

[초보 서버 개발자](#)[Search...](#)[홈](#)[Github](#)[태그](#)[방명록](#)[글쓰기](#)

BlockChain and NFT/NFT

# [NFT] NFT 거래를 통해 기술적 측면 알아보기

Dbswnstjd 2022. 2. 24. 16:48

## ***NFT 거래 하기***

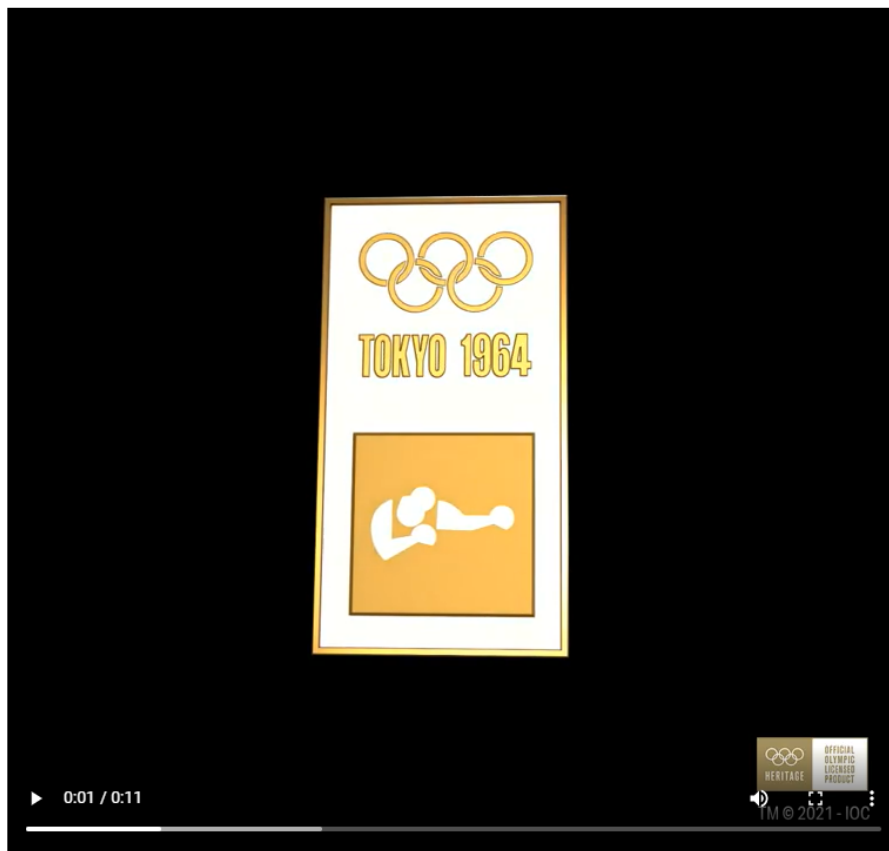
해외 유명거래소를 이용하려고 찾아보았는데 OpenSea라는 곳이 제일 많이 거래하는 곳이었던 것 같다. 하지만 이더리움을 보내는 수수료가 거의 6만원 가까이 되었기 때문에 국내로 알아보다가 얼마전 업비트가 NFT거래를 열어준 것을 확인 하였다. 수수료 문제도 있고 간단하게 NFT를 구매하기에는 업비트가 가장 적당해 보여서 업비트를 선택하게 되었다.

## ***NFT 정보***

## 에디션 #97 거래 이력

| 거래일시                | 마켓 ?        | 판매자     | 소유자     | 금액         |
|---------------------|-------------|---------|---------|------------|
| 2022.02.24<br>15:50 | Marketplace | 예**1    | y*****a | 39,000 KRW |
| 2021.08.17<br>17:34 | Event       | 업*****람 | 예**1    | 0 KRW      |

### 거래 이력



Boxing Tokyo 1964 Pictogram

### 체인정보

|         |  |
|---------|--|
| 컨트랙트 주소 | 0x00dbD4a3C42a1d9c0807Bb1A5A3b95D6aef58001 |
| 토큰 ID   | 403  |

### 체인 정보

업비트에서 NFT를 직접 구매 하면서 여러가지 정보를 얻을 수 있었다.

체인정보에 있는 컨트랙트 주소를 통해 컨트랙트가 어떤 식으로 진행되고 토큰 ID는 어떤 식으로 보내지는지 알아보았다.

이를 알기 전에 '스마트 컨트랙트' 를 이해해야 한다.

## 스마트 컨트랙트란 ?

예를 들어 물건을 산다고 할 때 '상대방이 돈을 입금하면 자동으로 물건을 전송하게 함' 과 같이 설계를 하거나 '돈을 입금하면 구매자의 소유로 변경' 등으로 설계가 가능하다.

이와 같이 진행되도록 설계한 것이 스마트 컨트랙트이다.

- 특정한 조건을 정해놓고, 그 조건이 충족되면 자동으로 계약이 이루어지도록 설계해 놓은 것
- 중간에서 위조와 변조가 어려움
- 중계자가 필요하지 않고 투명한 계약이기 때문에 다른 누구도 검증이 가능

```
pragma solidity ^0.4.20;
```

```
interface ERC721 /* is ERC165 */ {
```

```
    event Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId);
```

```
    event Approval(address indexed _owner, address indexed _approved, uint256 indexed _tokenId);
```

```
    event ApprovalForAll(address indexed _owner, address indexed _operator, bool _approved);
```

```
    function balanceOf(address _owner) external view returns (uint256);
```

```
    function ownerOf(uint256 _tokenId) external view returns (address);
```

```
    function safeTransferFrom(address _from, address _to, uint256 _tokenId, bytes data) external payable;
```

```
    function safeTransferFrom(address _from, address _to, uint256 _tokenId) external payable;
```

```
    function transferFrom(address _from, address _to, uint256 _tokenId) external payable;
```

```
    function approve(address _approved, uint256 _tokenId) external payable;
```

```
    function setApprovalForAll(address _operator, bool _approved)
```

```
external;
```

```
function getApproved(uint256 _tokenId) external view returns
(address);
```

```
function isApprovedForAll(address _owner, address _operator) external
view returns (bool);
}
```

```
interface ERC165 {
function supportsInterface(bytes4 interfaceID) external view returns
(bool);
}
```

```
interface ERC721TokenReceiver {
function onERC721Received(address _operator, address _from, uint256
_tokenId, bytes _data) external returns(bytes4);
}
```

*NFT 개발에 사용된 ERC-721 프로토콜에서 Solidity라는 언어를 사용*

여기서 몇가지 내용을 알 수 있었다.

- getApproved는 매개변수로 받은 토큰ID를 소유하고 있는 주소를 반환하는 함수이다.
  - ownerOf는 매개변수로 받은 토큰ID의 소유자 주소를 반환하는 함수이다.
- 그리고 찾아본 결과 ERC-721은 항상 토큰을 전송할 때 토큰 ID를 받아 해당 ID의 토큰을 전송한  
다고 한다. 이에 따라 NFT는 대체 가능 토큰과 달리 토큰을 분할해서 사용이 불가능 한 것이다.  
따라서 NFT는 '희소성'을 갖게 되고 대체 불가능 토큰이라고 불리는 이유이다.

## Fungible-Token



Balance : 3

## Non-Fungible-Token



Balance : 3