# 460.https://stackoverflow.com/questions/70886749/how-to-pass-in-arguments-to-a-contracts-constructor-for-initializing-characters

**T:**How to pass in arguments to a contract's constructor for initializing characters images and attributes

**Q:**I have a contract like this that has a constructor like this:  
  
constructor( string[] memory characterNames, string[] memory characterImageURIs, uint256[] memory characterHp, uint256[] memory characterAttackDmg, string memory bossName, string memory bossImageURI, uint256 bossHp, uint256 bossAttackDamage) ERC721("Heroes", "HERO") { for (uint256 i = 0; i < characterNames.length; i += 1) { defaultCharacters.push( CharacterAttributes({ characterIndex: i, name: characterNames[i], imageURI: characterImageURIs[i], hp: characterHp[i], maxHp: characterHp[i], attackDamage: characterAttackDmg[i] }) ); CharacterAttributes memory c = defaultCharacters[i]; console.log( "Done initializing %s w/ HP %s, img %s", c.name, c.hp, c.imageURI ); } bigBoss = BigBoss({ name: bossName, imageURI: bossImageURI, hp: bossHp, maxHp: bossHp, attackDamage: bossAttackDamage }); console.log( "Done initializing boss %s w/ HP %s, img %s", bigBoss.name, bigBoss.hp, bigBoss.imageURI ); \_tokenIds.increment();}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Question  
  
 ● for the characters names and HP and Attack... how should I type them is it in a JSON or in arrays or what?  
  
 ● for the imageURI how can I parse them to the constructor and where as I'm using Pinata IPFS to host my images?  
  
Please if you can help me with this.

1 **Answer**

**A1:**1- type for constructor arguments are already defined:  
  
 ● string[] = array of strings  
  
 ● string =string  
  
 ● uint (uint256) = 256 bits unsigned number ranging from 0 to 2²⁵⁶  
  
2- constructor arguments mean, when you create the contract, you have to pass those parameters inorder to initialize the contracts. So you should already have them in hand.  
  
you store the metadata in ipfs: an example would be like this:  
  
{"name": "name","description": "descruption","image": "ipfs://cid/name.jpeg"}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
So when you work in front-end, you make a request to ipfs Uri, to get the metadata:  
  
 // make a request with axios library // this will store the above json data inside `metaData`.data const metaData = await axios.get(tokenUri) const data=metaData.data  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
From metadata, you get the imageUri as data.image  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]