**Implementing console log in your smart contract**

**Debugging using console log**

Do you remember how inside our Raffle.t.sol we did import {Test, console} from "forge-std/Test.sol"; to import the console? And then again, we did the same thing inside the Script files? This is not limited only to testing and scripting. Let's demonstrate:

1. Add the following in the import section of your Raffle.sol file: import {console} from "forge-std/Script.sol";
2. Change the enterRaffle function as follows:

function enterRaffle() public payable {

if (s\_raffleState == RaffleState.CALCULATING) revert Raffle\_\_RaffleNotOpen();

if (msg.value < i\_entranceFee) revert Raffle\_\_NotEnoughEthSent();

console.log("Debugging at its finest");

s\_players.push(payable(msg.sender));

emit EnteredRaffle(msg.sender);

}

1. Run forge test --mt testRaffleRecordsPlayerWhenTheyEnter -vv.

The result:

Ran 1 test for test/unit/RaffleTest.t.sol:RaffleTest

[PASS] testRaffleRecordsPlayerWhenTheyEnter() (gas: 71482)

Logs:

Creating subscription on ChainID: 31337

Your sub Id is: 1

Please update subscriptionId in HelperConfig!

Funding subscription: 1

Using vrfCoordinator: 0x90193C961A926261B756D1E5bb255e67ff9498A1

On ChainID: 31337

Adding consumer contract: 0xBb2180ebd78ce97360503434eD37fcf4a1Df61c3

Using VRFCoordinator: 0x90193C961A926261B756D1E5bb255e67ff9498A1

On chain id: 31337

Debugging at its finest

Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 11.59ms (84.60µs CPU time)

You can see the Debugging at its finest message at the end of the log. Super nice!

**Note:** Make sure to delete those before deploying to mainnet, because this will cost gas, and you do not want to spend that!