# **How To Create NFTs With Solidity**

## A glorious guide to building and creating NFTs, the ERC-721 standard for creating collectibles, art, and any type of unique asset on-chain.

Patrick Collins

[Patrick Collins](https://medium.com/@patrick.collins_58673?source=post_page-----4fa1398eb70a-----------------------------------)

[Mar 2, 2021·7 min read](https://betterprogramming.pub/how-to-create-nfts-with-solidity-4fa1398eb70a?source=post_page-----4fa1398eb70a-----------------------------------)

[](https://betterprogramming.pub/how-to-create-nfts-with-solidity-4fa1398eb70a?source=post_page-----4fa1398eb70a-----------------------------------)

Photo by [Anton Maksimov juvnsky](https://unsplash.com/@juvnsky?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral).

[NFTs](https://eips.ethereum.org/EIPS/eip-721) (Non-Fungible Tokens) are the hot new up-and-comers in the smart contract space. If 2020 was the year of DeFi, then at least the start of 2021 belongs to NFTs. NFTs are a token standard similar to the [ERC20](https://www.investopedia.com/news/what-erc20-and-what-does-it-mean-ethereum/).

A Non-Fungible Token means that it’s a unique token that has no other token like it. This is starkly different from the ERC20s, which are fungible. Fungible means “replaceable” or “interchangeable.” For example, your dollar bill is going to be worth $1 no matter what dollar bill you use. The serial number on the dollar bill might be different, but the bills are interchangeable since they’ll be worth $1 no matter what.

There are a lot of ERC20s, like MKR, AAVE, and SNX. There are not a lot of NFTs (or ERC721s). In fact, there is just one of each, hence its uniqueness. These can be built and programmed to do anything you please, just like with regular smart contracts, but they come with cryptographically proven authenticity since their history of deployment will always be guaranteed. This solves a number of problems — particularly for those in the art world —with deciphering the authenticity of a piece of art. This can also solve issues with royalties, create digital status symbols, allow gaming applications to be interoperable, and so much more.

# Where Are They Now?

And they have a lot of value. Recently, Axie Infinity [just sold nine land plots for about $1.5 million](https://www.blockchaingamer.biz/features/14926/most-expensive-game-nft-digita-assets/).



Tweet by [@seedphrase](https://twitter.com/seedphrase/status/1358918272767324160).

People are getting pretty excited about these. NFTs are a new venture enabled by the advent of blockchain and smart contracts, and people are building amazing things with them.

So let’s teach you everything you need to know about NFTs.

# Standards

As we mentioned, NFTs start with the [ERC721 token standard](https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC721/ERC721.sol), which has a similar syntax to that of ERC20s — with a few tweaks.

In order for your contract to be considered an NFT, all it has to do is follow this standard. We make it easy by just importing the standard into our contracts so that we don’t have to reinvent the wheel every time we want to make a new one. See the code below for what I mean.

NFTs have a tokenURI variable that we will talk about in a minute, and they have a mapping of a tokenId to its owner, where each token has its own “owner.” This differentiates them from the ERC20s, which only have a mapping of an address to a balance. ERC721s still let people transfer the tokens, set permissions on the tokens, and more. It’s a lightweight standard by convention so that we can build whatever we want with them.

Now here is where things get *really* different. Metadata. When you go on to [OpenSea](https://opensea.io/) (a popular NFT marketplace), you can see a bunch of images. Since NFTs are on the blockchain, all their data should be on-chain right? Well, not quite.

Storing data on-chain can get really expensive, and art isn’t known for its small size. Ethereum and smart contract developers realized that uploading even a 1MB image could break their bank accounts, so they wanted to come up with a way to display art without having to upload the entire image.

As a workaround, most NFTs have what’s called a tokenURI. This is a globally unique identifier for all the image aspects of an NFT. This makes it easier to give NFTs visuals. A URI is a Uniform Resource Identifier, which could be an HTTPS API call, something through IPFS, or some other type of unique identifier. This metadata looks like this:

They are JSON files that have:

* name
* description
* image
* attributes

These are generally stored in an API or on IPFS.

Now, if you’re like me, you’re thinking, “Wait… that means the image comes from a centralized location. That defeats the purpose of me deploying to the blockchain.”

# On-Chain Metadata vs. Off-Chain Metadata

If you want your NFT to do anything interesting, in my mind, it needs to have on-chain attributes. A few examples include if you wanted to give them stats to battles like in *Pokemon*, some type of rarity stat, or have them have cryptographically guaranteed scarcity. The attributes would have to be on-chain.

Right now, NFT platforms don’t have great ways to visualize the on-chain attributes, so you just have to create a tokenURI with your minted NFTs.

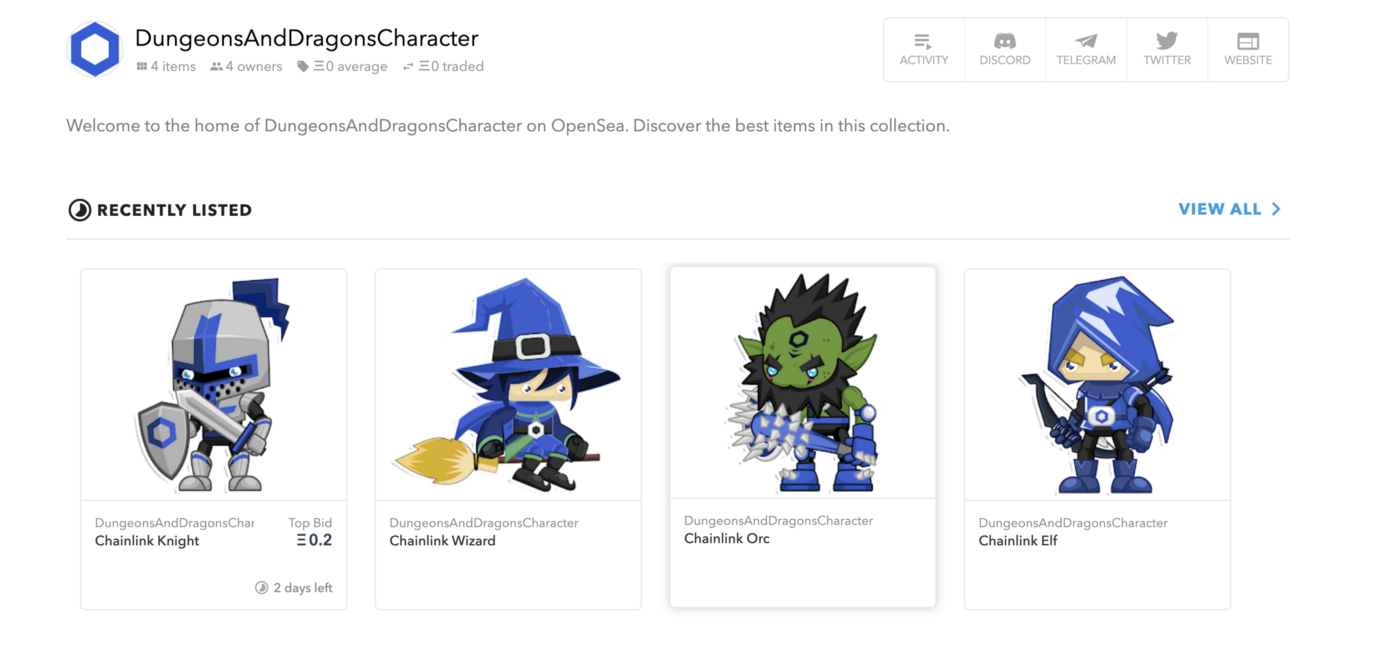


Image from [OpenSea D&D Characters](https://opensea.io/collection/dungeonsanddragonscharacter).

For the tokenURI, one of the most popular approaches is using [IPFS](https://ipfs.io/) to store your data. You’ll upload your data there and then use a pinning service to make sure the data stays there forever. I’m excited for when we’ll have more interoperability between our smart contract platforms and our storage.

*Hint: This goes into my blockchain full-stack theory. Smart contract platform on the front, Chainlink in the middle, dStorage on the back… but that’s another conversation.*

*NFT marketplace*

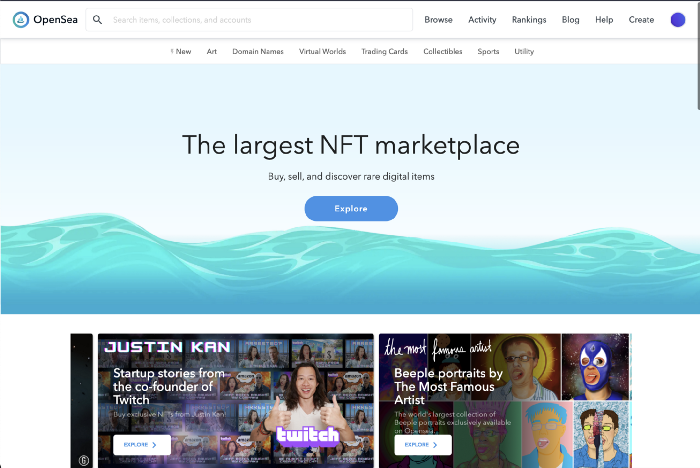
**

Image from [OpenSea](https://opensea.io/).

To be honest, I thought NFTs were stupid at first. Then again, I’m also the type of dude who used to think art was dumb… but I have to remember I love music, movies, and games with good visuals, so I’d be a hypocrite if I said I didn’t like art. NFTs solve an issue that the art world is facing with royalties and authenticity. We now have a decentralized auditing service. If you want to use someone’s art, you can see everything about that piece of work on-chain.

The original creator of the Nyan [cat sold an NFT version of it for 300 ETH](https://www.scmp.com/magazines/style/tech-design/article/3123302/would-you-pay-us590000-meme-nyan-cat-just-sold-six). There is value in authenticity. If you create something amazing, there is value in scarcity. This is why I think having Chainlink VRF NFTs is so outstandingly amazing since you can create collectibles that are so rare that there could be a chance only you have it. I’m excited for when someone makes a *Pokemon*-type game on these platforms. These NFTs are here to stay.

There are a ton of NFT platforms [that are skyrocketing in value](https://coinmarketcap.com/nfts/), and we can do some really cool stuff with them.

Let’s look at my nft-mix I made on the [Brownie](https://eth-brownie.readthedocs.io/en/stable/) platform. If you want an end-to-end demo showing how to upload to IPFS as well, check out [the Chainlink blog](https://blog.chain.link/build-deploy-and-sell-your-own-dynamic-nft/) we created that deployed these adorable little dungeons and dragons characters you saw above. Check out the description to see the code and the blog on that.

Let’s look at the code for a simple NFT. All the code for this can be found on the [nft-mix](https://github.com/PatrickAlphaC/nft-mix) repo on GitHub and the [*Dungeons and Dragons* code can be found on GitHub as well](https://github.com/PatrickAlphaC/dungeons-and-dragons-nft).

To get the overall understanding of everything above, feel free to check out our youtube video on it.

A Glorious Guide to NFTs

# The Simple NFT Contract

This is the simplest way to spin up an NFT contract. This is the syntax for Truffle, Hardhat, and Brownie, and it won’t work in Remix. I love working with [OpenZepplin](https://github.com/OpenZeppelin/openzeppelin-contracts) contracts because they are *nice* and have all the tools we usually need to get started. So be sure to install these with:

npm install @openzeppelin/contracts

We create a token with the DOG symbol and the name is Dogie. We can then mint as many DOGs as we want with the createCollectible function, which stores us a new tokenId every time we do so. All we need to do is pass a tokenURI, which is just any URL/URI that points to something in the metadata JSON format of:

{

"name": "Name",

"description": "Description",

"image": "URI",

"attributes": []

}

This is nice, but let’s level up. If you want a walkthrough of the simple NFT, check this video.

Simple NFT deployment in Solidity

# The Advanced NFT Contract

Now we’re talking! This one used the [Chainlink VRF](https://docs.chain.link/docs/get-a-random-number) to give our NFTs random stats. In this case, we are using randomness to give our dog a random breed!

When we call createCollectible this time, we actually send a request out to a Chainlink oracle to return a cryptographically proven random number. This way, we don’t have any tampering going on that could potentially ruin the fairness of our game. The request is asynchronous, and we have to wait for the Chainlink oracle to make a second transaction back with the random number.

Once the Chainlink oracle is done, it calls the fulfillRandomness function with its random number, and that’s what mints the NFT. The mappings are used to make sure that whoever initially hit the createCollectible function gets to be the owner of the NFT. This is how we can start making dynamic NFTs with incredibly powerful features and true scarcity.

For a walkthrough on the advanced NFT, check out this walkthrough.

Here is the FULL walkthrough:

And here is just the advanced one:

Advanced NFT Deployment in Solidity

# Conclusion

NFTs are awesome. Feel free to leave any comments or questions, and we will talk soon!