

Table of Contents

1	uint16
1.1	Constants
1.2	Arithmetic Operations
1.3	Bitwise Operations
1.4	Comparison and Equality
1.5	Default Value and Hashing
1.6	Type Conversions

uint16

The moonbitlang/core/uint16 package provides functionality for working with 16-bit unsigned integers. This package includes constants, operators, and conversions for UInt16 values.

Constants

The package defines the minimum and maximum values for UInt16:

```
1
2  test "UInt16 constants" {
3
4      inspect(@uint16 min_value, content="0")
5
6
7      inspect(@uint16 max_value, content="65535")
8  }
```

Arithmetic Operations

UInt16 supports standard arithmetic operations:

```
1
2  test "UInt16 arithmetic" {
3      let a : UInt16 = 100
4      let b : UInt16 = 50
5
6
7      inspect(a + b, content="150")
8
9
10     inspect(a - b, content="50")
11
12
13     inspect(a * b, content="5000")
14
15
16     inspect(a / b, content="2")
17
18
19     inspect(@uint16 max_value + 1, content="0")
20     inspect(@uint16 min_value - 1, content="65535")
21 }
```

Bitwise Operations

UInt16 supports various bitwise operations:

```

1
2  test "UInt16 bitwise operations" {
3      let a : UInt16 = 0b1010
4      let b : UInt16 = 0b1100
5
6
7      inspect(a & b, content="8")
8
9
10     inspect(a | b, content="14")
11
12
13     inspect(a ^ b, content="6")
14
15
16     inspect(a << 1, content="20")
17     inspect(a << 2, content="40")
18
19
20     inspect(a >> 1, content="5")
21     inspect(b >> 2, content="3")
22 }

```

Comparison and Equality

UInt16 supports comparison and equality operations:

```

1
2  test "UInt16 comparison and equality" {
3      let a : UInt16 = 100
4      let b : UInt16 = 50
5      let c : UInt16 = 100
6
7
8      inspect(a == c, content="true")
9      inspect(a != b, content="true")
10
11
12     inspect(a > b, content="true")
13     inspect(b < a, content="true")
14     inspect(a >= c, content="true")
15     inspect(c <= a, content="true")
16 }

```

Default Value and Hashing

UInt16 implements the Default trait:

```

1
2 test "UInt16 default value" {
3
4     let a : UInt16 = 0
5     inspect(a, content="0")
6
7
8     let value : UInt16 = 42
9     inspect(value hash(), content="42")
10 }

```

Type Conversions

UInt16 works with various conversions to and from other types:

```

1
2 test "UInt16 conversions" {
3
4     inspect((42) to_uint16(), content="42")
5
6
7     let value : UInt16 = 100
8     inspect(value to_int(), content="100")
9
10
11     inspect((-1) to_uint16(), content="65535")
12     inspect((65536) to_uint16(), content="0")
13     inspect((65537) to_uint16(), content="1")
14
15
16     inspect(b'A' to_uint16(), content="65")
17     inspect(b'\xFF' to_uint16(), content="255")
18 }

```