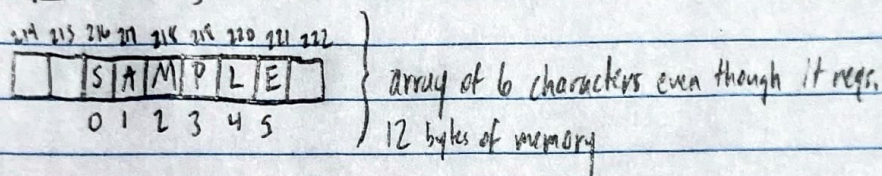# Chapter 5: Array Based Sequences

Low-level Arrays

→ typical unit: byte, which is 8 bits

→ to keep track of information and its storage location, computer uses an abstract called a <u>memory address</u>

    → each byte associated w/ a random # (unique) that serves as its address

→ a group of related variables can be stored in a contiguous portion of the computer memory; this is what we consider an <u>array</u>

→ example: text string stored as individual characters

214	215	216	217	218	219	220	221	222
		S	A	M	P	L	E	
		0	1	2	3	4	5	

} array of 6 characters even though it reqs. 12 bytes of memory

Referential Analysis

→ to represent a list w/ an array, Python adheres to the requirement that each cell of the array must use the same # of bytes

→ Python uses an internal storage mechanism of an array of object <u>references</u>

    → at lowest level, a consecutive sequence of memory addresses at which the elements reside are stored

→ same semantics are demonstrated when making a new list as a copy of an existing one, with a syntax such as backup = list(primes).

→ produces a new list that is a <u>shallow</u> copy, in that it references the same elements as the original list

→ If the contents of a list are mutable, a <u>deep copy</u>, meaning a list w/ new elements