Table of Contents

int64

This package provides operations for working with 64-bit signed integers (Int64) in MoonBit.

Basic Operations

Int64 values can be created from regular 32-bit integers using from_int. The package also provides constants for the maximum and minimum values representable by Int64.

```

test "basic operations" {
  let i : Int64 = -12345L

  inspect(@int64.from_int(-12345) == i, content="true")


  inspect(@int64.max_value, content="9223372036854775807")
  inspect(@int64.min_value, content="-9223372036854775808")


  inspect(@int64.abs(i), content="12345")
}
```

Binary Representation

The package provides functions to convert Int64 values to their binary representation in both big-endian and little-endian byte order:

```

test "binary conversion" {
  let x = 258L
  let be_bytes = x.to_be_bytes()
  let le_bytes = x.to_le_bytes()


  inspect(
    be_bytes.to_string(),
    content=(
      #|b"\x00\x00\x00\x00\x00\x00\x01\x02"
    ),
  )
  inspect(
    le_bytes.to_string(),
    content=(
      #|b"\x02\x01\x00\x00\x00\x00\x00\x00"
    ),
  )


  let len = be_bytes.length()
  inspect(len, content="8")
}
```

Method-Style Usage

All operations are also available as methods on Int64 values:

```

test "method style" {
  let x = -42L


  inspect(x.abs(), content="42")


  inspect(
    x.to_be_bytes(),
    content=(
      #|b"\xff\xff\xff\xff\xff\xff\xff\xd6"
    ),
  )
}
```

Note that Int64 implements the Hash trait, allowing it to be used as keys in hash maps and members of hash sets.