

SUMMARY

Note: Since I have to finish examples that cover the topics in each chapter, and since I may not have time — especially if I cover programs — in a 50 minute lecture, the videos can take longer than an hour. Please just view them over 3 or 4 days, say with 20 mins a day. Unless I rush through examples, they do tend to take time to explain, especially when Python is new to you.

At a certain point, when you know how to write basic programs, your learning rate and (hopefully) also your interest in experimenting climb fast. Things get easier. But the programs get larger. Your goal then is to MINIMIZE errors. The best way? DO INCREMENTAL CODING.

Write a couple of lines of code, even very small functions. Test by running the code. When you are sure that part is correct, add the next couple of lines. The idea is to go from CORRECT PIECE to LARGER CORRECT PIECE. This takes more time but your code will be correct, assuming your algorithm is correct. You want to AVOID bugs (logic errors).

You see, when you have a logic error you have now created a new problem different from the original problem you wanted to solve, and you waste time trying to track it down. Incremental coding goes step by step, and takes more time, but you avoid the infinite time you can end up spending with logic errors.

What we covered so far:

Variables, assignments

Data types

Type conversion

Numbers, operators (+, - etc), expressions, limitations of arithmetic, representation

Boolean expressions, conditionals

Chap 7:

For now skip sections 7.1.3 (conditional program execution) and 7.4 (exception handling)

Simple decisions (if-, if-else)

Conditions (the expression in the if-)

Two-way decisions (if-else)

Multiway decisions (if-elif-elif-elif-else)

Decision Trees (e.g., find max of 3 numbers).

The idea of going from a case of 2 to a case of 3,
and then to a case of n .

By now you should be able to write simple programs to do all kinds of things based on conditions.