# 12.https://stackoverflow.com/questions/71914170/how-to-burn-erc721-token

**T:**How to burn erc721 token?

**Q:**I am trying to burn erc721 token. I have inherited ERC721Burnable contract,but the transaction gets failed.While debugging I found that in ERC721URIStorage contract,it reverts back at:  
  
if (bytes(\_tokenURIs[tokenId]).length != 0) { delete \_tokenURIs[tokenId];}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Also I want the owner of the contract to have the ability to modify the attributes of metadata,even after the token is minted and transferred. This is my contract:  
  
// SPDX-License-Identifier: MITpragma solidity ^0.8.4;import "@openzeppelin/contracts/token/ERC721/ERC721.sol";import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Burnable.sol";import "@openzeppelin/contracts/access/Ownable.sol";import "@openzeppelin/contracts/utils/Counters.sol";contract MyNFT is ERC721, ERC721Enumerable, ERC721URIStorage, ERC721Burnable, Ownable { using Counters for Counters.Counter; Counters.Counter private \_tokenIdCounter; uint MAX\_SUPPLY=1; constructor() ERC721("MyNFT", "MNFT") {} function safeMint(address to, string memory uri) public onlyOwner { require(totalSupply() < MAX\_SUPPLY, "Can be minted only one time."); uint256 tokenId = \_tokenIdCounter.current(); \_tokenIdCounter.increment(); \_safeMint(to, tokenId); \_setTokenURI(tokenId, uri); } // The following functions are overrides required by Solidity. function \_beforeTokenTransfer(address from, address to, uint256 tokenId) internal override(ERC721, ERC721Enumerable) { super.\_beforeTokenTransfer(from, to, tokenId); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

1 **Answer**

**A1:**The code you provided works (although it's missing an ending } ). You must be trying to burn a token that doesn't exist (maybe it's already been burned)?  
  
To change the props of the NFT after it's already been minted, all you need to do is just return different properties in the return value from the token URI.