

malloc | Coursera

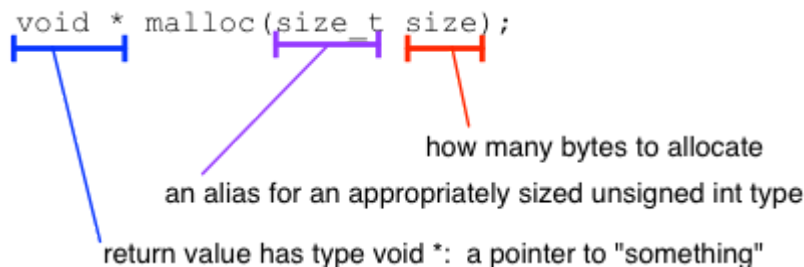
 coursera.org/learn/interacting-system-managing-memory/supplement/15v3H/malloc

malloc

malloc

The most basic function for dynamic memory allocation is called **malloc**(which performs **memory allocation**). Calling this function is how you allocate memory dynamically. The malloc function, shown in the figure below takes one argument telling it how much memory is needed and it returns a pointer to that allocated memory in the form of a *void **. Many beginning programmers are intimidated by the concept of a *void **, but you should not be! Recall that a *void ** just means a pointer, but we do not know what type of thing it points to. If malloc instead returned something more specific (for example, an *int **), we would need a new version of malloc for every data type. This would be both unwieldy and (in the context of user-defined data types) impossible. Just remember that you can assign a *void ** to any other pointer type—so just assign the return result of malloc to whatever pointer you want to initialize.

```
void * malloc(size_t size);
```



The diagram shows the function signature `void * malloc(size_t size);` with three colored brackets and arrows pointing to explanatory text:

- A blue bracket under `void *` points to the text: "return value has type void *: a pointer to 'something'"
- A purple bracket under `size_t` points to the text: "an alias for an appropriately sized unsigned int type"
- A red bracket under `size` points to the text: "how many bytes to allocate"



Completed