



# VYPER



## 19CSE100 – PROBLEM SOLVING AND ALGORITHMIC THINKING PROGRAMMING LANGUAGE SURVEY ASSIGNMENT

**TIFAC-CORE in cyber security**  
**Amrita Vidya Vidyapeetham©**

**Presented by:**  
S Deepak Kumar  
CB.EN.U4CYS22020  
I-B.Tech CYS , 1<sup>st</sup> year

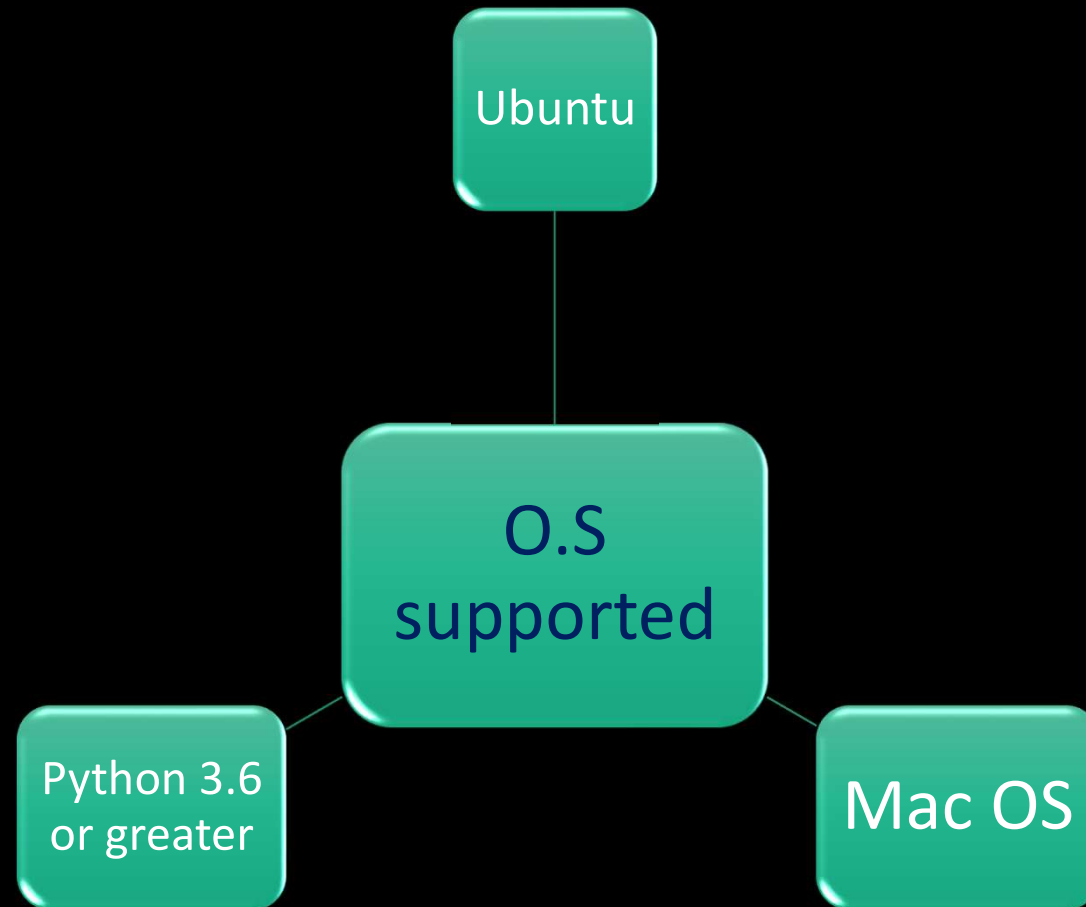
# Who created it and when was it created.?

- The programming language was developed by a team led by Taylor Monahan , a software developer and co-founder of MyEtherWallet. Monahan started the project in 2016 and the vyper team is currently actively developing and maintaining the language
- As per GitHub , the first commit on Vyper was done by Charles cooper. It's Beta version was first released and committed on 22 August 2020 .And the first commit was on March 13 2018 on version 0.0.4

# Paradigm

Vyper is an

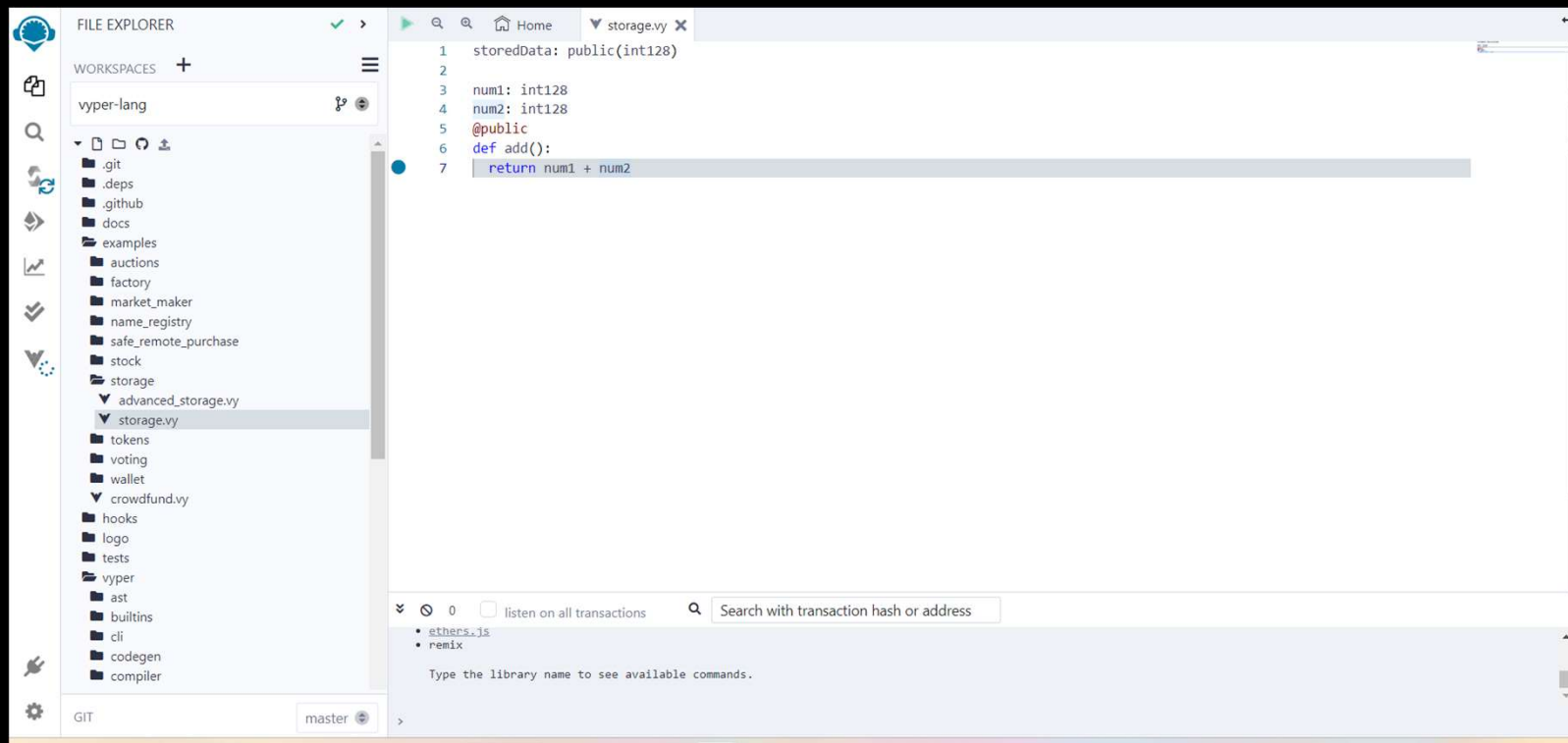
- **Object oriented**
- **Experimental oriented**
- **Statically typed oriented**
- **contract oriented programming language.** Like objects in object oriented programming, each contract contains state variables, functions, and common data types.



# What did Vyper influence.?

- Vyper is a relatively new, pythonic programming language used **to write smart contracts**. Vyper targets Ethereum Virtual Machine (EVM). After solidity vyper is the second most popular language to write smart contracts.

# Adding of two numbers program



The screenshot shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'storage.vy' selected. The code editor displays the following Vyper code:

```
1 storedData: public(int128)
2
3 num1: int128
4 num2: int128
5 @public
6 def add():
7     return num1 + num2
```

The bottom of the interface shows a search bar and a list of libraries: ethers.js and remix. The text 'Type the library name to see available commands.' is displayed below the list.

# Syntax explanation

- I had used Remix –Ethereum IDE to work on Vyper. I plugged in in Vyper and started my code on addition of two numbers

# Declare two variables to hold the input numbers

- num1: int128
- num2: int128

# A function to add the two numbers and return the result

- @public
- def add():
- return num1 + num2

# Real - time application

- Vyper is relatively new pythonic language used to write smart contracts that runs on the Ethereum virtual machine .

The features of Vyper are

- It is strongly typed.
- Has support for signed integers and decimal fixed-point numbers.
- Small and understandable code like Python.
- Has Bounds and Overflow checking.
- Decidability: With Vyper, It is possible to compute a precise upper bound for any function call's gas consumption.
- Limited support for pure functions: Any constant in Vyper cannot change state.



- Vyper omits most of the Object-Oriented Programming paradigms - perhaps requires a new paradigm definition for transactional programming.
- In a study conducted in 2018, where they analyzed nearly one million deployed Ethereum smart contracts, it was found out that many of these contracts had some severe vulnerabilities; the researchers divided these contracts into three categories to better identify and trace vulnerabilities:
  - **Suicidal Contracts** - Smart contracts that arbitrary addresses can kill.
  - **Greedy Contracts** - Sometimes, smart contracts can reach a state where they are unable to release ether, smart contracts in this phase are called Greedy contracts.
  - **Prodigal Contracts** - Smart contracts that can be made to release ether to arbitrary addresses.
- Vyper tries to eliminate this by letting users write secure code and making it difficult for programmers to write misleading or vulnerable code accidentally.



# THANK YOU