4️⃣

Basics IV: Conversions & Special Variables

Uints & Bits

- Uint is a variable of unsigned integers. There is also an int variable for signed integers.
- Uint can be declared in different sizes like `uint8` , `uint32` , `uint96` , `uint256` .
- `uint` is an alias for `uint256` .
- `uint256` ranges from 0 to 2^256 - 1.

```
contract Conversions {

    uint x = 2; // == uint256 x = 2;

    // conversion to smaller sizes costs higher order bits (msb)
    uint32 public a = 0x12345678;
    uint16 public b = uint16(a); // b = 0x5678

    // conversion to larger sizes adds padding bits to the left (msb)
    uint16 public c = 0x1234;
    uint32 public d = uint32(c); // b = 0x00001234;

    // conversion to smaller sizes costs higher order data
    bytes2 public e = 0x1234;
    bytes1 public f = bytes1(e); // f = 0x12;

    // conversion to larger sizes adds padding bits to the right
    bytes2 public g = 0x1234;
    bytes4 public h = bytes4(g); // h = 0x12340000;

    function returnValue() public view returns (bytes4) {
        return bytes4(d);
    }
}
```

Ether Units & Denominations

Denominations

Aa Unit	wei value	ether value
wei	1 wei	10^-18 ETH
kwei	10^3 wei	10^-15 ETH
mwei	10^6 wei	10^-12 ETH
gwei	10^9 wei	10^-9 ETH
microether	10^12 wei	10^-6 ETH
milliether	10^15 wei	10^-3 ETH
ether	10^18 wei	1 ETH

Time Units

Similar to currency, Solidity has time units where lowest unit is second and we can use seconds, minutes, hours, days and weeks as suffixes to denote time.

```
contract LearnUnits {

    function test() public pure returns (string memory){
        assert(1000000000000000000 wei == 1 ether); // 10^18 wei is 1 eth
        assert(1e18 wei == 1 ether);
        assert(2e18 wei == 2 ether);

        assert(1 seconds == 1);
        assert(1 minutes == 60 seconds);
        assert(1 hours == 60 minutes);
        assert(1 hours == 3600 seconds);
        assert(1 days == 24 hours);
        assert(1 weeks == 7 days);

        return "Executed";
    }
}
```

`require()` should be your go to function for checking conditions, `assert()` is just there to prevent anything really bad from happening, but it shouldn't be possible for the condition to evaluate to `false`

Global Variables

Global variables (special variables) are globally available variables and they provide information about blockchain.

Examples:

- `msg.sender` : sender of the current call
- `msg.value` : amount of wei sent with the message
- `block.timestamp` : current block timestamp since unix epoch in seconds (uint)
- `block.number` : current block number (uint)