



# ТОКЕНІЗАЦІЯ АКТИВІВ

ЛЕКЦІЯ 6




# ПЛАН

- 1) Поняття токенизації
- 2) Токен. Класифікація токенів
- 3) Взаємозамінний (FT) і незамінний (NFT) токени
- 4) Algorand Standard Asset
- 5) Створення власного Algorand Standard Asset за допомогою SDK та Dappflow
- 6) Pera Wallet. Створення облікового запису

# ТОКЕНІЗАЦІЯ

**Токенізація** - це процес перетворення прав на активи у цифрові токени, які існують на блокчейні. Вона включає створення цифрового представлення реальних або цифрових активів на блокчейні, де кожен токен є унікальним цифровим ідентифікатором активу.

Практично будь-який актив, чи то фізичний, наприклад нерухомість, чи нематеріальний, наприклад акції корпорації, теоретично може бути токенізований.



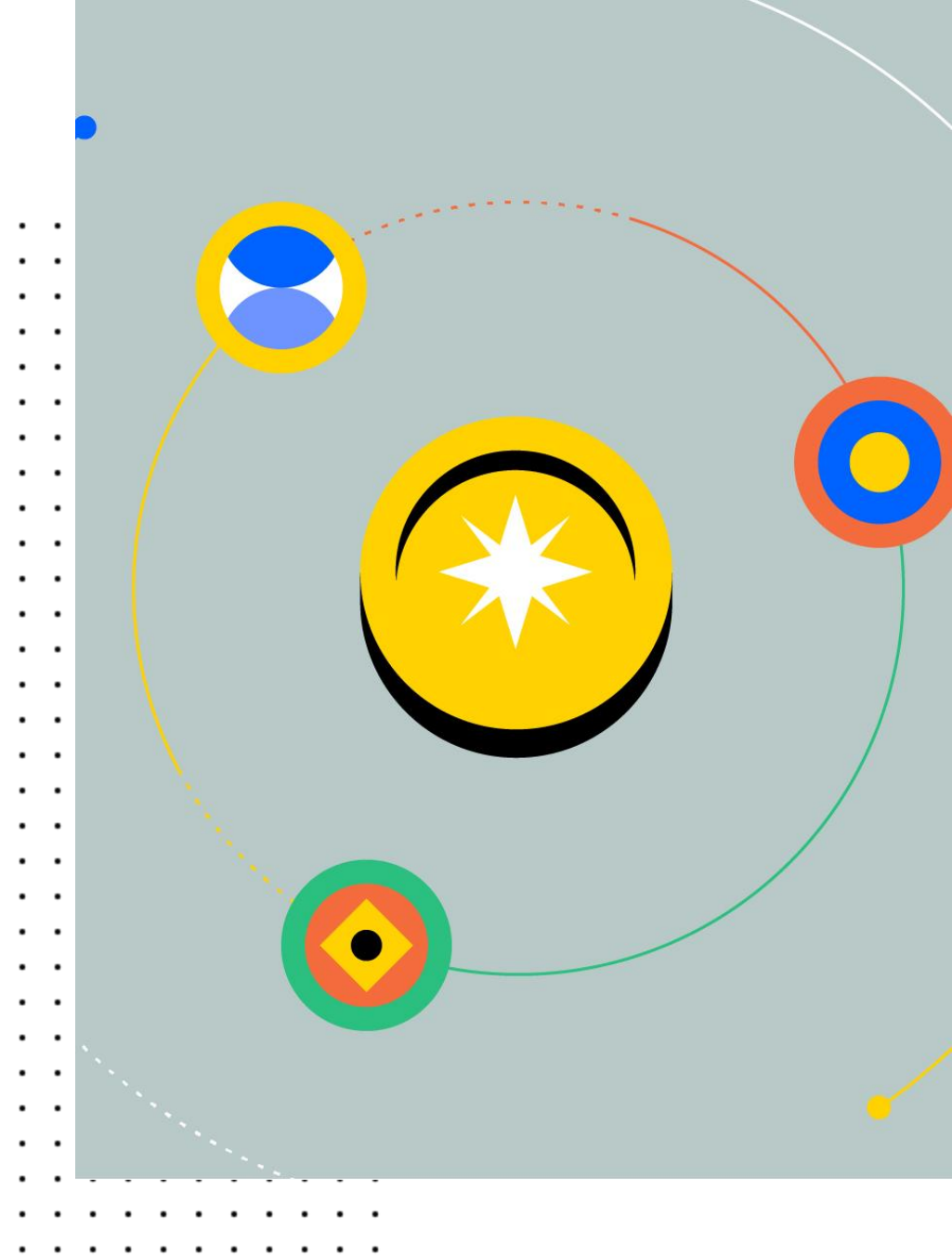
Давньоанглійське слово "tacen", що означає знак або символ, є джерелом терміна "токен". Його часто використовують для опису жетонів спеціального призначення, випущених приватним чином, і які не мають внутрішньої цінності, таких як жетони для ігрових автоматів або транспортних послуг.

# ПОНЯТТЯ "ТОКЕН"

# ТОКЕН

**Токен** - це цифрове представлення права власності або доступу до активу, яке існує на блокчейні. Токени можуть бути використані для представлення різноманітних активів, таких як:

- **Цифрові активи:** криптовалюти, інтелектуальна власність, ігрові предмети.
- **Фізичні активи:** нерухомість, автомобілі, витвори мистецтва.
- **Фінансові інструменти:** акції, облігації.
- **Послуги:** доступ до певних сервісів, таких як хмарне зберігання даних або стримінгові сервіси.





# ОСНОВНІ ЕТАПИ РОЗВИТКУ

- **Поява Bitcoin (2009):** Bitcoin став першим практичним застосуванням блокчейну. Однак, сам термін "токен" тоді ще не був у вжитку. Bitcoin використовувався як цифрова валюта або "монета".
- **Ethereum та смарт-контракти (2015):** Запуск Ethereum, платформи для створення смарт-контрактів, дозволив розробникам створювати власні токени на основі блокчейну Ethereum. Це стало можливим завдяки стандарту ERC-20, що визначив набір правил для випуску tokenів.
- **ICO-бум (2017-2018):** Період інтенсивного використання Initial Coin Offerings (ICO), коли стартапи залучали фінансування шляхом випуску та продажу tokenів. Це призвело до популяризації поняття "токен" і його використання в широкому спектрі проектів.
- **Розвиток DeFi та NFT (2020- до сьогодні):** Зростання децентралізованих фінансів (DeFi) та невзаємозамінних tokenів (NFT) показало, як токени можуть використовуватися для надання доступу до фінансових послуг та цифрових активів, що мають унікальну вартість.



# КЛАСИФІКАЦІЯ ТОКЕНІВ

- **Функціональні токени** (Utility Tokens): надають власникам доступ до певного продукту або послуги в екосистемі блокчейну. Вони зазвичай використовуються для внутрішніх транзакцій в межах платформи, таких як оплата за послуги або доступ до спеціальних функцій.

## Приклад:

- Ether (ETH) на платформі Ethereum, який використовується для оплати газу за виконання смарт-контрактів та стимулювання валідаторів до забезпечення безпеки мережі.
- BNB (Binance Coin) на платформі Binance, який використовується для оплати комісій на біржі.

# КЛАСИФІКАЦІЯ ТОКЕНІВ

- **Інвестиційні токени** (Security Tokens): цифрове представлення традиційних цінних паперів, таких як акції, облігації чи нерухомість, і підпадають під дію тих самих фінансових правил, що й їхні фізичні аналоги. Ці токени отримують свою вартість від базового активу та можуть пропонувати різноманітні переваги, включаючи дивіденди, права на участь у прибутку або права голосу.

## Приклад:

tZERO (TZROP), токенизований цінний папір, що представляє власний капітал у tZERO Group, Inc.

- **Токени, забезпечені активами** (Asset-backed Tokens): забезпечені реальними активами, такими як нерухомість, золото або інші матеріальні цінності. Ці токени дозволяють інвесторам отримувати часткову власність на активи без необхідності їх фізичного зберігання.



# КЛАСИФІКАЦІЯ ТОКЕНІВ

- **Платіжні токени** (Payment Tokens): також називають криптовалютами або цифровими валютами, призначені для використання як засіб обміну, подібно до традиційних валют. Вони використовуються для оплати товарів та послуг і часто створюються з метою забезпечення швидких та безпечних транзакцій.

**Приклад:** Bitcoin (BTC): найбільш відома криптовалюта, яка використовується як засіб обміну.

- **Токени управління** (Governance Tokens): надають власникам права на участь в управлінні блокчейн-платформою або проектом. Власники таких токенів можуть голосувати за зміни в протоколі, нові функції або використання фондів.

**Приклад:** Uniswap (UNI): токен, який надає власникам право голосу в управлінні децентралізованою біржею Uniswap.

# КЛАСИФІКАЦІЯ ТОКЕНІВ

Ще одне застосування токенів – це **стейблкоїни**, які вирішують проблему волатильності криптовалюти.

“Стейблкоїни – це криптовалюти, вартість яких прив'язана до вартості іншої валюти, товару або фінансового інструменту”

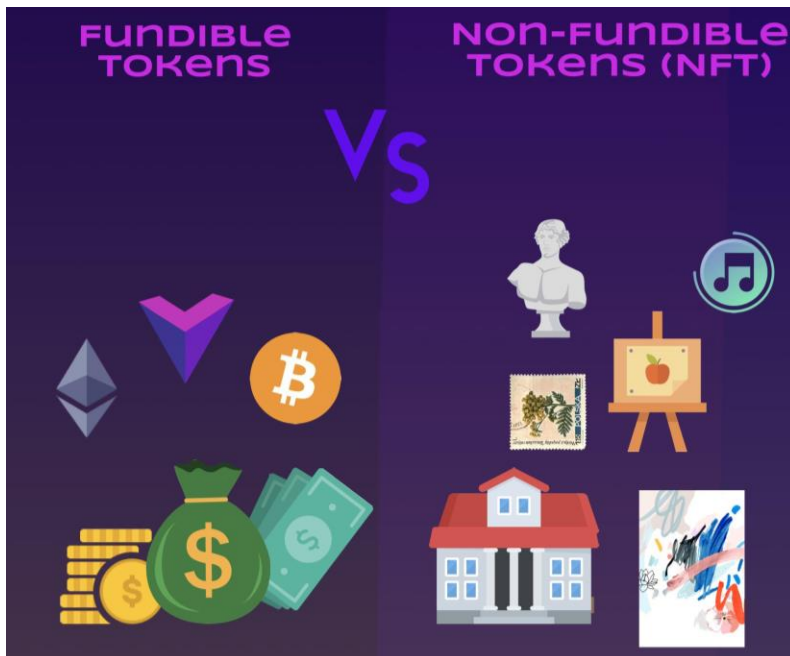
---

- Adam Hayes, Investopedia

Стейблкоїни можна розділити на кілька категорій, залежно від того, чим вони забезпечені. Це: фіатні, товарні, криптозабезпечені та незабезпечені.

**Приклад:** Tether (USDT), USD Coin (USDC), Dai (DAI) від MakerDAO

# NFT TA FT



- **Невзаємозамінні токени** (NFT-Non Fungible Token) – це унікальні криптографічні активи, які **мають власний ідентифікатор** і не можуть бути відтворені. NFT можуть виглядати схожими, але їх можна відрізнити за унікальним ідентифікатором, і вони не є взаємозамінними. NFT можуть представляти цифрові та реальні активи, такі як фотографії, мистецтво, ігрові предмети, квитки тощо.
- **Взаємозамінні токени** (FT - Fungible Token) схожі на NFT, але велика відмінність полягає в тому, що ці активи взаємозамінні та **мають однаковий ідентифікатор**. Користувачі не можуть відрізнити ці активи, які походять із того самого набору, оскільки зображення, метадані та ідентифікатор однакові. FT можуть представляти взаємозамінні активи, такі як внутрішньоігрові зілля, валюта, бали тощо.



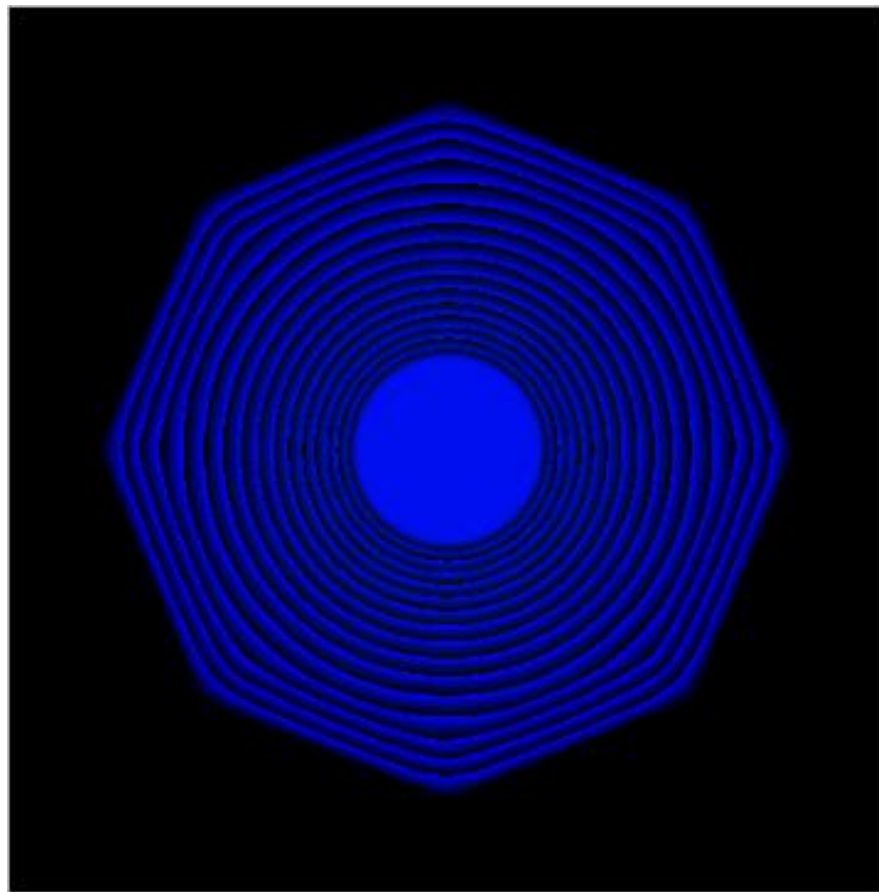
# ВЗАЄМОЗАМІННИЙ VS НЕ ВЗАЄМОЗАМІННИЙ

- **Взаємозамінність** — це економічний термін, що описує взаємозамінність певних товарів. Такі предмети можна легко обмінювати один на одний, тому що їхня цінність не прив'язана до їх унікальності.

Наприклад, доларова банкнота також дорівнює будь-якій іншій доларовій банкноті.

- **Не взаємозамінність** полягає в тому, щоб зробити елементи унікальними або відмінними один від одного. У NFT кожен токен має унікальні властивості і не коштує стільки ж, скільки інші аналогічні токени.

Наприклад, якщо ви візьмете доларову купюру та підпишете її у відомого художника, вона стане унікальною, несхожою на всі інші доларові купюри і, можливо, коштуватиме більше, ніж її номінальна вартість.



Перший незамінний токен – Quantum

# РІЗНИЦЯ МІЖ ТОКЕНАМИ ТА КРИПТОВАЛЮТОЮ

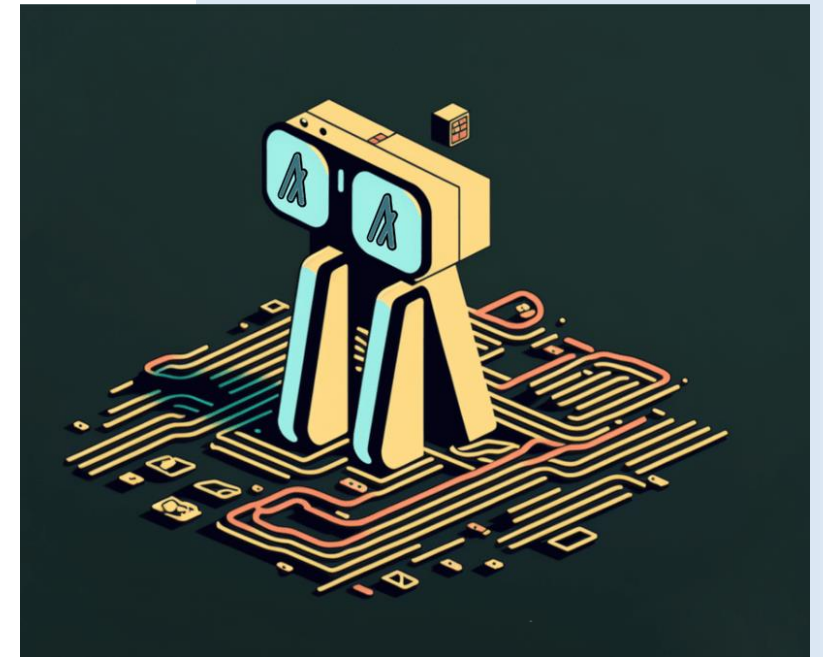
- **Платформа:** криптовалюти є нативним активом певного протоколу блокчейну, тоді як токени створюються платформами, побудованими на основі цих блокчейнів. Наприклад, власним токеном блокчейну Ethereum є ефір (ETH). Але існує безліч інших tokenів, які використовують блокчейн Ethereum, серед іншого, DAI, LINK, COMP та CryptoKitties.
- **Призначення:** криптовалюти переважно використовуються як засіб обміну або зберігання вартості, тоді як токени можуть мати широкий спектр застосувань, включаючи утиліти та представлення активів.
- **Технічна Реалізація:** криптовалюти функціонують як незалежні валюти, тоді як токени функціонують як частина більшої екосистеми, використовуючи смарт-контракти для забезпечення своїх функцій.



# ALGORAND STANDARD ASSETS

У контексті Algorand актив – це цифрове уявлення цінного предмета, який можна передавати та зберігати у блокчейні Algorand. ASA - це цифрові активи в ланцюжку, побудовані на блокчейні Algorand.

Ці активи можуть представляти будь-що: від стейблкоїнів і криптовалют до токенизованих реальних активів, таких як нерухомість чи мистецтво.



# КЛЮЧОВІ ХАРАКТЕРИСТИКИ ASA

- **Універсальність:** ASA дозволяють створювати широкий спектр активів, включаючи:
  - Взаємозамінні токени
  - Ігрові бали
  - Бали лояльності та винагороди
  - Стейблкоїни
  - Токенізовані цінні папери
  - NFT тощо
- **Простота створення:** ASA не вимагають написання смарт-контрактів! ASA можна карбувати та налаштовувати за допомогою простих транзакцій (SDK / CLI).
- **Налаштовуваність:** користувачі можуть налаштовувати різні параметри активів, такі як кількість випуску, можливість переказу та інші атрибути.

# 01

За кожен актив, створений обліковим записом або що належить йому, мінімальний баланс цього облікового запису збільшується на 0,1 Algo (100 000 microAlgo).

# 02

Якщо буде здійснено будь-яку транзакцію, яка порушить вимоги до мінімального балансу, транзакція буде невдалою.

# 03

Перш ніж новий актив може бути переведений на певний рахунок, одержувач повинен погодитись на отримання активу ([opt-in](#)).

# 04

Кількість активів, які може створювати або використовувати один обліковий запис, необмежена.

# НЕЗМІННІ ПАРАМЕТРИ АКТИВІВ ¶

Ці вісім параметрів можна вказати лише під час створення активу:

Параметер	Опис	Обов'язковий
Creator	Вказує обліковий запис, який створює актив.	Так
AssetName	Розширена назва активу. Допомагає ідентифікувати актив для користувачів.	Ні, але рекомендовано
UnitName	Коротка назва одиниці активу. Використовується для позначення одиниці активу (наприклад, "USD", "BTC").	Ні, але рекомендовано
Total	Загальна кількість одиниць активу, що буде випущена.	Так
Decimals	Визначає, скільки десяткових знаків може мати одиниця активу.	Так
DefaultFrozen	Забезпечує контроль над тим, чи будуть нові активи автоматично заморожені до подальшого схвалення.	Так
URL	Вказує на додаткову інформацію або веб-сайт, пов'язаний з активом.	Ні
MetaDataHash	Хеш метаданих активу.	Ні

# ЗМІНЮВАНІ ПАРАМЕТРИ АКТИВІВ ¶

Algorand Standard Assets (ASA) мають кілька параметрів, які **можуть бути змінені** після створення активу. Ці параметри відповідають адресам, що уповноважені виконувати певні функції для активу. Хоча ці адреси необхідно вказувати під час створення активу, вони також можуть бути змінені пізніше.

Параметр	Опис
Manager Address	Єдиний обліковий запис, який може авторизувати транзакції з <u>переналаштування</u> або <u>знищення</u> активу.
Reserve Address	Резервна адреса зберігає невипущені (непереміщені) одиниці активу. Це означає, що всі активи, які ще не були переведені іншим користувачам, зберігаються в резервному обліковому записі.
Freeze Address	Обліковий запис, який має право заморожувати або розморожувати активи для конкретного облікового запису. Якщо параметр порожній, заморожування не допускається.
Clawback Address	Рахунок, який дозволяє переказувати активи від будь-якого і до будь-якого власника активів (якщо вони прийняли (opt-in) цей актив). Clawback Address зазвичай використовується для відкликання активів у користувачів.

# СТВОРЕННЯ АКТИВУ

Створити власний актив  
в блокчейні Algorand можна 3 способами:

- За допомогою Algorand SDK
- За допомогою "goal" CLI
- За допомогою [DappFlow](#)





# СТВОРЕННЯ АКТИВУ

Використання SDK для створення власного ASA

**КРОК 1:** створення транзакції, в якій визначено всі параметри активу.

*Може створити транзакцію: будь-який рахунок із достатнім балансом Algo*

```
const suggestedParams = await algodClient.getTransactionParams().do();
const txn = algosdk.makeAssetCreateTxnWithSuggestedParamsFromObject({
  from: creator.addr,
  suggestedParams,
  defaultFrozen: false,
  unitName: 'rug',
  assetName: 'Really Useful Gift',
  manager: creator.addr,
  reserve: creator.addr,
  freeze: creator.addr,
  clawback: creator.addr,
  assetURL: 'http://path/to/my/asset/details',
  total: 1000,
  decimals: 0,
});
```



# СТВОРЕННЯ АКТИВУ

Використання SDK для створення власного ASA

**КРОК 2:** підписання транзакції та відправлення в мережу

```
const signedTxn = txn.signTxn(creator.privateKey);
await algodClient.sendRawTransaction(signedTxn).do();
const result = await algosdk.waitForConfirmation(
  algodClient,
  txn.txID().toString(),
  3
);

const assetIndex = result['asset-index'];
console.log(`Asset ID created: ${assetIndex}`);
```



# СТВОРЕННЯ NFT

## 01

NFT створюється за допомогою транзакції створення ASA, як і взаємозамінні токени.

## 02

Не потрібно створювати окремий смарт-контракт для NFT.

## 03

Необхідно встановити загальну кількість одиниць, які ви хочете створити для цього активу, на 1, а кількість знаків після коми - на 0. Це гарантує, що ви створите рівно одну одиницю вашого ASA і не зможете розділити нещодавно створений актив.

# СТАНДАРТИ ТОКЕНІВ ДЛЯ NFT НА ALGORAND

ARC-3 (або ARC-0003) є основним стандартом токенів для створення NFT на Algorand. Визначає, як зберігати метадані поза мережею та пов'язує їх з активом за допомогою URL-адреси. Забезпечує сумісність із ринками NFT на Algorand.

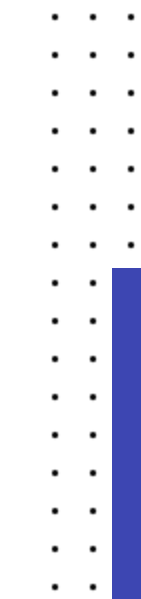
- **Зберігання метаданих:** такі метадані, як ім'я, опис і ознаки, зберігаються поза ланцюгом, як правило, на IPFS або інших децентралізованих або централізованих серверах.
- **Ідентифікація:** токени ARC-3 можна ідентифікувати за URL-адресою активу, яка закінчується на #arc3.
- **Масове карбування:** ARC-3 підтримує масове карбування, корисне для обробки великої кількості метаданих, як-от відео чи інших великих форматів.
- **Підтримка Marketplace:** ARC-3 підтримується всіма торговими майданчиками Algorand NFT.
- **Вартість:** створення та володіння ARC-3 ASA вимагає блокування приблизно 0,1 Algo для кожного творця та власника.



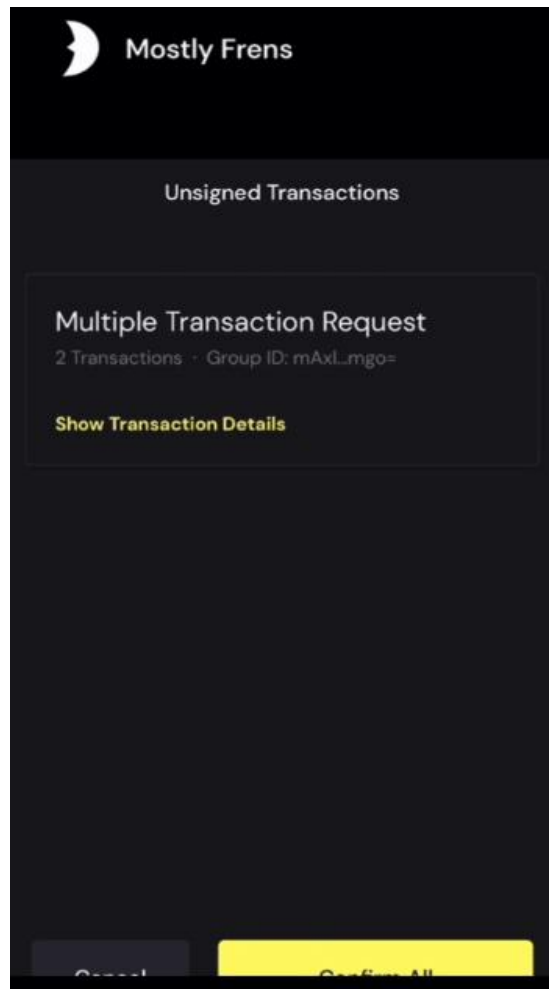
# СТАНДАРТИ ТОКЕНІВ ДЛЯ NFT НА ALGORAND

ARC-19: оновлювані або динамічні NFT

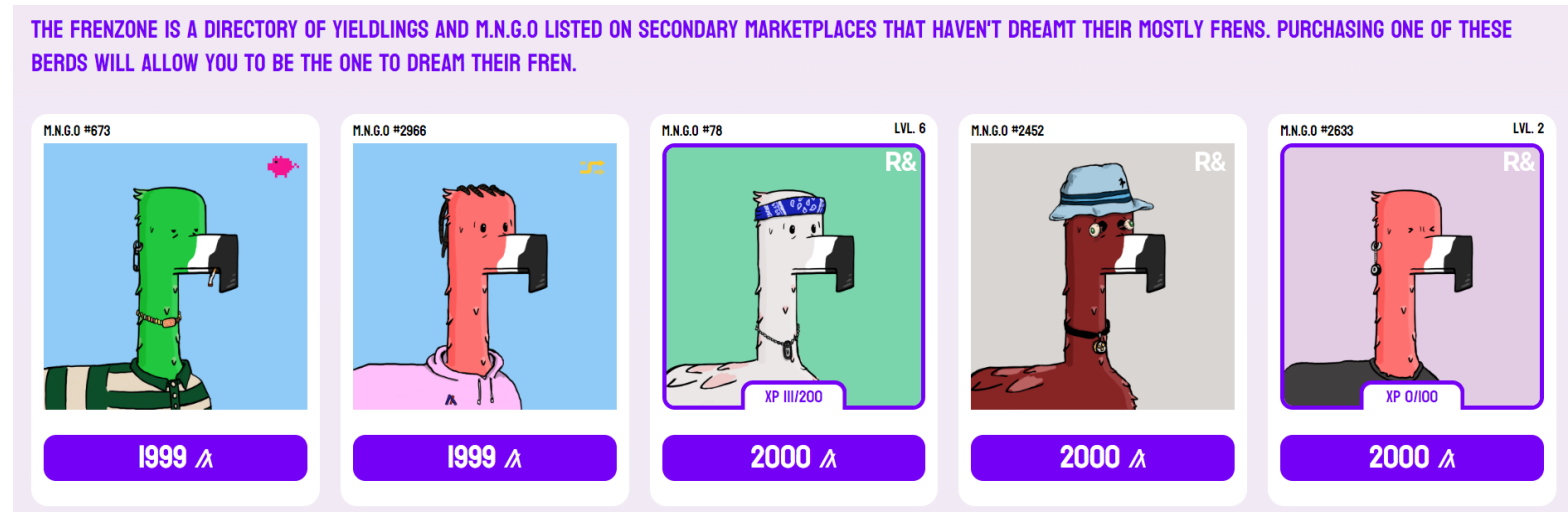
- **Функціональність:** ARC-19 дозволяє оновлювати метадані або цифрові носії, пов'язані з NFT, після створення, забезпечуючи динамічні та інтерактивні NFT.
- **Варіанти використання:** підходить для NFT, які повинні розвиватися з часом або вимагають оновлення вручну.
- **Інструменти:** такі інструменти, як AlgoKit та Evil Tools, можуть полегшити створення та керування NFT ARC-19.



# ПРИКЛАД ARC-19



Mostly Frens є прикладом ARC-19. Це динамічна змінна колекція PFP, де Frens розвиваються, коли їм надсилають токени.



Джерело: [9 Dynamic NFT Collections](#)



# СТАНДАРТИ ТОКЕНІВ ДЛЯ NFT НА ALGORAND

ARC-72 – це стандарт створення NFT за допомогою смарт-контрактів на Algorand.

- **Смарт-контракт NFT:** дає змогу створювати NFT, які використовують можливості смарт-контрактів, надаючи більш складні та налаштовані функції порівняно з традиційними ASA.
- **Непотрібна opt-in транзакція:** ARC-72 усуває необхідність для творців та колекціонерів погоджуватися на отримання токенів ASA (opt-in транзакція).
- **Знання програмування:** для розгортання NFT на основі смарт-контрактів розробникам потрібні навички програмування.
- **Подібність до ERC-721:** працює подібно до стандарту ERC-721 Ethereum.
- **Економічна ефективність:** ARC-72 вимагає блокування 0,1 Algo тільки від творця, від власника - ні. Навпаки, ASA вимагає, щоб мінімальний баланс і автора, і власника NFT збільшувався на 0,1 Algo.

# СТАНДАРТИ ТОКЕНІВ ДЛЯ NFT НА ALGORAND

ARC-18 – це стандарт для виплати роялті на NFT. Його слід використовувати в поєднанні з ARC-3 і ARC-19.

- ARC-18 дозволяє творцям отримувати виплати роялті кожного разу, коли їх NFT продається.
- ARC-18 означає, що творцям не потрібно покладатися на ринки для спрощення виплати роялті.
- Творець вказує ставку роялті під час початкового карбування NFT.



# ЯК ОБРАТИ ПОТРІБНИЙ ARC?

Вимоги	ARC стандарт токена
Я хочу, щоб зображення та метадані завжди залишалися незмінними.	ARC-3
Я хочу запрограмувати свій NFT, щоб він змінювався з часом.	ARC-19
Я хочу оновити зображення та метадані в майбутньому.	ARC-3 та ARC-19
Я хочу, щоб зображення та метадані завжди залишалися незмінними. Я також хочу отримувати виплати роялті кожного разу, коли мої NFT торгуються.	ARC-3 та ARC-18

ARC (Algorand Request for Comments) – це набір стандартів і рекомендацій, призначених для полегшення розробки додатків і активів у блокчейні Algorand. Представлений у вигляді проектного документа, який надає інформацію спільноті Algorand або описує нову функцію Algorand, його процеси та середовище. Необхідно посилатися на ARC, створюючи взаємозамінні та незамінні токени, розробляючи додатки, здійснюючи транзакції та, загалом, усе, що потребує консенсусу екосистеми для роботи.

NFT можуть представляти будь-який унікальний цифровий або фізичний предмет. Це представлення зазвичай виконується шляхом призначення NFT файлів, що представляють його. Однак важливо зазначити, що ці файли не зберігаються в блокчейні. Це мало б економічні та технічні наслідки, оскільки блокчейни не призначені для зберігання великих файлів.

Натомість розробники зберігають файли, що представляють NFT, використовуючи рішення для хмарного зберігання.

# IPFS

InterPlanetary File System, або [IPFS](#), – це розподілена система для завантаження, зберігання та доступу до веб-сайтів, програм, даних і файлів.

Що таке IPFS?

- **Децентралізована файловий система:** IPFS використовує технології P2P (peer-to-peer) для зберігання і передачі файлів між користувачами без необхідності централізованих серверів.
- **Контент-адресація:** Файли в IPFS адресуються за допомогою їх хешів (захешованих значень), що робить їх унікальними і незмінними. Це означає, що кожен файл або його версія мають унікальний ідентифікатор, який базується на його контенті. Таким чином, усі частини вмісту в екосистемі IPFS мають унікальний хеш, який служить ідентифікатором вмісту (CID).





# НАВІЩО ПОТРІБЕН IPFS ПРИ СТВОРЕННІ NFT?

- **Децентралізоване зберігання:** багато NFT посилаються на мультимедійний контент (зображення, відео, аудіо тощо). Зберігання цих файлів на централізованих серверах робить їх вразливими до втрати або зміни. Використання IPFS забезпечує довготривале зберігання даних без централізованої залежності.
- **Незмінність даних:** завдяки контент-адресації, файли на IPFS не можуть бути змінені після їх завантаження. Це забезпечує довіру до автентичності і унікальності NFT.
- **Доступність та стійкість:** Файли на IPFS можуть бути дубльовані і зберігаються на багатьох вузлах мережі, що забезпечує їх високу доступність і стійкість до втрат даних.

Наразі IPFS є основним протоколом для зберігання цифрових медіа, який представляють багато NFT. Коротше кажучи, NFT знаходиться в блокчейні та вказує на медіафайли IPFS у своїх метаданих.



# ЯК ЦЕ ПРАЦЮЄ?

Кожному файлу, що додається до IPFS, надається унікальна адреса на основі хешу вмісту файлу. Ця адреса називається **ідентифікатором вмісту (CID)**, і вона поєднує хеш файлу та унікальний ідентифікатор використовуваного алгоритму хешування в один рядок.

У результаті, якщо ви знову завантажите той же файл, він завжди отримає той же хеш IPFS / CID. Це означає, що навіть якщо третя сторона, яку ви використовували для завантаження своїх токенів, більше не підтримує ваші файли та «видале» їх, ви можете повторно завантажити файли самостійно, і NFT тепер правильно витягуватиме медіафайли.

# ПРИКЛАД

1. Створіть файл метаданих JSON: створіть файл JSON, що містить метадані для вашого NFT. Цей файл має мати певний формат. Ось приклад формату (ARC-3 JSON Metadata file):

```
{
  name: 'NFT::ARC3::IPFS::1@arc3',
  description: 'This is a Scenario1 NFT created with metadata JSON in ARC3 compliance and using
IPFS via Pinata API',
  image: 'ipfs://QmQp7k5JrwQJ2XLfmvEDzmwsTgKaLxzNUNh5eZWmXEAHSe',
  image_integrity: 'sha256-48DEQpj8HBSa...',
  image_mimetype: 'image/png',
  external_url: 'https://github.com/emg110/arc3ipfs',
  properties: {
    file_url: 'arc3-asa',
    file_url_integrity: 'sha256-48DEQpj8HBSa...',
    file_url_mimetype: 'image/png'
  }
}
```

2. Завантажте файл JSON метаданих в IPFS (наприклад, [Pinata](#)). Після завантаження ви отримаєте CID для файлу JSON.
3. Використовуйте CID файлу метаданих JSON як URL-адресу в процесі карбування NFT.

ipfs://YOUR\_METADATA\_CID

\*URL-адреса ресурсу (au) вказує на URI файлу метаданих JSON. У разі використання IPFS слід використовувати лише стандартний URI IPFS (ipfs://...), а не формат шлюзу (https://ipfs.io/ipfs/...). З міркувань стандартів веб-безпеки не можна використовувати http URI. Якщо назва активу не закінчується на @arc3, URL-адреса активу має закінчуватися на #arc3.



# МОДИФІКАЦІЯ АКТИВУ ¶

*Створює транзакцію:*

обліковий запис - Manager  
Address для даного активу

```
const manager = accounts[1];

const configTxn = algosdk.makeAssetConfigTxnWithSuggestedParamsFromObject({
  from: creator.addr,
  manager: manager.addr,
  freeze: manager.addr,
  clawback: manager.addr,
  reserve: undefined,
  suggestedParams,
  assetIndex,
  // don't throw error if freeze, clawback, or manager are empty
  strictEmptyAddressChecking: false,
});

const signedConfigTxn = configTxn.signTxn(creator.privateKey);
await algodClient.sendRawTransaction(signedConfigTxn).do();
const configResult = await algosdk.waitForConfirmation(
  algodClient,
  txn.txID().toString(),
  3
);
console.log(`Result confirmed in round: ${configResult['confirmed-round']}`);
```



# ПОГОДЖЕННЯ НА ОТРИМАННЯ АКТИВУ

Транзакція згоди (opt-in transaction) – це просто передача активів із сумою 0 як до облікового запису, який приймає погоджується на цей актив, так і з нього. Мінімальний баланс для даного облікового запису збільшується на 100 000 microAlgos.

*Створює транзакцію: обліковий запис, який погоджується на отримання активу*

```
// opt-in is simply a 0 amount transfer of the asset to oneself
const optInTxn = algosdk.makeAssetTransferTxnWithSuggestedParamsFromObject({
  from: receiver.addr,
  to: receiver.addr,
  suggestedParams,
  assetIndex,
  amount: 0,
});

const signedOptInTxn = optInTxn.signTxn(receiver.privateKey);
await algodClient.sendRawTransaction(signedOptInTxn).do();
await algosdk.waitForConfirmation(algodClient, optInTxn.txID().toString(), 3);
```



# ПЕРЕДАЧА АКТИВУ ¶

Активи можна передавати між обліковими записами, які погодилися на отримання активів.

*Створює транзакцію:* обліковий запис, на якому зберігається актив, який буде передано.

```
const xferTxn = algosdk.makeAssetTransferTxnWithSuggestedParamsFromObject({  
  from: creator.addr,  
  to: receiver.addr,  
  suggestedParams,  
  assetIndex,  
  amount: 1,  
});  
  
const signedXferTxn = xferTxn.signTxn(creator.privateKey);  
await algodClient.sendRawTransaction(signedXferTxn).do();  
await algosdk.waitForConfirmation(algodClient, xferTxn.txID().toString(), 3);
```



# ОТРИМАННЯ ІНФОРМАЦІЇ ПРО АКТИВ ¶

Отримати інформацію про конфігурацію активу з мережі за допомогою

- SDK
- 'goal'
- blockchain explorer



# ОТРИМАННЯ ІНФОРМАЦІЇ ПРО АКТИВ ¶

Отримати інформацію про актив за допомогою Algorand JS SDK можна наступним чином:

```
const assetInfo = await algodClient.getAssetByID(assetIndex).do();
console.log(`Asset Name: ${assetInfo.params.name}`);
console.log(`Asset Params: ${assetInfo.params}`);
```

Результат:

```
Asset Name: Test Algorand Standart Asset
Asset Params: {
  "clawback": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4054",
  "creator": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4054",
  "decimals": 0,
  "default-frozen": false,
  "freeze": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4054",
  "manager": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4054",
  "name": "Test Algorand Standart Asset",
  "name-b64": "VGVzdCBDbGdvcmFuZCBTdGFuZGFydCBBc3NldA==",
  "reserve": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4054",
  "total": 1000,
  "unit-name": "TASA",
  "unit-name-b64": "VEFTQQ==",
  "url": "http://path/to/my/asset/details",
  "url-b64": "aHR0cDovL3BhdGdvZG8vbXkvYXNzZXQvZGV0YXlscw=="
}
```



# BLOCKCHAIN EXPLORER

Blockchain explorer - це веб-інструмент, який дозволяє користувачам переглядати та здійснювати пошук у загальнодоступному реєстрі блокчейну. Він надає докладну інформацію про блоки, транзакції, облікові записи та активи в блокчейні Algorand.



[AlgoExplorer](#)



[Pera Algorand Explorer](#)

allo'

[Allo.info](#)

# СТВОРЕННЯ АКТИВУ. DARPPFLOW

Darppflow — веб-інтерфейс користувача, який дозволяє візуалізувати облікові записи, транзакції, активи та програми в мережі Algorand, а також надає можливість розгортання та виклику смарт-контрактів.

Darppflow також підтримує створення Algorand Standard Assets

# СТВОРЕННЯ АКТИВУ. DAPFLOW

## КРОК 1: Створити тестовий гаманець

Settings

×

Nodes

Local wallets

+ Create wallet

Address	Balance	Created	Actions

No Rows found

⏪

<

>

⏩

Settings

×

Nodes

Local wallets

+ Create wallet

Delete all

Address	Balance	Created	Actions
<a href="#">FBAA5EKNUDVXEXL2GX5EJ7SBSPB6QP...</a>	0 Algo	0 seconds	<div>🗑️</div>

⏪

<

1

>

⏩

# СТВОРЕННЯ АКТИВУ. DAPPFLOW

КРОК 2: Поповнити тестовий гаманець через Dispenser

## Dispenser

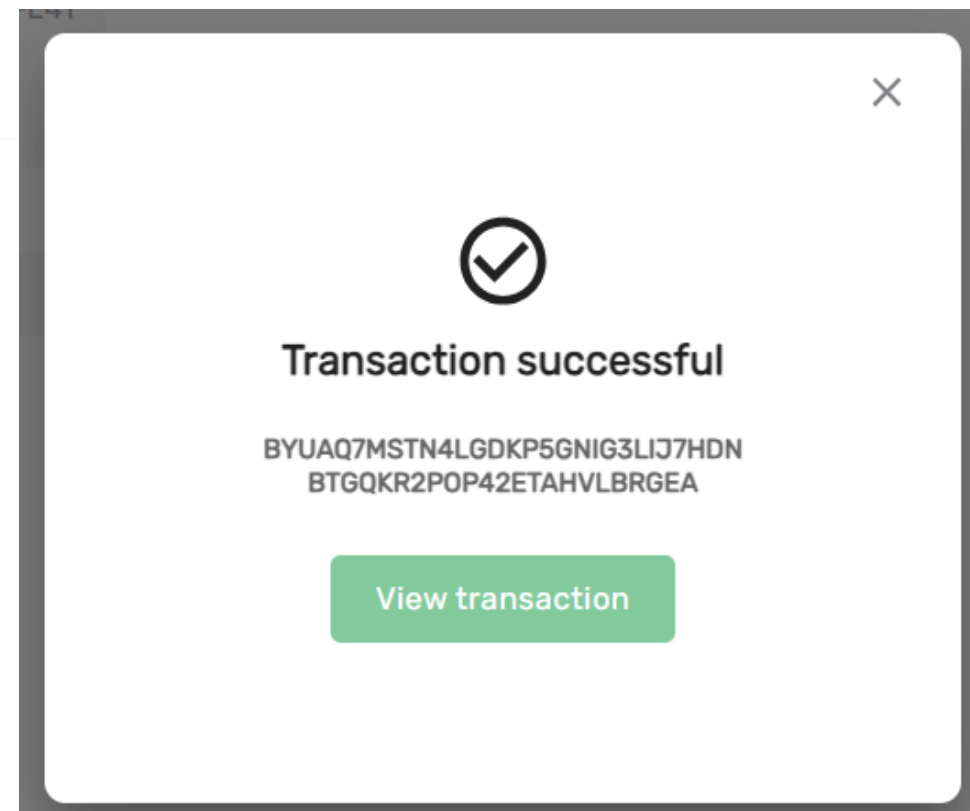
Address

FBAA5EKNUDVXEXL2GX5EJ7SBSPB6QPYSLFE4T  
63LVXKX43MV2LWZUF4QRA

Amount

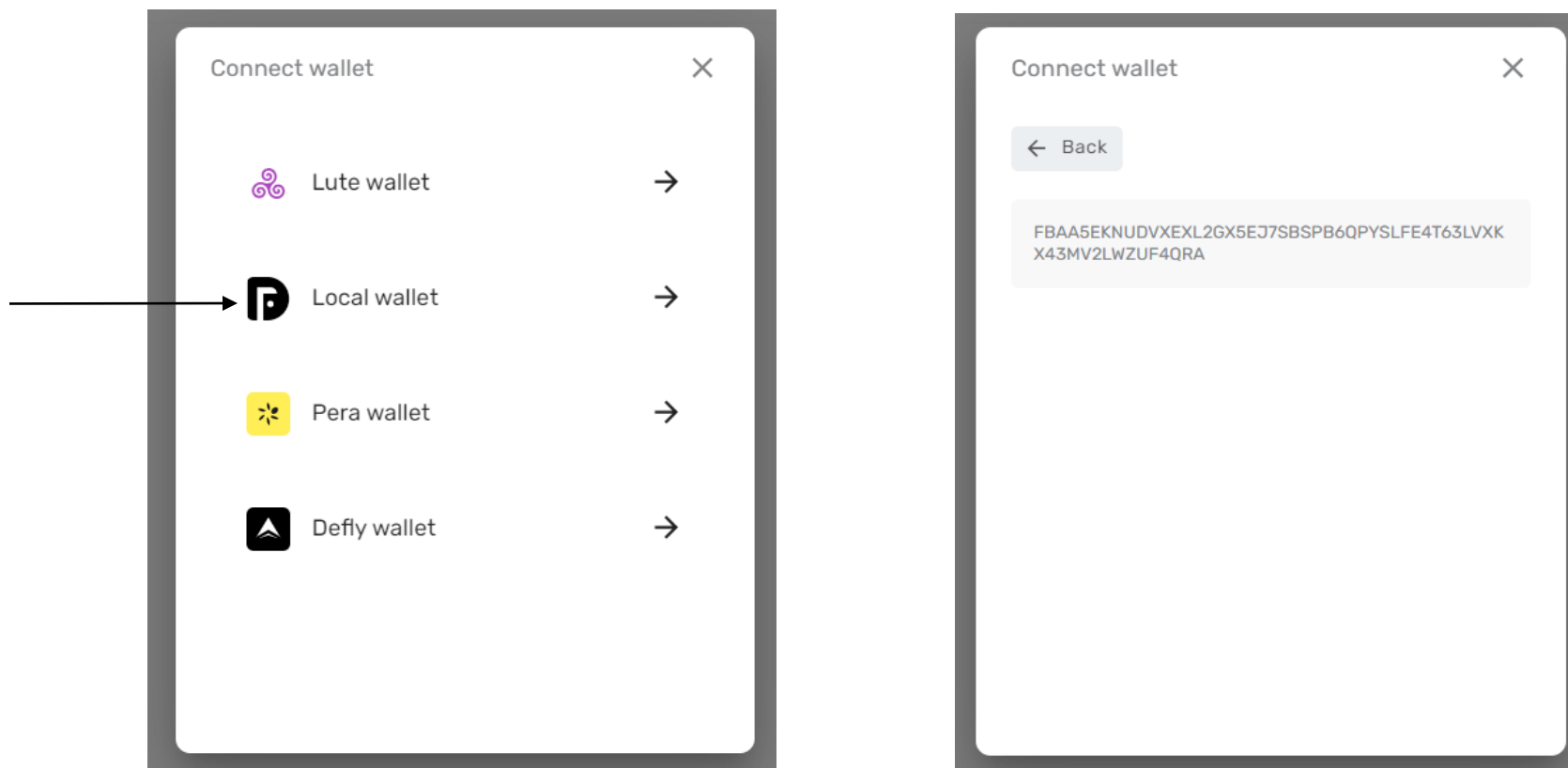
100

Dispense



# СТВОРЕННЯ АКТИВУ. DAPPFLOW

КРОК 3: Перейти в Asset manager та додати створений гаманець



# СТВОРЕННЯ АКТИВУ. DAPPFLOW

## КРОК 4: Визначити параметри нового активу

Create Asset

Name

Test Algorand Standart Asset 2

Unit name

TASA2

Total supply

1000000

Decimals

0

Url

https://www.algorand.com

Metadata hash

32 characters | 32 base64 characters | 64 Hex characters

Manager

FBAA5EKNUDVXEXL2GX5EJ7SBSP  
B6QPYSLFE4T63LVXKX43MV2LWZ  
UF4QRA

Reserve

FBAA5EKNUDVXEXL2GX5EJ7SBSP  
B6QPYSLFE4T63LVXKX43MV2LWZ  
UF4QRA

Freeze

FBAA5EKNUDVXEXL2GX5EJ7SBSP  
B6QPYSLFE4T63LVXKX43MV2LWZ  
UF4QRA

Clawback

FBAA5EKNUDVXEXL2GX5EJ7SBSP  
B6QPYSLFE4T63LVXKX43MV2LWZ  
UF4QRA

Cancel

Create

Asset manager

Created

Opted

Create Asset

Test Algorand Standart Asset 2

ID

1263

Total supply

1,000,000 TASA2

Manager

FBAA5EKNUDVXEXL...

Reserve

FBAA5EKNUDVXEXL...

Freeze

FBAA5EKNUDVXEXL...

Clawback

FBAA5EKNUDVXEXL...

# Інформація про актив, представлена у Dappflow

TA

Asset #1263

Test Algorand Standart Asset 2

✓ Freeze

✓ Clawback

Raw data

Unit name

TASA2

Creator

FBAA5EKNUDVXEXL2GX5E...

Manager

FBAA5EKNUDVXEXL2GX5E...

Reserve

FBAA5EKNUDVXEXL2GX5E...

Freeze

FBAA5EKNUDVXEXL2GX5E...

Clawback

FBAA5EKNUDVXEXL2GX5E...

URL

--Empty--

Default frozen

No

Creation txn

7HVCF3JD3YAR7X5DKFDP...

Creation block

#58529 3 weeks, 18 hours ago

Circulating

Reserve

Total supply

1 Million

Circulating supply

0

Reserve supply

1 Million

Decimals

0

Transactions

Opted accounts

Transaction ID	Block	Age	From	To	Amount

No Rows found



# PERA WALLET





Pera Wallet – це некастодіальний гаманець для покупки, відправки та зберігання криптовалюти в блокчейні Algorand.

Має мобільну та веб версії.

Pera Wallet – це контейнер для ваших облікових записів Algorand

Pera Wallet дозволяє створювати облікові записи Algorand і керувати ними, але облікові записи існують у блокчейні, а не «в Pera»


Pera Wallet – це гаманець без кастодіального права. Будь-які облікові записи, створені за допомогою Pera, залишаються відповідальністю користувачів, і Pera не зможе допомогти, якщо доступ до облікових записів втрачається.

# СТВОРЕННЯ ОБЛІКОВОГО ЗАПИСУ В PERA WALLET

## КРОК 1: створити пароль

Пароль Pera Web Wallet використовується для розблокування гаманця **ТІЛЬКИ** у цьому браузері та на цьому пристрої, де його створили. Він не працюватиме в режимі інкогніто та на іншому комп'ютері.

**Create a passcode**

 Your passcode is used to encrypt your accounts locally, only on this device.  
[Learn more →](#)

Create passcode  
Enter passcode

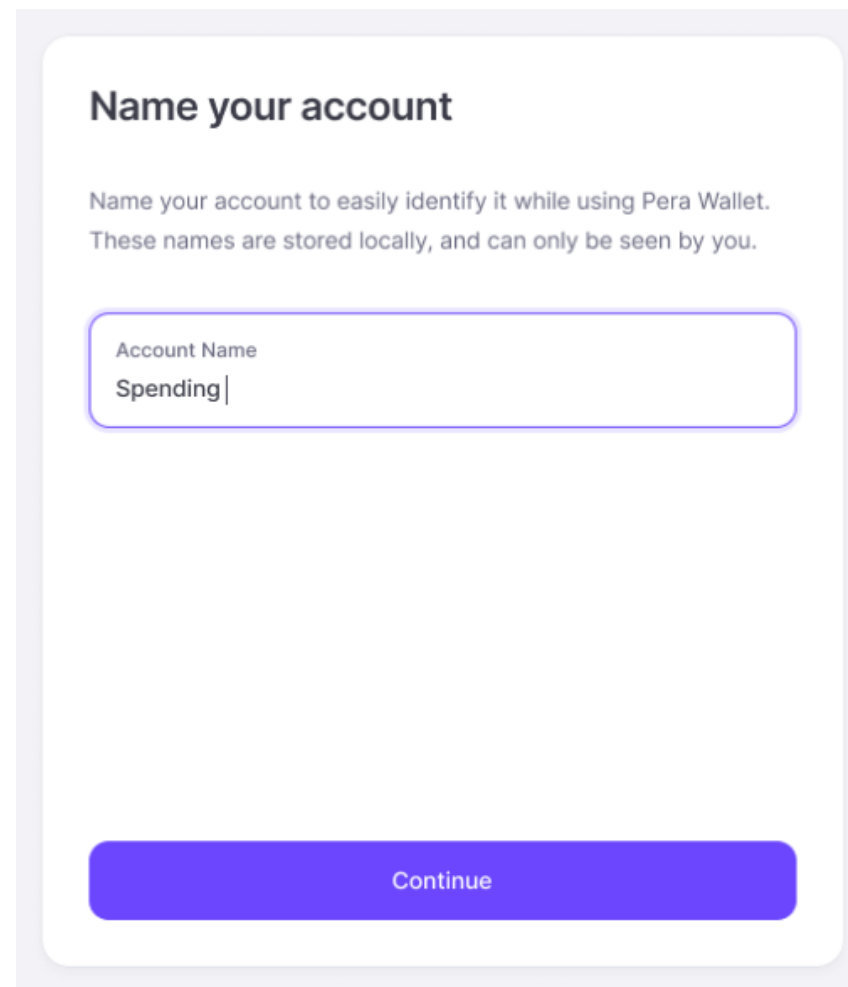
Confirm passcode  
Enter your passcode again

Submit

# СТВОРЕННЯ ОБЛІКОВОГО ЗАПИСУ В PERA WALLET

## КРОК 2: Дати назва акаунту

Ця назва застосовується тільки до вашого веб-гаманця — він ніколи не буде використовуватися в блокчейні або будь-якому іншому гаманці або веб-сайті.



**Name your account**

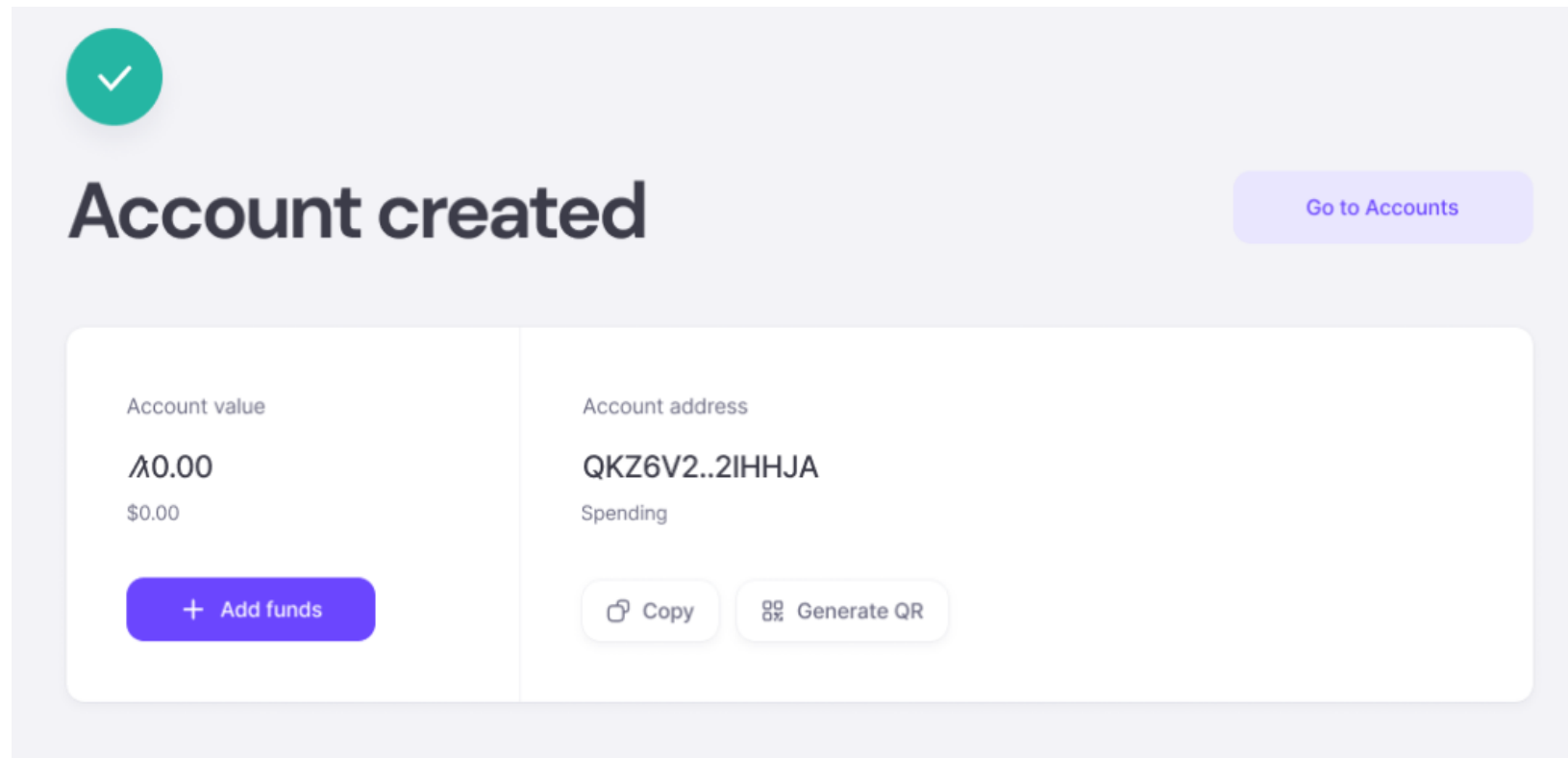
Name your account to easily identify it while using Pera Wallet.  
These names are stored locally, and can only be seen by you.

Account Name  
Spending|

Continue

# СТВОРЕННЯ ОБЛІКОВОГО ЗАПИСУ В PERA WALLET

## КРОК 3: Готовий акаунт



Кожен обліковий запис має власну парольну фразу для відновлення (мнемонічна фраза). Вони повинні зберігатися окремо, причому кожна з них надає доступ лише до одного облікового запису Algorand.

# ДОДАТКОВІ МАТЕРІАЛИ

- [Intro to Assets \(ASA\)](#): вступ до активів у блокчейні Algorand (відео).
- [Building Solutions Using ASAs](#): дізнайтеся про життєвий цикл ASA та подивіться, як створювати взаємозамінні токени та незамінні токени за допомогою SDK, AlgoDesk.io та незамінних доменів (NFD) (відео).
- [The 4 A's of Algorand](#): інформація про облікові записи, активи, атомарні транзакції та додатки в Algorand (відео).
- [How to mint an NFT](#)