

# SimpliPy: A notional machine for learning Python

Venkatesh Choppella

Moida Praneeth Jain, Gnaneswar Kulindala,  
Prabhav Shetty, Anushka Srikanth

International Institute of Information Technology, Hyderabad

# Outline

Motivation: Learning, Comprehension and Composition

Background: The Central Dogma, Machines and Abstraction

Notional Machines and SimpliPy

Workshop Objectives Tasks and Structure

Crew and Credits

Let's Get Started!

# Topic

Motivation: Learning, Comprehension and Composition

Background: The Central Dogma, Machines and Abstraction

Notional Machines and SimpliPy

Workshop Objectives Tasks and Structure

Crew and Credits

Let's Get Started!

# The 3 R's of Learning

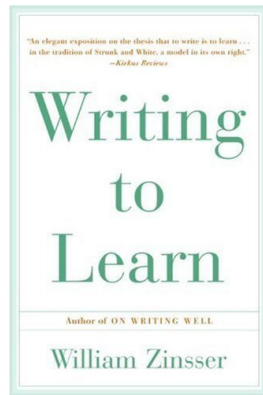
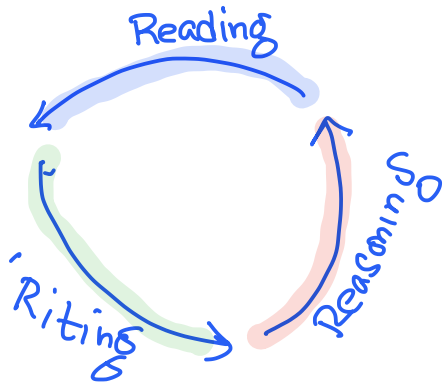


Figure: Zinsser, Writing to Learn, 1988

# What does it mean to learn a programming language?

1. Comprehension: Reading and Reasoning about programs (Nelson et al. 2017)
2. Composition: Writing Programs

# Learning Artefacts

1. Artefacts needed for learning (inputs to learning)
  - teaching material
  - reading material
  - examples
2. Artefacts witnessing learning (byproducts of learning)
  - programs
  - traces
  - diagrams

# Topic

Motivation: Learning, Comprehension and Composition

Background: The Central Dogma, Machines and Abstraction

Notional Machines and SimpliPy

Workshop Objectives Tasks and Structure

Crew and Credits

Let's Get Started!

# The "Central Dogma" of Computer Science

## Program

A program is a sequence of instructions.

## Machine

A machine is a device that runs a program one instruction at a time.

## Programming Language

A Programming Language is a specification of 'acceptable' programs.

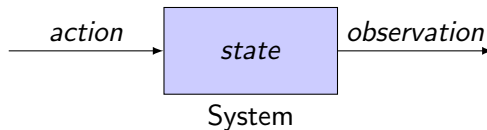
## Computation

A computation is the process of running a program on a machine.



# What is a Machine?

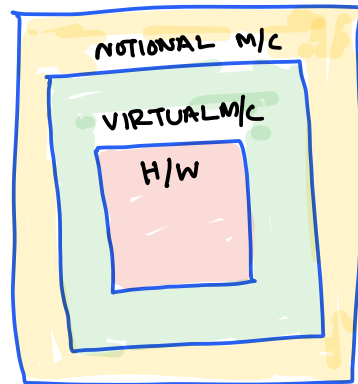
1. **Action:** that which is fed to the machine
2. **State:** moving parts
3. **Output:** that which is observed
4. **Dynamics:** how action changes state
5. **Transition:**  $state \xrightarrow{action} newstate$



# Machines and Abstraction: Context and Purpose

*The essence of abstraction is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context. – John V. Guttag*

- H/W: Physical realisability
- Virtual Machine: Portability
- Notional Machine: Understandability



# Topic

Motivation: Learning, Comprehension and Composition

Background: The Central Dogma, Machines and Abstraction

Notional Machines and SimpliPy

Workshop Objectives Tasks and Structure

Crew and Credits

Let's Get Started!

# What is a Notional Machine?

1. Idealised, conceptual computer ... implied by ... the programming language ...  
(du Boulay et al. 1981)
2. Notional Machine should be part of programming education (Sorva 2013)

# Notional Machines: from Informal to Formal

1. Notional machines have been mostly visual and informal (Fincher et al. ITICSE WG 2020)
2. Can notional machines be specified from formal semantics? (Guzdial et al. Dagstuhl 2019)

# What is SimpliPy?

1. A family of **sublanguages** of Python designed to simplify learning Python
2. A **notional machine** designed to run SimpliPy programs

# SimpliPy family of languages

This workshop:

Advanced workshops (coming soon!)

1. Sequential

4. Exceptional

2. Conditional

5. Classy

3. Iterative

6. Modular

4. Procedural

7. Concurrent

# Why SimpliPy?

1. Precise semantics
2. Line by line interpretation
3. Designed specifically for teaching
4. Encourages drawing artefacts (runs) that demonstrate understanding



# Pedagogical Style: Discrete Maths, Dimensions and Dynamics

1. **Discrete Maths:** Sets, finite partial functions
2. **Dimensions (Types):** Emphasis on shape of quantities.

$$State = Loc \times Env$$

$$Env = Id \rightarrow Val$$

3. **Dynamics:** Capture change

$$(i, e) \xrightarrow{tick} (i', e')$$

# Topic

Motivation: Learning, Comprehension and Composition

Background: The Central Dogma, Machines and Abstraction

Notional Machines and SimpliPy

Workshop Objectives Tasks and Structure

Crew and Credits

Let's Get Started!

## Workshop objectives

1. **Syntax:** Understand the relation between SimpliPy and full Python and the need for the simplification.
2. **Notional Machine:** Understand, in a precise way, the components of the notional machine that runs SimpliPy programs.
3. **Execution Diagrams:** Construct precise execution traces and execution diagrams of how a program runs.
4. **Other languages:** Carry this knowledge to programs in other languages (e.g., Javascript, C, etc.)

## Worksheet exercises

1. **Identify** lexical blocks and variable declarations
2. **Construct** control transfer functions and control flow graph
3. **Trace** the execution of a notional machine given a program
4. **Draw** execution diagrams

## Workshop resources



Figure: <https://github.com/PraneethJain/SimpliPy-Compute-2024-Resources>

# Topic

Motivation: Learning, Comprehension and Composition

Background: The Central Dogma, Machines and Abstraction

Notional Machines and SimpliPy

Workshop Objectives Tasks and Structure

Crew and Credits

Let's Get Started!

## Crew and Credits

- Venkatesh: Conceptualization and Semantics
- Praneeth Jain: Semantics and implementation
- Gnaneswar Kulindala: Control Transfer Functions, Worksheets and Field Studies
- **Prabhav** Shetty: Examples and Quizzes
- **Anushka** Srikanth: Worksheets and Field Studies
- (Nitheesh Chandra): Examples and classroom teaching of SimpliPy

# Topic

Motivation: Learning, Comprehension and Composition

Background: The Central Dogma, Machines and Abstraction

Notional Machines and SimpliPy

Workshop Objectives Tasks and Structure

Crew and Credits

Let's Get Started!