

Null Vs. void * ?

- **Null pointer** is a special reserved value of a pointer. A pointer of any type has such a reserved value.
- Formally, each specific pointer type (**int ***, **char *** etc.) has its own dedicated null-pointer value. Conceptually, when a pointer has that null value it is not pointing anywhere.
- It's actual value is *system dependent* and may vary depending on the type.
- A billion dollar mistake!

Null Vs. void * ?

- **Void pointer** is a specific pointer type - **void *** - a pointer that points to some data location in storage, which doesn't have any specific type.
- So, once again, null pointer is a value, while void pointer is a type. These concepts are totally different and non-comparable. That essentially means that your question, as stated, is not exactly valid.
- *It is like asking, for example, "What is the difference between a triangle and a car?"*

Null Vs. void * ?

- They are two different concepts:
 - "void pointer" is a type (`void *`).
 - "null pointer" is a pointer that has a value of zero (`NULL`).
- Example:

```
void *pointer = NULL;
```

- That's a NULL void pointer.

What is the difference between

NULL, '\0' and 0

- All three define the meaning of zero in different context.
- pointer context - **NULL** is used and means the value of the pointer is 0, independent of whether it is 32bit or 64bit (one case 4 bytes the other 8 bytes of zeroes).
- string context - the character representing the **digit zero (0)** has a hex value of 0x30.
- whereas the **NUL** character has hex value of 0x00 (used for terminating strings).

What is the difference between

NULL, '\0' and 0

- These three are always different when you look at the memory:

```
NULL - 0x00000000 or 0x00000000'00000000 (32 vs 64 bit)
NUL - 0x00 or 0x0000 (ascii vs 2byte unicode)
'0' - 0x20
```