**Testing events**

**Testing events**

Picking up from where we left in the previous lesson. The only point left is:

4. Our function emits the `EnteredRaffle` event.

Before jumping into the test writing we need to look a bit into the cheatcode that we can use in Foundry to test events: [expectEmit](https://book.getfoundry.sh/cheatcodes/expect-emit?highlight=expectEm#expectemit).

The first step is to declare the event inside your test contract.

So, inside RaffleTest.t.sol declare the following event:

event EnteredRaffle(address indexed player);

Then we proceed to the test:

function testEmitsEventOnEntrance() public {

// Arrange

vm.prank(PLAYER);

// Act / Assert

vm.expectEmit(true, false, false, false, address(raffle));

emit EnteredRaffle(PLAYER);

raffle.enterRaffle{value: entranceFee}();

}

* We prank the PLAYER
* We call the expectEmit cheatcode - vm.expectEmit(true, false, false, false, address(raffle)); I know this looks a bit weird. But let's look at what expectEmit expects:

function expectEmit(

bool checkTopic1,

bool checkTopic2,

bool checkTopic3,

bool checkData,

address emitter

) external;

function expectEmit(

bool checkTopic1,

bool checkTopic2,

bool checkTopic3,

bool checkData,

address emitter

) external;

* The checkTopic 1-3 corresponds to the indexed parameters we are using inside our event. The checkData corresponds to any unindexed parameters inside the event, and, finally, the expectEmit expects the address that emitted the event. It looks like this vm.expectEmit(true, false, false, false, address(raffle)); because we only have one indexed parameter inside the event.
* We need to manually emit the event we expect to be emitted. That's why we declared it earlier;
* We make the function call that should emit the event.

Run the test using the following command: forge test --mt testEmitsEventOnEntrance

Everything passes, amazing!