

# FLOURISH COIN VIRTUE CAN ONLY FLOURISH AMONG EQUALS.





## Synopsis

- Flourish is a BEP20 protocol which offers unique tokenomics. While beeing deflationary and yield generating with static rewards, it also features an integrated fee for charity.
- Concepts like Staking, Liquidity Mining and Yield Farming have one crucial drawback: impermanent loss (IL). Let's assume a situation when the value of the token changes significantly even though you still receive the farming rewards, their value will be inferior to what you would receive by just holding the tokens in your wallet. The only way to completely avoid IL is to stop farming altogether.
- The Flourish protocol solves these problems increasing the amounts of tokens you have for holding them in your wallet. No staking or farming is needed. This also doesn't force the holder to do additional work.
- While providing all these different solutions Flourish also tackles another issue on it's very own way – an integrated fee for charity which will be distributed towards a 100% transparent and tracked wallet.
- The buy-back feature will force another transaction to buy back the sold tokens. This will help to keep selling streaks short.



#### Tokenomics

Our smart contract collects a 12% fee on every transaction (buy/sell). The ratios are as following:

- Buy-back feature
- The buyback get's split 50/50 50% is sent to our burn address and 50% is added to the liquidity pool.
- 3% Fair redistribution towards all holders.
- 3% Marketing fee Is beeing used to further improve and promote our project.
- 2% Charity fee These fees will be sent towards the charity wallet which will be used for donations.
- & Whale and bot protection explained further on our "Safety measures" page.



#### Static rewards

Static rewards try to solve one common issue:

- The redistribution tax solves several problem:
  - The mechanism of Static Rewards inspires people to hold on to their tokens instead of selling them, because the percentage of their return will be dependent on how big part of the Total Supply they are holding. The amount of rewards received increases over time the longer an investor holds the tokens for.
- Secondly, this algorithm seeks to alleviate the downward pressure on token sales cused by prior users selling their tokens after earning insanely high APYs. Holders are also rewarded for the growing price and transaction volume of Flourish, leading to higher transaction fees, and rewards for distribution

Flourish encourages the community to hold on to their tokens for longer and actively participate in the project, which significantly increases Flourish'es chances of success.



# Statistics / Informations

**SYMBOL: FLRS** 

**TOTAL SUPPLY: 1 Quadrillion** 

DEV. WALLETS: 3% split into 3x 1%

LP LOCKED: 1 Year



## Safety measures

- Liquidity will be locked at launch to ensure safety. The developers won't have any possibilities to withdraw these funds as shown in the audit.
- The contract will be audited before launch to give every holder the possibility to inform themselves about our project in detail. Flourish and the team behind it knows that safety measures and certificates are important any rug-type scenario will be actively avoided.
- Within the smart contract safety measures against whales and bots these restrictions are implemented: Anyone who purchases or holds over 0.5% of the token supply will be subjected to the following:
  - They may not sell more than 0.1% of total supply at a time
  - Once a user gets below the 0.5% line, they will not be affected by these restrictions anymore

Thanks to these measures, Flourish is: unraggable, whale-proof and bot-proof (almost!), not mintable, and much less likely to experience big price drops.



#### Team

The Team behind the project is fully transparent and their names, social media links and pictures can be found on our dedicated website. - Feel free to take a look at it!

You can find and conctact us directly either on our Website or under the following links:

- Flourish Telegram
- Flourish Twitter
- **Flourish Reddit**