# 608.https://stackoverflow.com/questions/69339123/how-is-a-nft-token-id-assigned-to-an-object-e-g-picture-tweet-video-etc

**T:**How is a NFT Token ID assigned to an object (e.g., picture, tweet, video, etc.)

**Q:**Background: Suppose I want to verify the ownership of some tweet or some picture by looking up the data in the blockchain. If my understanding is correct, I need to be able to assign a unique tokenId that represents that tweet/picture.  
  
Question 1: Is there some standardization on how this ID is assigned? Do different platform perform this mapping in a unique way? I feel like without such a standardization, the non-fungibility becomes questionable (e.g., do you really own this tweet, or do you only own it if you apply the tweet->token ID mapping you crafted yourself?).  
  
Question 2: Could you provide a few examples of how exactly is the token ID assigned to some (well-known) NFTs?

2 **Answer**

**A1:**Is there some standardization on how this ID is assigned?  
  
The ERC-721 standard explicitly states that there is no standard to assign the ID (except for the uint256 datatype):  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
While some ERC-721 smart contracts may find it convenient to start with ID 0 and simply increment by one for each new NFT, callers SHALL NOT assume that ID numbers have any specific pattern to them, and MUST treat the ID as a “black box”.  
  
e.g., do you really own this tweet, or do you only own it if you apply the tweet->token ID mapping  
  
Token ownership does not mean that you own the underlying resource. It only means that you own the token (representing the resource).  
  
Could you provide a few examples of how exactly is the token ID assigned to some (well-known) NFTs?  
  
 ● CryptoKitties - link, line 412, incrementing  
uint256 newKittenId = kitties.push(\_kitty) - 1;  
  
  
  
 ● CryptoPunks - link, lines 73 and 83, assigning ID set by the (authorized) caller  
mapping (uint => address) public punkIndexToAddress;  
  
function setInitialOwner(address to, uint punkIndex) {  
 // ...  
 punkIndexToAddress[punkIndex] = to;

**C1:**I see. So is there any connection between e.g., the famous Jack Dorsey tweet and the underlying token? Like, what prevents me from simply minting another token, claiming its connected to the tweet, and selling it on an arbitrary NFT exchange?

**C2:**@Zuza Technically, the token metadata probably links to the tweet or contains its hash or something (I'm not really sure about this particular case). And the tweet doesn't need to link to the token. So from the technical point of view - nothing prevents you from doing so... From the legal standpoint, that's a different story.

**A2:**When an NFT is minted with a smart contract, associated with the NFT TokenID, the block chain also typically records a URI (universal resource identifier) that points to the NFT metadata (usually a json file). This URI can be obtained by querying the blockchain via the smart contract address and TokenID. The json file itself includes, among other things, another URI that points to the digital asset linked with the NFT. If immutable URIs are used (e.g. IPFS URIs), the URI recorded in the block chain for the NFT TokenID is unambiguously linked to the unchangeable json file which unambiguously links to the unchangeable digital asset. As Petr correctly points out, technologically, nothing prevents anyone else from minting another NFT that also records the same URI and therefore points to the same json file (and by transitivity to the same digital asset). I've tried to be precise in the preceding statements to avoid incorrect extrapolatory claims that would actually depend on the smart contract implementation details.