# 79.https://stackoverflow.com/questions/73053421/how-to-make-minting-function-that-takes-0-1-ethereum-in-solidity

**T:**How to make minting function that takes 0.1 ethereum in Solidity?

**Q:**Can someone please explain how can I make a function that mints a token for 0.1 eth in Solidity and verify it in HardHat? I have done this so far:  
  
HardHat:  
  
[owner] = await ethers.getSigners();const Nft = await ethers.getContractFactory("contract");const nft = await Nft.deploy(owner.address);prov = ethers.getDefaultProvider();let balance = await prov.getBalance(owner.address);console.log(balance); <-- evaluates to 10000000000000await hoodie.mint({ value: ethers.utils.parseEther("0.1") });console.log(balance); <-- still evaluates to 10000000000000  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Solidity:  
  
function mint() payable public returns (uint256) {; require(msg.value == 0.1 ether || msg.value == 100000000000000000 wei, "Transaction amount has to be 0.1 eth"); \_safeMint(msg.sender, token\_id); return token\_id; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Thanks in advance!

**C1:**if I understand correctly, you want to mint an NFT and transfer 0.1 ether to the contract, right?

2 **Answer**

**A1:**in that case,  
  
you may try adding approve and transferFrom methods.  
  
approve is needed to make you approve the fund transfer  
  
transferFrom is needed to make the fund transfer happen  
  
the contract:  
  
function mint() payable public returns (uint256) { require(msg.value == 0.1 ether || msg.value == 100000000000000000 wei, "Transaction amount has to be 0.1 eth"); IERC20(\*the ETH address here\*).approve(msg.sender, msg.value); IERC20(\*the ETH address here\*).transferFrom(msg.sender, address(this), msg.value); \_safeMint(msg.sender, token\_id); return token\_id; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Is this implementation also true for ERC721, forgot to mention that I was using that standard?

**C2:**yes, it's all about transfering funds from an address to another

**A2:**You need to use payable to transfer native token such as ETH or BNB in smart contract.  
  
And then you can call \_safeMint() private function with token ID.You also need to increase token ID after mint.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
function mint() payable public returns (uint256) { require(msg.value == 0.1 ether || msg.value == 100000000000000000 wei, "Transaction amount has to be 0.1 eth"); payable(this).transfer(msg.value); \_safeMint(msg.sender, token\_id); token\_id.increament(); return token\_id;}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]