# 335.https://stackoverflow.com/questions/71554048/compiling-an-erc721uristorage-contract

**T:**Compiling an ERC721URIStorage contract

**Q:**This question is essentially a follow-up to this question. I have been trying to follow Alchemy documentation to create an NFT contract but encountered the need (as described in the linked question) to import ERC721URIStorage. However, I now get a number of compilation problems that do not make clear sense to me.  
  
In response to the first error (see below), I have tried adding in the import statement for ERC721: import "@openzeppelin/contracts/token/ERC721/ERC721.sol"; This did not change anything in the set of compilation errors.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

2 **Answer**

**A1:**It should be kinda like this :>  
  
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";import "@openzeppelin/contracts/access/Ownable.sol";import "@openzeppelin/contracts/utils/Counters.sol";contract LeafNFT is ERC721, ERC721URIStorage, Ownable { using Counters for Counters.Counter; Counters.Counter private \_tokenIdCounter; constructor() ERC721("LeafNFT", "NFL") {} function safeMint(address to, string memory uri) public onlyOwner { uint256 tokenId = \_tokenIdCounter.current(); \_tokenIdCounter.increment(); \_safeMint(to, tokenId); \_setTokenURI(tokenId, uri); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Thanks for the help. Unfortunately, I get errors similar to the following: "TypeError: Derived contract must override function "\_burn". Two or more base classes define function with same name and parameter types." Apparently the compiler does not know which function to use when there are definitions in multiple parents.

**C2:**it seems like your parent function does not have the override keyword :/

**A2:**I was able to get the code to work with a slight variation. The relevant points of confusion seem to be:  
  
 ● Whether zero, one, or two of {ERC721, ERC721URIStorage} should be imported. The answer appears to be to import only ERC721URIStorage.  
  
 ● Whether the contract is defined as ERC721URIStorage, ERC721, Ownable, or as ERC721URIStorage, Ownable. The answer appears to be the latter.  
  
 ● Whether the constructor uses ERC721 or ERC721URIStorage. The answer seems to be ERC721.  
  
I think the first two of these points are intuitive (they are the first thing I tried and appear in the code shown in the question). This is because ERC721URIStorage inherits from ERC721, so there is no need to explicitly import ERC721 or define the contract as such.  
  
The third point was less expected, but in hindsight, I think the constructor must cite ERC721 because ERC721URIStorage does not have a constructor of its own and because there is ambiguity about where a constructor can be inherited from (either from a parent of ERC721URIStorage or from Ownable).  
  
Below is the working contract: