

EXCEPTION HANDLING

Require/Assert/Revert and Throw

SEND/TRANSFER

- `Address.send()`
 - `Address.call()`
 - Low Level calls
 - Return boolean
 - Don't delegate exceptions
- Bad Design
- `Address.transfer()`
 - Delegate gas
 - Delegate exception
 - Revert everything if something fails

EXCEPTIONS

- Throw;
 - Can't really catch!
 - No Message, just revert everything and throw exception
- Better solution?
- Assert/Require
 - `Assert(1==1);`
 - `Require(balance[msg.sender] >= amount);`
 - Clearer structure

BEHAVIOR ASSERT STYLE EXCEPTIONS

- Assert-Style Exceptions
 - Consume all gas if thrown!!!
 - Revert all changes made
 - E.g. access array out of bounds
 - Shift by a negative amount
 - Division by zero
 - Or „assert(...)“ with an argument that evaluates to false
- And more:
- <http://solidity.readthedocs.io/en/latest/control-structures.html#error-handling-assert-require-revert-and-exceptions>

BEHAVIOR REQUIRE-STYLE EXCEPTIONS

- Require Style Exceptions
 - Return all gas
 - Revert all changes made
 - E.g. „throw;“
 - „require(..)” with an argument evaluates to false
 - Call another contract and this throws an exception
 - Create a contract (new) and throws an exception
 - .transfer(...) fails
 - Receive ether without „payable” modifier
- And more: <http://solidity.readthedocs.io/en/latest/control-structures.html#error-handling-assert-require-revert-and-exceptions>

EXAMPLES

- Find it in the course repository
- Lets look at some examples NOW.