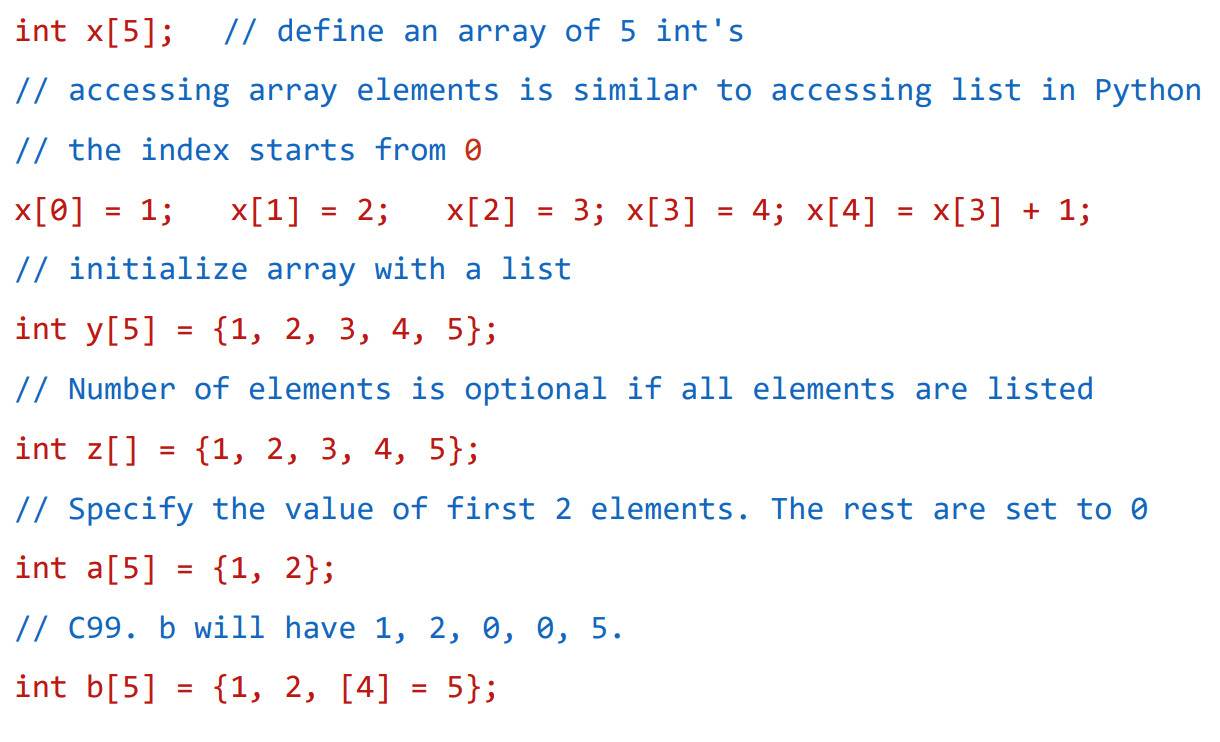
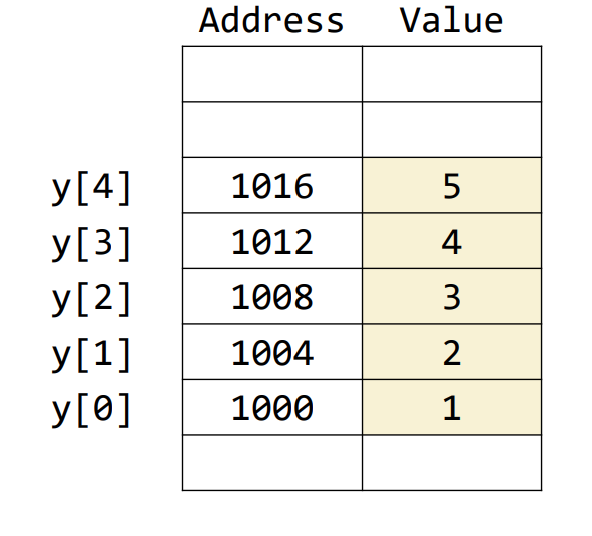
Arrays and Pointer Basics



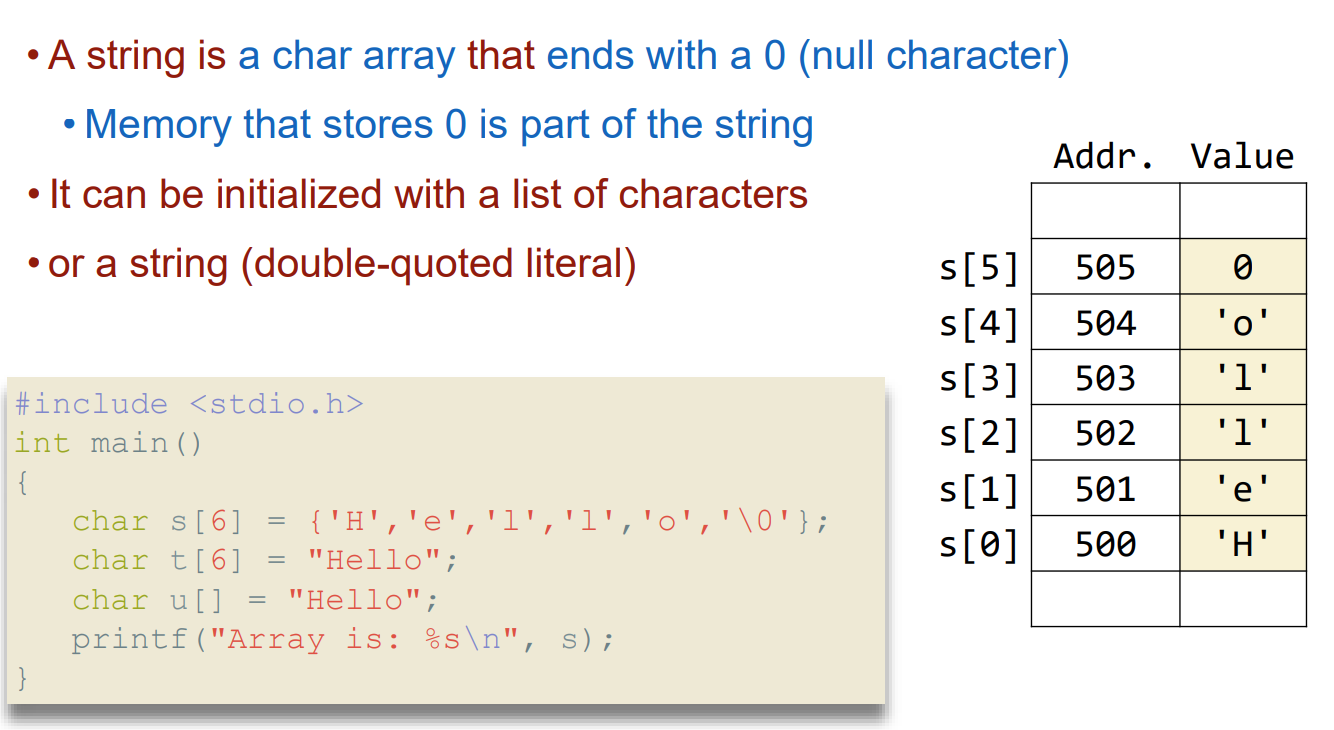
Array in memory

Think about how array elements are stored in memory

Index starts from 0, the last one is 4 = (5-1)



String Initialization



Arrays as Automatic Variables

You can declare arrays inside any function or block

Destroyed when exiting function or block

Variable length arrays he size of your array can depend on function arguments or other know values

Array Assignment

You cannot assign a while array at once to another array

Even when the types match

Arrays and Functions

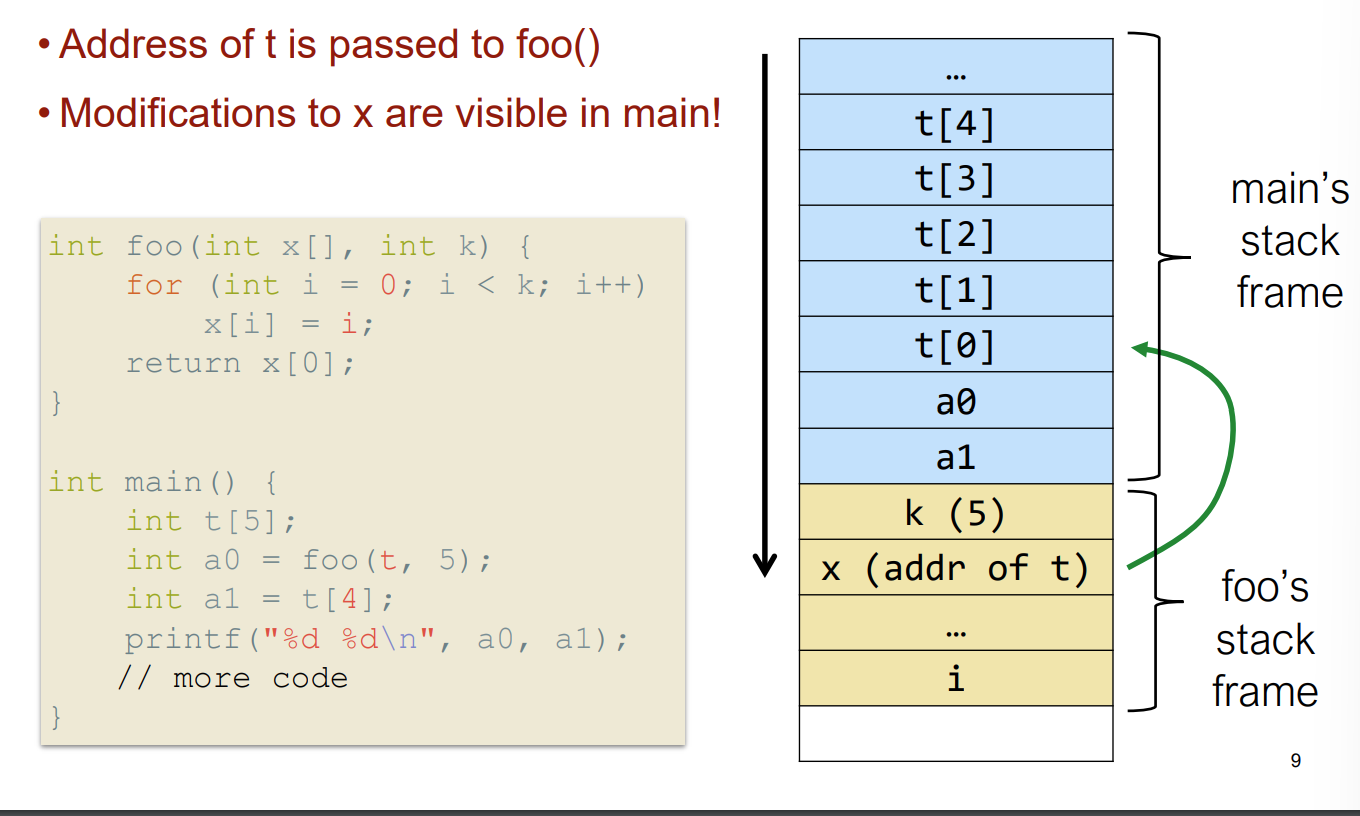
Arrays can be passed to functions with on big caveat

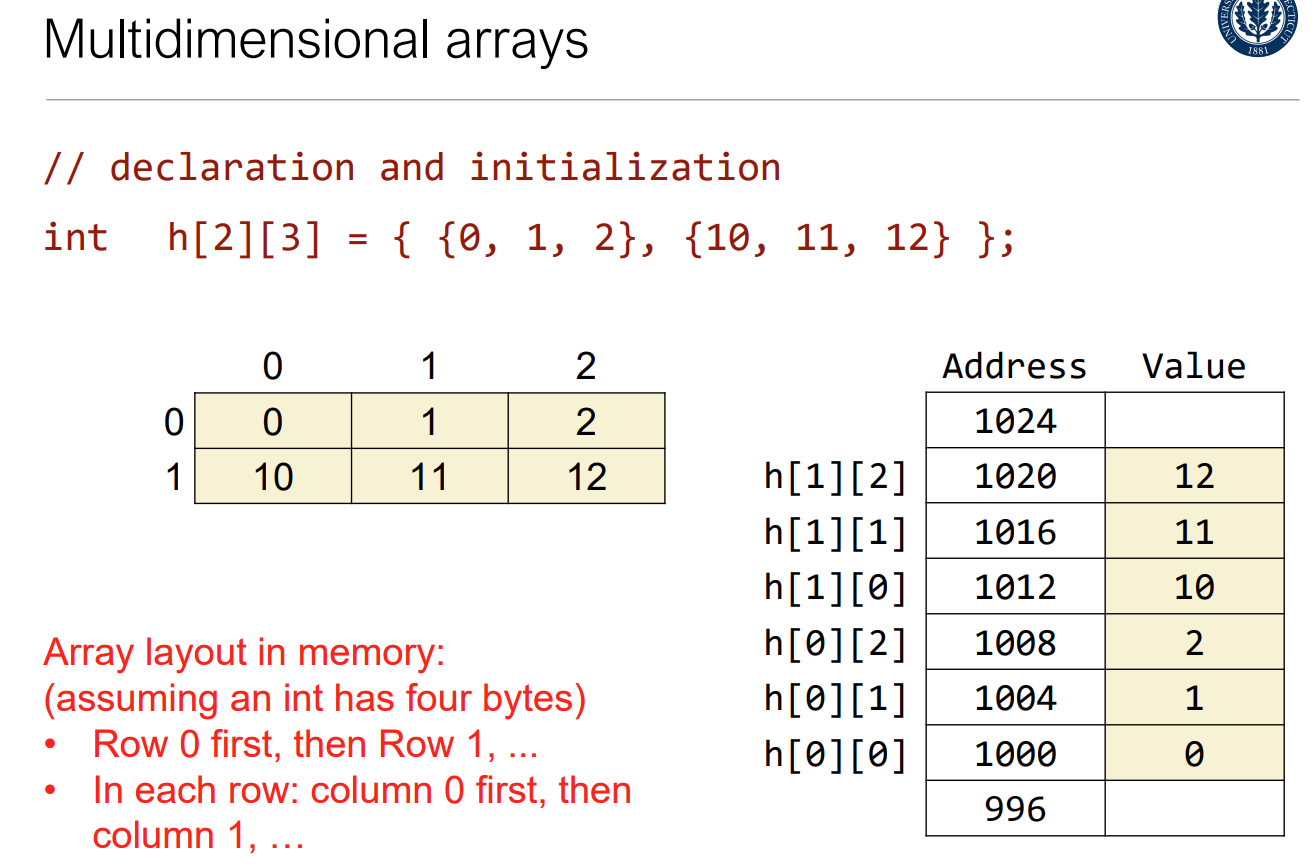
Calling convention in C except arrays

Arrays are always passed by reference

Functions cannot return arrays

Array argument example





Pointers

A value denoting the address of a memory cell

Variables and Memory

The memory is an array of bytes

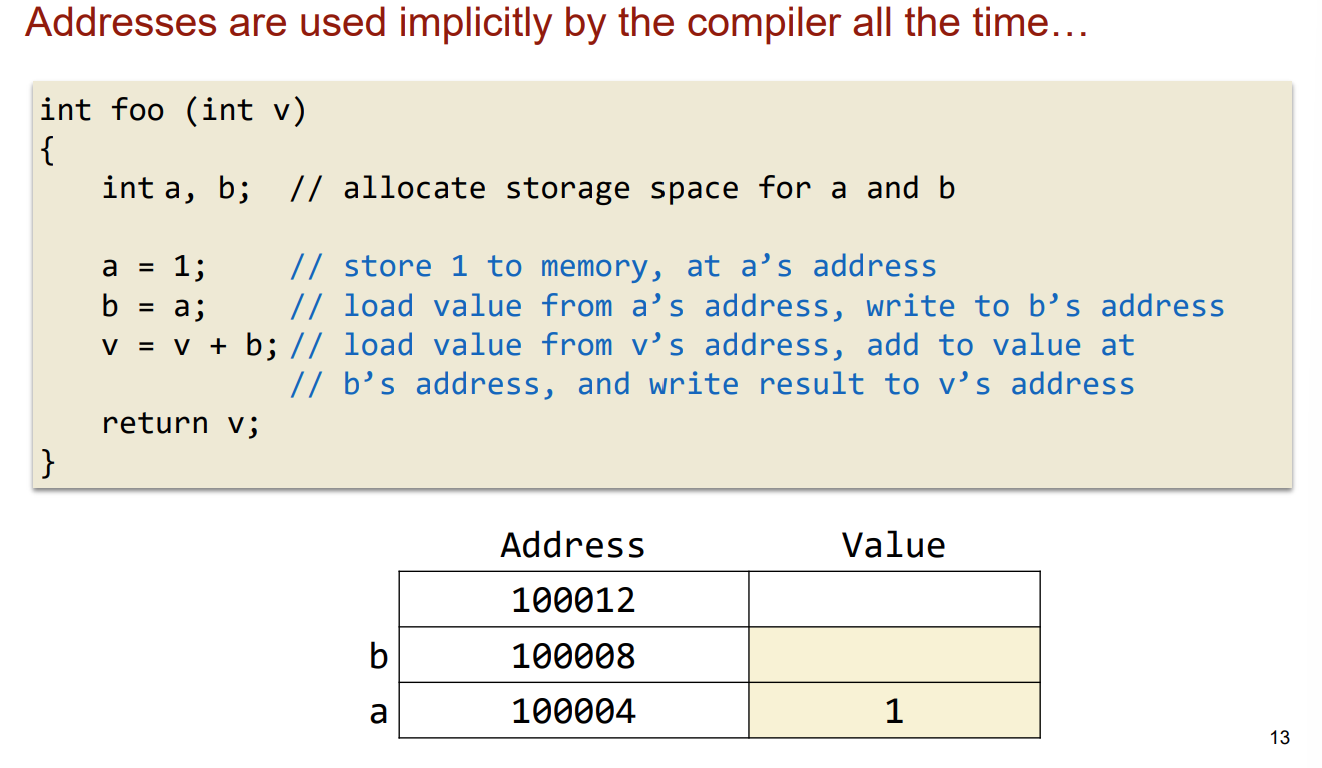
Every byte in memory is numbered: the address!

Every variable is kept in memory and is associated with two numbers:

The address

The value stored at that address

Implicit Address Use



Explicit Use: Points!

A pointer is a variable that holds an address of something

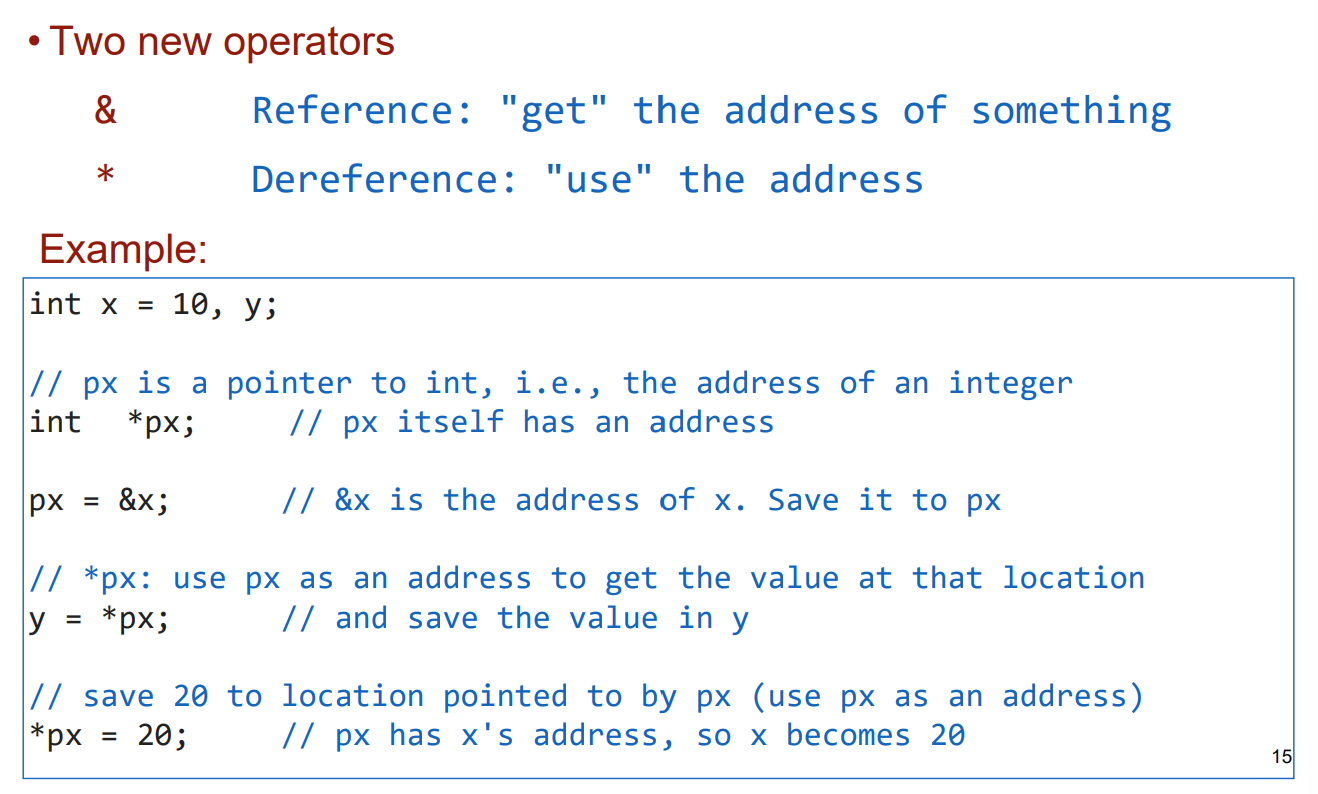
Declaration

Int \*p;

The value of p is an address of an int

P itself has an address

Referencing and dereferencing



Picture it

