

void* is a Pointer to Nothing

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

But what is type void*?

Type **void***

- (a pointer to nothing)
- is **auto-converted to/from any pointer type**
- without generating a warning.

Do not use it for pointer arithmetic.

Do not dereference it.

Pitfall: Assuming 0 Bits in New Memory

What's in the new chunk of memory returned from malloc?

Bits!

They may be 0 bits.

Unfortunately,

- they're likely to be 0 bits
- if you do a little bit of testing.

In general, however, they are bits.

calloc Allocates a Zeroed Chunk of Memory

If you want 0 bits, use

```
void* calloc (size_t num_elts,
             size_t elt_size);
```

The **number of bytes needed** is the **product of the two arguments**.

(Originally, **calloc** was probably meant for arrays.)

calloc Allocates a New Chunk of Memory

```
void* calloc (size_t num_elts,
             size_t elt_size);
```

As with **malloc**, **calloc** returns

- a **pointer to a new chunk of memory** (from the heap), or
- **NULL on failure** (memory not available).

The **memory returned from calloc** is filled with 0 bits.