# 517.https://stackoverflow.com/questions/70618106/how-to-call-a-erc20-contract-fucntion-inside-erc721-contract

**T:**how to call a ERC20 contract fucntion inside ERC721 contract

**Q:**what I am trying to accomplish is calling a function defined in ERC20 contract from ERC721 contract as shown below, specifically transferFrom function from ERC20 contract inside the same function in ERC721.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
This does not compile.what it ultimately does is that everytime a erc721 NFT transfer occurs, royalty is tranferred to the creator in the form of erc20 tokens.  
  
import "./myToken.sol" as erc20contract //This is my ERC20 contractcontract ERC721 is ERC721Interface, myToken { function transferFrom(address \_from, address \_to, uint256 \_tokenId) public payable { address addr\_owner = ownerOf(\_tokenId); require( addr\_owner == \_from, "\_from is NOT the owner of the token" ); require( \_to != address(0), "Transfer \_to address 0x0" ); address addr\_allowed = allowance[\_tokenId]; bool isOp = operators[addr\_owner][msg.sender]; require( addr\_owner == msg.sender || addr\_allowed == msg.sender || isOp, "msg.sender does not have transferable token" ); //transfer : change the owner of the token tokenOwners[\_tokenId] = \_to; balances[\_from] = balances[\_from].sub(1); balances[\_to] = balances[\_to].add(1); //reset approved address if (allowance[\_tokenId] != address(0)) { delete allowance[\_tokenId]; } erc20contract.transferFrom(\_to, nftInfo[\_from].creator, 10); // This is what I'd like to achieve { emit Transfer(\_from, \_to, \_tokenId); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

1 **Answer**

**A1:**Edit: I've just noticed your contract seemingly implements both the ERC721 and ERC20 standards. You'll need to decide on one, but using composition (as I've shown below) might be sufficient to provide all the necessary functionality you're after.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Unless I'm missing something, you only seem to lack an actual instance of your ERC20 token. For example:  
  
contract YourToken is ERC721 { ERC20 private immutable token; constructor(ERC20 t) { token = t; } function transferFrom(address \_from, address \_to, uint256 \_tokenId) public payable { // Your code here. token.transferFrom(\_from, \_to, 10); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**@Tadeg, Thanks. But I need to separately implement both ERC20 and 721 standards as my dapp allows users to buy erc20 tokens with Eth and mint/trade 721 NFT tokens.