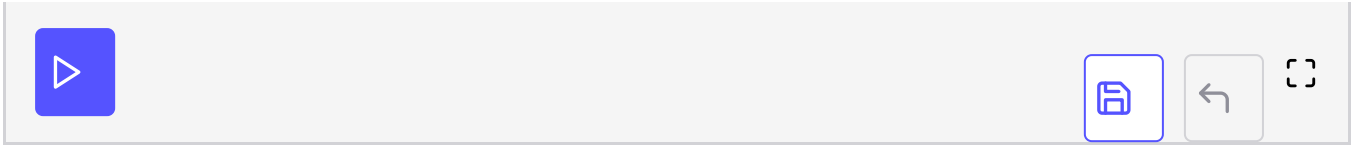
- Example

An example for perfect forwarding in modern C++.

WE'LL COVER THE FOLLOWING

Lvalue and rvalue arguments

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

template <typename T, typename T1>
T create(T1&& t1){
  return T(std::forward<T1>(t1));
}

int main(){

  std::cout << std::endl;

  // Lvalues
  int five = 5;
  int myFive = create<int>(five);
  std::cout << "myFive: "  << myFive << std::endl;

  std::string str{"Lvalue"};
  std::string str2 = create<std::string>(str);
  std::cout << "str2: " << str2 << std::endl;

  // Rvalues
  int myFive2 = create<int>(5);
  std::cout << "myFive2: " << myFive2 << std::endl;

  std::string str3 = create<std::string>(std::string("Rvalue"));
  std::cout << "str3: " << str3 << std::endl;

  std::string str4 = create<std::string>(std::move(str3));
  std::cout << "str4: " << str4 << std::endl;

  std::cout << std::endl;

};
```



Explanation

- We have used a universal reference in line 6 of the code so it can bind rvalues or lvalues.
- In lines 16 and 20, we have called the function `create` using lvalues `myfive` and `str`.
- In lines 24 and 27, we have called the function `create` using the rvalues `5` and `Rvalue`.
- We have implemented an interesting thing in line 30. We have called the function `create` with an rvalue reference of `str3` generated by using the function, `std::move`.

Let's test our understanding of this topic with an exercise in the next lesson.