



6.100L | Fall 2022 | Undergraduate

# Introduction To CS And Programming Using Python

Menu

More Info

## Materials by Lecture

- [Lecture 1: Introduction](#)
- [Lecture 2: Strings, Input/Output, Branching](#)
- [Lecture 3: Iteration](#)
- [Lecture 4: Loops over Strings, Guess-and-Check, Binary](#)
- [Lecture 5: Floats and Approximation Methods](#)
- [Lecture 6: Bisection Search](#)
- [Lecture 7: Decomposition, Abstraction, Functions](#)
- [Lecture 8: Functions as Objects](#)
- [Lecture 9: Lambda Functions, Tuples, and Lists](#)
- [Lecture 10: Lists, Mutability](#)
- [Lecture 11: Aliasing, Cloning](#)
- [Lecture 12: List Comprehension, Functions as Objects, Testing, Debugging](#)
- [Lecture 13: Exceptions, Assertions](#)
- [Lecture 14: Dictionaries](#)
- [Lecture 15: Recursion](#)
- [Lecture 16: Recursion on Non-Numerics](#)
- [Lecture 17: Python Classes](#)
- [Lecture 18: More Python Class Methods](#)
- [Lecture 19: Inheritance](#)
- [Lecture 20: Fitness Tracker Object-Oriented Programming Example](#)
- [Lecture 21: Timing Programs, Counting Operations](#)
- [Lecture 22: Big Oh and Theta](#)
- [Lecture 23: Complexity Classes Examples](#)
- [Lecture 24: Sorting Algorithms](#)
- [Lecture 25: Plotting](#)
- [Lecture 26: List Access, Hashing, Simulations, and Wrap-Up](#)



Over 2,500 courses & materials  
Freely sharing knowledge with learners and educators around the world. [Learn more](#)

- [Accessibility](#)
- [Creative Commons License](#)
- [Terms and Conditions](#)

Feedback

Proud member of: 



© 2001–2024 Massachusetts Institute of Technology