

- Solution

In this lesson, we'll discuss the solution to the exercise.

WE'LL COVER THE FOLLOWING ^

- Solution
- Explanation

Solution

```
#include <regex>
#include <iostream>
#include <string>

int main(){

    std::cout << std::endl;

    std::string emailText = "A text with an email adresse: rainer@grimm-jaud.de.";

    // regular expression for the email address
    std::string regExprStr(R"([\w.%+-]+)@([\w.-]+\.[a-zA-Z]{2,4})");

    // regular expression holder
    std::regex rgx(regExprStr);

    // search result holder
    std::smatch smatch;

    // searching for a match
    if (std::regex_search(emailText, smatch, rgx)){

        std::cout << "Text: " << emailText << std::endl;
        std::cout << std::endl;
        std::cout << "Before the email address: " << smatch.prefix() << std::endl;
        std::cout << "After the email address: " << smatch.suffix() << std::endl;
        std::cout << std::endl;
        std::cout << "Length of email address: " << smatch.length() << std::endl;
        std::cout << std::endl;
        std::cout << "Email address: " << smatch[0] << std::endl;
        std::cout << "Local part: " << smatch[1] << std::endl;
        std::cout << "Domain name: " << smatch[2] << std::endl;

    }

}
```

```
std::cout << std::endl;
```

```
}
```



Explanation

We have created a string that contains an email address along with some text. We used a regular expression for the email address and inserted it in a regular expression holder `rgx`. `std::smatch` is used to partition the string according to the requirements as it matches the given `emailText` with the regular expression holder `rgx`. Now, the `smatch` object contains all the information regarding the email address and the remaining text.

In the next lesson, we'll discuss how we can alter the text if it matches our regex condition.