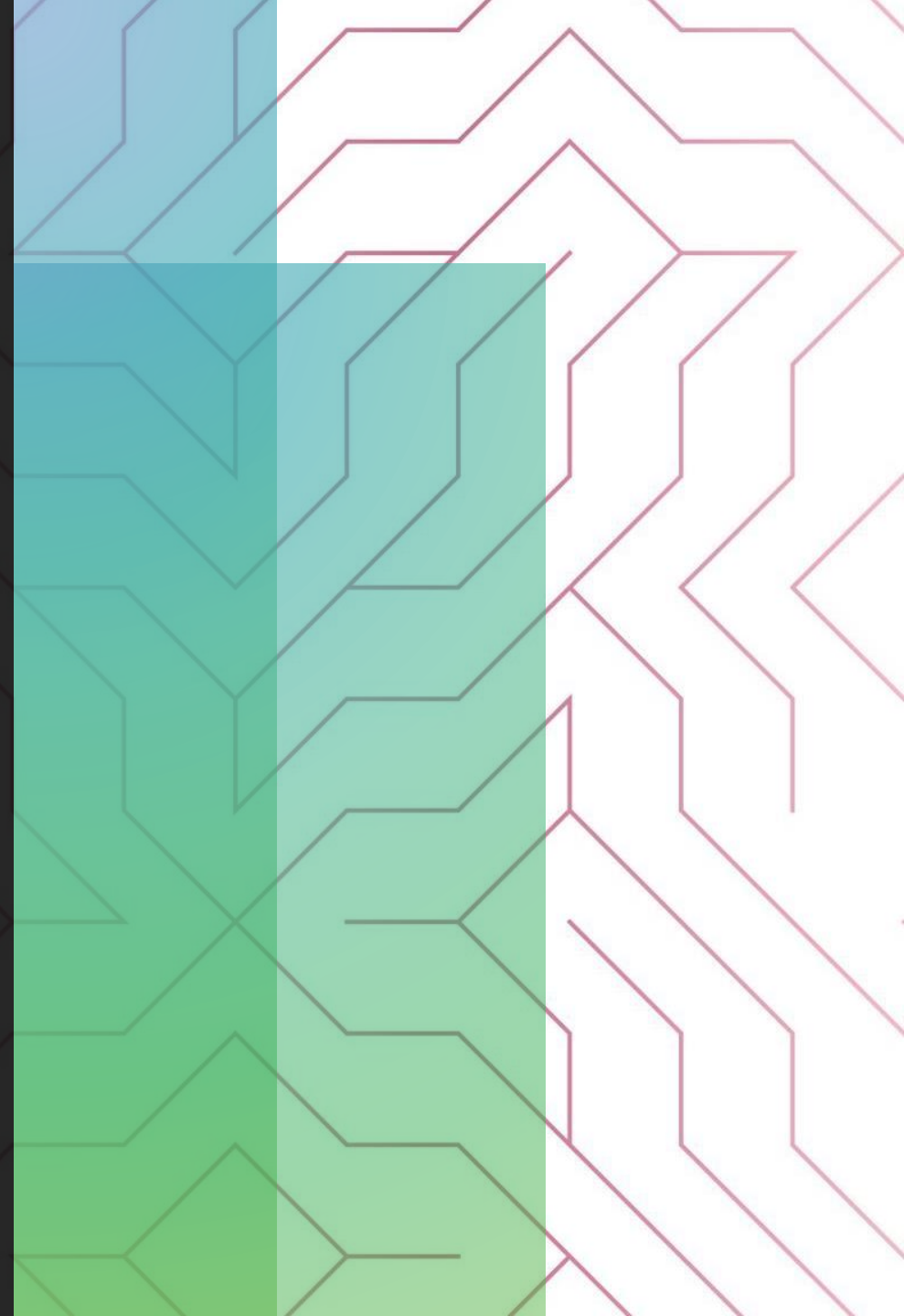



What is pointer in C- programming language





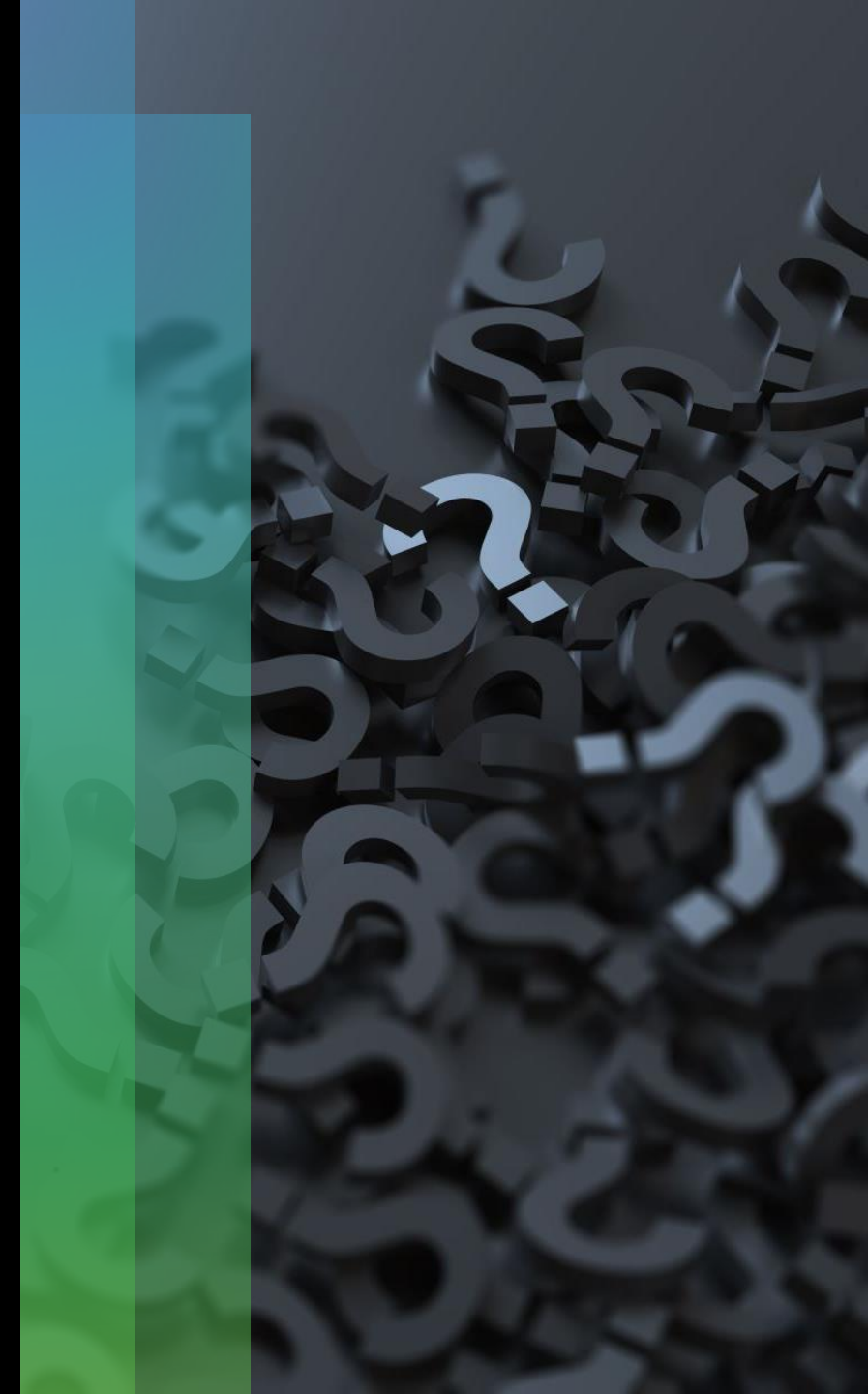
In the C programming language, a pointer is a variable that stores the memory address of another variable. Pointers are essential for tasks like dynamic memory allocation, working with arrays, and passing parameters to functions by reference. Here's a brief overview of pointers in C

1.Declaration: Pointers are declared using an asterisk (*) before the variable name. For example:copy code

- `int *ptr; // Declares a pointer`

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
----- OPERATOR CLASSES -----  
  
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

2. Assignment: Pointers can be assigned the address of a variable using the address-of operator (&):
copy code
`int x = 10; int *ptr = &x; // ptr now points`



3. Accessing the Value: You can access the value stored at the memory location pointed to by a pointer using the dereference operator (*):
copy code
`int y = *ptr; // y now contain`



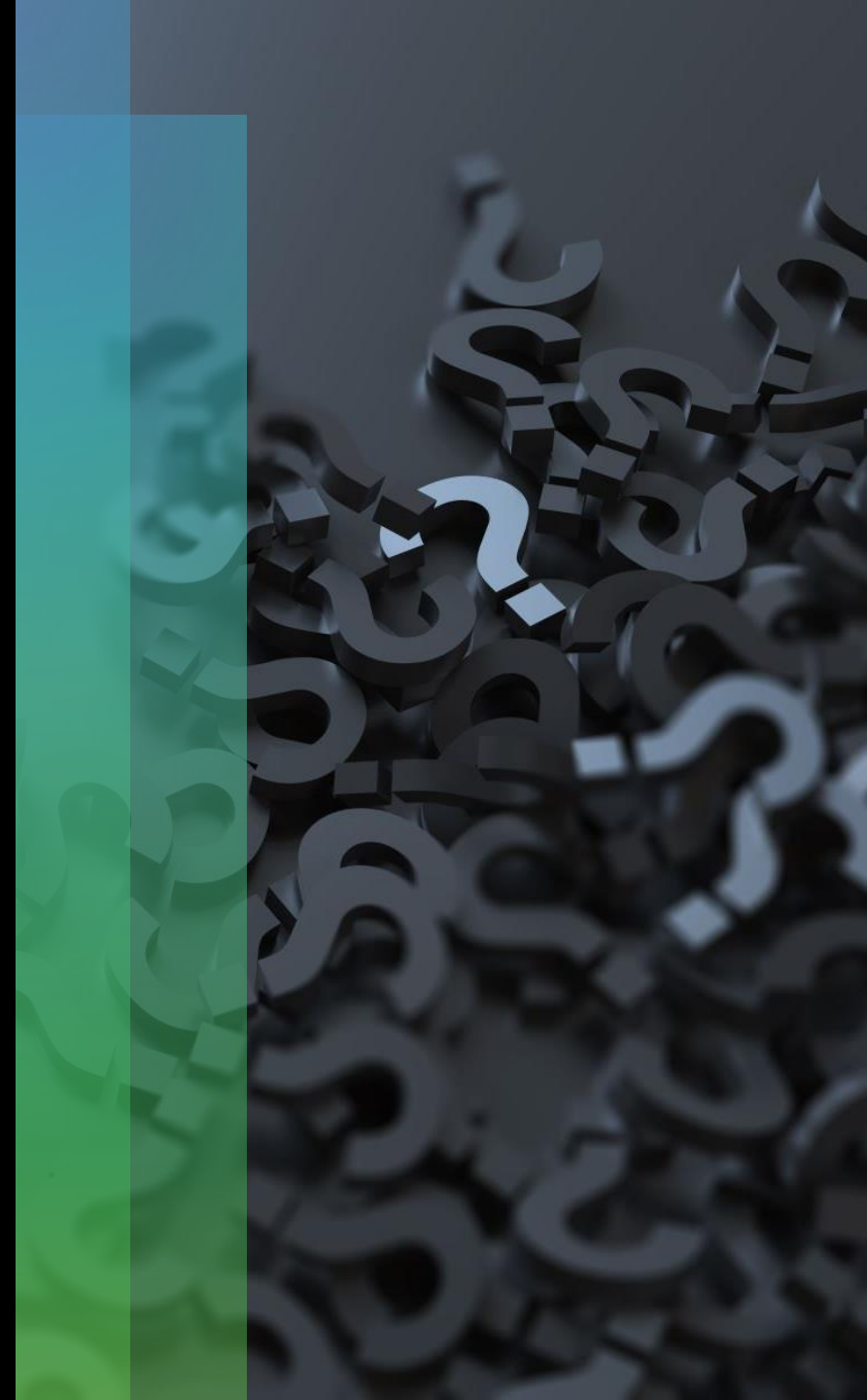


4.Pointer Arithmetic: Pointers can be incremented and decremented to move to the next or previous memory locations:copy codeptr++; // Moves ptr to the next

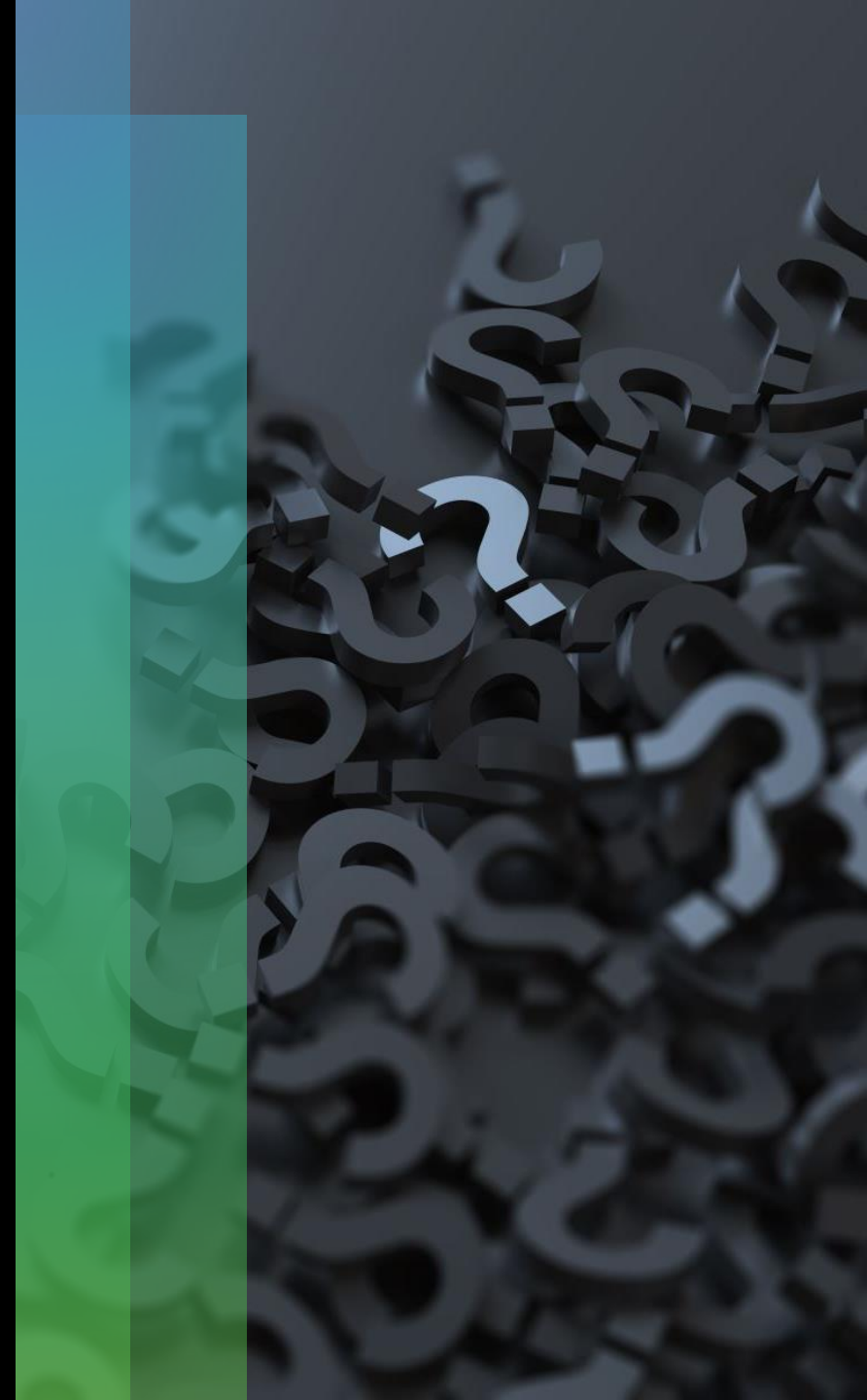
5.Null Pointers: Pointers can be assigned the special value NULL to indicate that they do not point to a valid memory location.



6.Pointer and Arrays: Arrays in C are closely related to pointers. The name of an array can be used as a pointer to its first element.



7.Pointers and Functions: Pointers can be used to pass variables by reference to functions, allowing the function to modify the original variable.



**8.Dynamic Memory Allocation:
Pointers are commonly used
with functions like malloc and
free to allocate and deallocate
memory dynamically**

