 1.0 4.0.-
-3.0
```

`String a, String b:` Removes `b` from the end of `a` iff `b` is a suffix of `a`.

```
blsq ) "README.md" ".md".-
"README"
blsq ) "README.md" ".txt".-
"README.md"
```

`Int a, Block b:` Removes the first `a` elements from `b`.

```
blsq ) 3{1 2 3 4}.-
{4}
```

`String a, Int b:` Removes the first `b` characters from `a`.

```
blsq ) "abcd"2.-
"cd"
```

`Int a, String b:` Removes the first `a` characters from `b`.

```
blsq ) 2"abcd".-
"cd"
```

`Block a, Int b:` Removes the first `b` elements from `a`.

```
blsq ) {1 2 3 4}2.-
{3 4}
```

`Int a, Double b:` Converts `a` to Double, then subtracts.

```
blsq ) 4 3.0.-
1.0
```

`Double a, Int b:` Converts `b` to Double, then subtracts.

```
blsq ) 4.0 3.-
1.0
```

`Block a, Block b:` Removes `b` from the end of `a` iff `b` is a suffix of `a`.

```
blsq ) {1 2 3 4}{3 4}.-
{1 2}
blsq ) {1 2 3 4}{3 4 5}.-
{1 2 3 4}
```

Swap `j` \/

Swaps the top two elements.

```
blsq ) 1 2
2
1
blsq ) 1 2j
1
2
```