

12 - CONDITIONS and BRANCHES in C++

domingo, 2 de fevereiro de 2025

19:38

- evaluate a certain condition and execute a code if true or false
- Evaluation of a certain condition
- Everything from the code is stored in memory, and when a condition happens, it says to jump to certain portion of a memory to execute commands there
- If want to write fast code, can prevent to use if statements (so it don't consume memory)
- The comparison operator == was implemented somewhere
 - Each bit needs to be equal in this case
- The operators are not magical, they are implemented somewhere and we also can implement them
- 0 is the only false bool number, the rest is true
- if 1 is always true, and if 0 is always false. Same with nullptrs

```
const char* ptr = "Hello";
if(!ptr)
    Logr(ptr);
// else if(ptr == "Hello") // In C++, string literals are stored as
// arrays of characters, and their type decays into const char*. Comparing a
// pointer (like ptr) to a string literal using == checks if both pointers
// point to the same memory address, not if their contents are the same.
// This leads to unspecified behavior
else if(strcmp(ptr, "Hello"))
    Logr("Ptr is Hellow"); // Only check this condition if the first
// one fails
else
    Logr("Ptr is Null");
```