Search

In this lesson, we'll examine the different search features available in the string class.

C++ offers many ways to search in a string. Each way exists in various overloaded forms.

> **ℹ Search is called find**
> It seems a bit odd but the algorithms for searching in a string start with the prefix “find”. If the search was successful, we get an index of type `std::string::size_type`, if not, we get the constant `std::string::npos`. The first character has the index 0.

The find algorithms support:

- searching for a character from a C or C++ string.
- searching forward and backward.
- searching for the presence or lack of characters in a C or C++ string.
- starting the search at an arbitrary position in the string.

The arguments of all six variations of the find function follow a similar pattern. The first argument is the text we are searching for. The second argument holds the start position of the search, and the third represents for the number of characters starting from the second argument.

Here are the six variations:

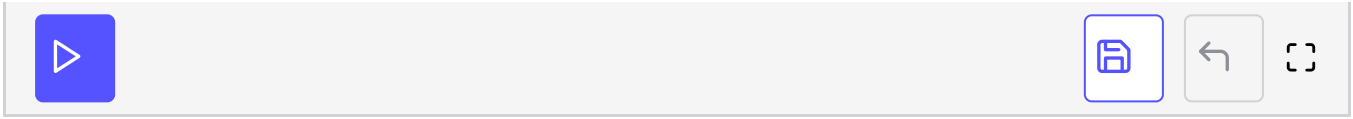
Methods	Description
`str.find(...)`	Returns the first position of a character, a C or C++ string in `str`.

`str.rfind(...)`	Returns the last position of a character, a C or C++ string in `str`.
`str.find_first_of(...)`	Returns the first position of a character from a C or C++ string in `str`.
`str.find_last_of(...)`	Returns the last position of a character from a C or C++ string in `str`.
`str.find_first_not_of(...)`	Returns the first position of a character in `str`, which is not from a C or C++ string.
`str.find_last_not_of(...)`	Returns the last position of a character in `str`, which is not from a C or C++ string.

Find variations of the string

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

int main(){
  std::string str;

  auto idx= str.find("no");
  if (idx == std::string::npos) std::cout << "not found";  // not found

  str= {"dkeu84kf8k48kdj39kdj74945du942"};
  std::string str2{"84"};

  std::cout << str.find('8') << std::endl;                                  // 4
  std::cout << str.rfind('8') << std::endl;                                 // 11
  std::cout << str.find('8', 10) << std::endl;                              // 11
  std::cout << str.find(str2) << std::endl;                                 // 4
  std::cout << str.rfind(str2) << std::endl;                                // 4
  std::cout << str.find(str2, 10) << std::endl;                             // 18446744073709551

str2="0123456789";

std::cout << str.find_first_of("678") << std::endl;                        // 4
std::cout << str.find_last_of("678") << std::endl;                         // 20
std::cout << str.find_first_of("678", 10) << std::endl;                    // 11
std::cout << str.find_first_of(str2) << std::endl;                         // 4
std::cout << str.find_last_of(str2) << std::endl;                          // 29
```

```
std::cout << str.find_first_of(str2, 10) << std::endl;                  // 10
std::cout << str.find_first_not_of("678") << std::endl;                 // 0
std::cout << str.find_last_not_of("678") << std::endl;                  // 29

std::cout << str.find_first_not_of("678", 10) << std::endl;             // 10
std::cout << str.find_first_not_of(str2) << std::endl;                  // 0
std::cout << str.find_last_not_of(str2) << std::endl;                   // 26
std::cout << str.find_first_not_of(str2, 10) << std::endl;              // 12
return 0;
}
```



Find(search) in a string

The call `std::find(str2, 10)` returns `std::string::npos`. The value for this, on this platform, is `18446744073709551615`.

To build upon our understanding of this concept, we'll solve an exercise.