

# Integer types

Cairo uses the following integer types: Type Minimum Maximum u8 0  $2^8 - 1$  u16 0  $2^{16} - 1$  u32 0  $2^{32} - 1$  u64 0  $2^{64} - 1$  u128 0  $2^{128} - 1$  u256 0  $2^{256} - 1$  The `usize` type is an alias for `u32`, and is used for array indexing.

## Integer Creation

All integer types except `u256` can be created as literals using the appropriate suffix. The `u256` type is only supported by construction from other types. 

```
fn main() { let x: u8 = 10_u8; let y = 0xff_u64; // let z = 10_u256; // Error: u256 literals are not supported let z: u256 = u256 { high: 0_u128, low: 10_u128 } // 10 in u256 type }
```

## Integer Operations

Currently, all integer types but `u256` support the operations in the following table.

Operation	Description	Supported by
+	Addition	Yes
-	Subtraction	Yes
*	Multiplication	Yes
/	Integer division	Yes
%	Modulo	No
==	Equality	Yes
!=	Inequality	Yes
<	Less than	Yes
<=	Less than or equal	Yes

Greater than Yes = Greater than or equal Yes | Bitwise or Yes & Bitwise and Yes ^ Bitwise xor Yes

[9.1.2 Felt252 type](#) § [9.1.4 String types](#) §