Math

The math functions used by cosmwasm are based upon standard rust, but helper functions are provided for u128, u64 and decimals.

Uint128

A thin wrapper around u128 that is using strings for JSON encoding/decoding, such that the full u128 range can be used for clients that convert JSON numbers to floats, like JavaScript and jq.

Including in file:use cosmwasm_std::Uint128;

Usefrom to create instances of this andu128 to get the value out:

Uint128(number)

Uint128::new(number)

Uint128::from(number u128/u64/u32/u16/u8)

Uint128::try_from("34567")

Uint128::zero()

checked

All the checked math functions work with Unit128 variables: checked_add, checked_sub, checked_mul, checked_div, checked_div_euclid, checked_rem

saturating

All the saturating math functions work with Unit128 variables: saturating_add, saturating_sub, saturating_mul, saturating_pow

wrapping

All the wrapping math functions work with Unit128 variables: wrapping_add, wrapping_sub, wrapping_mul, wrapping_pow

Uint64

A thin wrapper around u64 that is using strings for JSON encoding/decoding, such that the full u64 range can be used for clients that convert JSON numbers to floats, like JavaScript and jq.

Including in file:use cosmwasm_std::Uint64;

Usefrom to create instances of this andu64 to get the value out:

Uint64(number)

Uint64::new(number)

Uint64::from(number u64/u32/u16/u8)

Uint64::try_from("34567")

Uint64::zero()

checked

All the checked math functions work with Uint64 variables: checked_add, checked_sub, checked_mul, checked_div, checked_div_euclid, checked_rem

saturating

All the saturating math functions work with Uint64 variables: saturating_add, saturating_sub, saturating_mul, saturating_pow

wrapping

All the wrapping math functions work with Uint64 variables: wrapping_add, wrapping_sub, wrapping_mul, wrapping_pow

Decimal

A fixed-point decimal value with 18 fractional digits, i.e. Decimal(1_000_000_000_000_000_000) == 1.0 The greatest possible value that can be represented is 340282366920938463463.374607431768211455 (which is (2^128 - 1) / 10^18)

Including in file:use cosmwasm_std::Decimal;

Decimal::from_str("1234.567")

Decimal::one()

Decimal::zero()

Decimal::percent(50)

Decimal::permille(125)

Decimal::from_ratio(1u128, 1u128) Previous Events Next Compilation * Uint128 * * checked * * saturating * * wrapping * Uint64 * * checked * * saturating * * wrapping * Decimal