

6.189_summary

readings by session

readings are from [*how to think like a computer scientist*](#) unless otherwise noted.

LEC#	READINGS	TOPIC	knowledge
1	Chapter1	The way of the program	<ul style="list-style-type: none">• problem solving: The process of formulating a problem, find a solution, and expressing the solution.• interpret: To excute a program in a high-level language by translating it one line at a time• compile: To translate a program written in a high-level language into a low-level language all at once, in preparation for later execution• source code: A program in a high-level language before being compiled• object code/executable: The output of the compiler after it translates the program• script: A program stored in a file(usually one that will be interpreted).
	CHAPTER2	Variables, expressions and statements	<ul style="list-style-type: none">• operator: A special symbol that represents a simple computation like addition, multiplication, or string concatenation.
	section 4.12	Keyboard input	<ul style="list-style-type: none">• input = input() #if the user types a character that is not a digit, the program crashes• input = raw_input() #To avoid this kind of error, it is generally a good idea to use raw input to get a string and then use conversion functions to convert to other types.
2	SECTIONS 4.1	The modulus operator	<ul style="list-style-type: none">• 7 % 3 #should output 1• you can extract the right-most digit or digits from a number. For example, x % 10 yields the right-most digit of x (in base 10). Similarly x % 100 yields the last two digits.
	SECTIONS 4.2	Boolean expressions	
	SECTIONS 4.4-4.7	<ul style="list-style-type: none">• Conditional execution• Alternative execution• Chained conditionals• Nested conditionals	
	SECTIONS 6.1-6.2	Multiple assignment The while statement	<ul style="list-style-type: none">• while<ol style="list-style-type: none">1. evaluate the condition, yielding 0 or 12. if the condition is false(0), exit the while statement and continue execution at the next statement.3. if the condition is true(1), execute each of the statements in the body and then go back to step 1.
3	CHAPTER 3	Functions	
	APPENDIX A	Debugging	

	SECTIONS 6.5-6.9	<ul style="list-style-type: none"> • Encapsulation(封装) and generalization • More encapsulation • Local variables • More generalization • Functions 	<ul style="list-style-type: none"> • development plan: We develop code by writing lines of code outside any function, or typing them in to the interpreter. When we get the code working, we extract it and wrap it up in a function.
	SECTIONS 2.1-2.3 FROM THE Introduction to Computation and Programming Using Python(2nd)	<ul style="list-style-type: none"> • The basic elements of Python • Branching Programs • Strings and Input 	<ul style="list-style-type: none"> • A python program, sometimes called a script, is a sequence of definitions and commands <ul style="list-style-type: none"> • command: often called a statement, instructs the interpreter to do something • Objects: are the core things that Python programs manipulate. Every object has a type that defines the kinds of things that programs can do with that object. • Types: are either scalar or non-salar <ul style="list-style-type: none"> • four types of scalar objects: int, float, bool, None • Objects and operators can be combined to form expressions, each of which evaluates to an object of some type. We will refer to this as the value of the expression. • The operator + is said to be overloaded: It has different meanings depending upon the types of the bojects to which it is applied • Strings are one of several sequence types in python. They share the following operations with all sequence types: <ul style="list-style-type: none"> • len('abc') • Indexing: can be used to extrat individual characters from a string. • Slicing: is used to extract substrings of arbirary length
4	CHAPTERS7	Strings	strings_note.py
	CHAPTERS8	Lists	lists_note.py 轻松学会python列表解析式
	CHAPTER2 from "Lists" on, from the 6.01COURSE NOTES	• 2.5 Python Style**	lists_note.py
5			

LEC #	READINGS
1	Chapters 1 and 2, section 4.12
2	Sections 4.1, 4.2, 4.4-4.7, and 6.1-6.2
3	Chapter 3, appendix A, sections 6.5-6.9 Sections 2.1, 2.2, and 2.3 (up to the heading "Lists") from the 6.01 course notes
4	Chapters 7 and 8 Chapter 2 from "Lists" on, from the 6.01 course notes More on list comprehensions is available here and on Wikipedia .
5	Chapters 9 and 10
6	Chapters 12-14 (read chapter 14 well; get through at least section 14.6) This tutorial on classes may also be helpful.
7	Finish chapters 12-14; read chapter 16
8	Wikipedia: Conway's Game of Life (Project 2)
9-11	For the final week, make sure you know and understand chapters 12-14 and 16. Chapters 11, 15, 18, 19, and 20 are very interesting, but not covered in this course; take 6.01 , 6.042 , and 6.006 to learn more about programming and data structures.