

Outline for September 29, 2025

Reading: §1, 2**Due:** Homework 1, due October 15, 2025

1. About the class
 - (a) Instructor
 - (b) Class web site, handouts
 - (c) Canvas and submitting homework
 - (d) Homework, grading, and extra credit
2. Python
 - (a) What is Python?
 - (b) Why Python for this class?
 - (c) Compilers, assemblers, interpreters
3. Python, files and shells
 - (a) Python: programming *language* that you use to tell the computer what to do
 - (b) Shell: what you can type Python statements directly into, to see what they do
 - (c) IDLE: the program that *interprets* Python statements (executes the Python program)
 - (d) File: type Python statements into this, and then have IDLE execute those statements by running the program in the file
 - (e) Jupyter, Anaconda, and Spyder
4. First program: hello, world [*hello_world.py*]
5. How to write a program; example is making change [*make_change.py*]
 - (a) Goal and general algorithm idea
 - (b) Representing data and basic program structure
 - (c) Translating this into a programming-like language
 - (d) Translating that into Python
 - (e) Defensive programming and debugging
6. Assignment [*swap.py*]
 - (a) Simple assignment: *variable = expression*
 - (b) Simultaneous assignment: *variableA, variableB = expressionA, expressionB*
7. Exception `NameError` — local or global name is not found
8. Example
 - (a) Compute the hypotenuse of a right triangle [*hypotnoex.py, hypotnoex1.py*]
 - (b) Handle input problem [*hypotex.py*]
9. Exception `SyntaxError` — Python parser encountered a malformed statement
10. Exception `ZeroDivisionError` — attempt to divide (or take the remainder of) something by 0 [*divby0.py, divby0ex1.py*]
11. The difference between strings and integers [*twoplustwo.py*]
 - (a) Difference between `'2 + 2'`, `2 + 2`
12. Exception `TypeError` — operation or function applied to operand of wrong type
 - (a) Typing letters when a number is expected [*divby0ex2*]

13. Type converter functions `int`, `float`, `str`
14. String operators
 - (a) Operator `+`: string concatenation
 - (b) Operator `*`: repetition
15. Input [*make_change.py*]
 - (a) `input(prompt)` prints prompt, waits for user
 - (b) When user hits enter, it returns what was typed as a string
16. Basic output [*printend.py*]
 - (a) `print(a, b, ...)`
 - (b) `end='XXX'`