**Interact with a smart contract**

**Basic NFT Interactions**

Alright, with our tests passing we're going to want a way to interact with our contract programmatically. We could use cast commands, but let's write an interactions script instead. Create the file script/Interactions.s.sol. You know the drill for our boilerplate by now.

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.18;

import {Script} from "forge-std/Script.sol";

contract MintBasicNft is Script{

function run() external {}

}

We know we'll always want to be interacting with the latest deployment, so let's install the foundry-devops library to help with this.

forge install Cyfrin/foundry-devops --no-commit

Now, we can import DevOpsTools and use this to acquire our most recent deployment. We'll use this address as a parameter for the mint function we'll call.

❗ **NOTE** I've copied over my PUG tokenUri for use in our mint function, remember to copy your own over too!

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.18;

import {Script} from "forge-std/Script.sol";

import {BasicNft} from "../src/BasicNft.sol";

import {DevOpsTools} from "lib/foundry-devops/src/DevOpsTools.sol";

contract MintBasicNft is Script{

string public constant TOKENURI =

"ipfs://bafybeig37ioir76s7mg5oobetncojcm3c3hxasyd4rvid4jqhy4gkaheg4/?filename=0-PUG.json";

function run() external {

address mostRecentlyDeployed = DevOpsTools.get\_most\_recent\_deployment("BasicNft", block.chainid);

mintNftOnContract(mostRecentlyDeployed);

}

function mintNftOnContract(address contractAddress) public {

vm.startBroadcast();

BasicNft(contractAddress).mintNft(TOKENURI);

vm.stopBroadcast();

}

}

❗ **PROTIP** Remember, if you don't recall which parameters are required for a function like get\_most\_recent\_deployment you can ctrl + left-click (cmd + click) to be brought to the function definition.

**Wrap Up**

That's all there is to our interactions script, albeit we're only interacting with a single function, great work nonetheless!

In the next lesson we'll look at deploying our contract to a testnet and using our script to test interacting with it on-chain.