Coding Styles for Loops

Considering assorted ways to terminate a for or while loop.
Simple Problem

Let’s try to find the location of the first occurrence of the character ‘c’ in a string.

(Never mind that this could easily be done by the indexOf() method.)

input: “wilson, chris”

return: 8
public static int lookFORc(String name)
{
    for (int i=0; i<name.length(); i++)
        if (name.charAt(i)==‘c’)
            return i;

    return -1;
}

The return i will terminate the loop (and method). It is adequate but not elegant. It is not adequate should we need i for further work after the loop.
With a For-Loop

If we needed i for some reason in the method…

```java
public static int lookFORc(String name) {
    for (int i=0; i<name.length(); i++)
        if (name.charAt(i)=='c')
            break;
    // here i is available
    if (i==name.length())
        return -1;
    else
        return i;
}
```
public static int lookFORc(String name) {
    boolean found = false;
    int loc = 0;
    while (loc<name.length()) {
        if (name.charAt(loc)=='c') {
            found = true; break;
        }
        loc++;
    }
    if (found)
        return loc;
    return -1;
}
public static int lookFORc(String name) {
    boolean found = false;
    int loc = 0;
    while (loc<name.length() && !found) {
        if (name.charAt(loc) == 'c')
            found = true;
        else
            loc++;
    }
    if (found)
        return loc;
    return -1;
}
public static int lookFORc(String name) {
    boolean found = false;
    int loc = 0;
    int len = name.length();
    while (loc<len && !found){
        if (name.charAt(loc)==‘c’)
            found = true;
        else
            loc++;
    }
    if (found)
        return loc;
    return -1;
}
Modified Problem

Now try to find the location of the first occurrence of the substring “chris”.

(This can also be done by a String method.)

input: “wilson, chris”

return: 8
public static int lookForMe(String name) {

    int myLen = 5; // length of “chris”
    boolean found = false;
    int loc = 0;
    int len = name.length();

    while (loc+myLen-1<len && !found){
        if (name.substring(loc,loc+myLen).equals("chris"))
            found = true;
        else
            loc++;
    }

    ...
}

Note how String types are tested for equality.