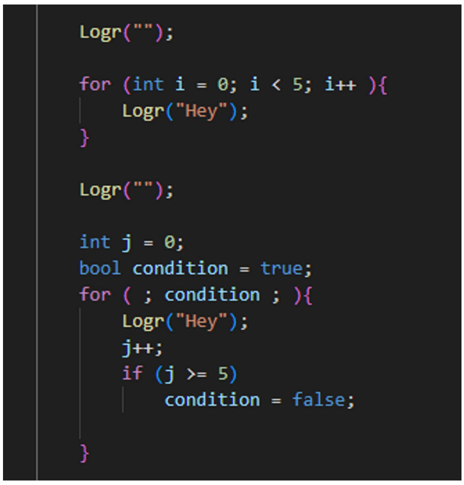
14 - Loops in C++ (for loops, while loops)

quarta-feira, 5 de fevereiro de 2025 07:29

- things that e can write to peform certain operations a lot of times without doce duplication
- important for big pictures, like keep a game running
- for(<iteratior>, <codition>, <code that will be called before the next iterator>
  - Define the variable, check if the condition is true, and execude the budy
  - Whe finished the bidy immplementation, it execites the code that will be called before the next iterator
  - incrementing the iterator and moving foward to te next item
- while(condition)
  - Executes the code until the condition becomes false
  - useful when you need a variable or not
  - convetion or style
- there is no difference between both loops

- ## Control Flow in C++ (continue, break, return)
- works with loops and give more control on it
- Continue
  - Go to the next iterator of this loop and ignore everything bellow this command
- Break
  - Get out of the loop, end the loop
- return
  - Get out of your function entirelly, exit the function (but mey need to return a value
  - Can be used anywhere to get out of a function, not only inside a loop
  - Every code bellow a return without a condition is dead code, and some compilrs will prevent this from happening



```cpp
Logr("");

for (int i = 0; i < 5; i++ ){
    Logr("Hey");
}

Logr("");

int j = 0;
bool condition = true;
for ( ; condition ; ){
    Logr("Hey");
    j++;
    if (j >= 5)
        condition = false;

}
```