- Example

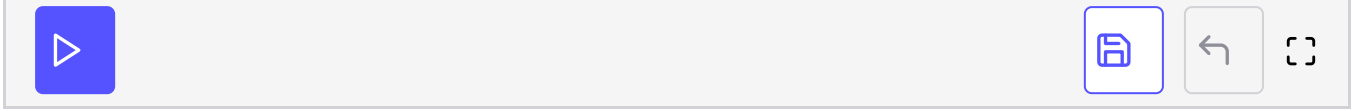
Let's have a look at an example of tag dispatching.

WE'LL COVER THE FOLLOWING

Example: Templates Tag Dispatching

```
// TemplatesTagDispatching.cpp

#include <iterator>
#include <forward_list>
#include <list>
#include <vector>
#include <iostream>

template <typename InputIterator, typename Distance>
void advance_impl(InputIterator& i, Distance n, std::input_iterator_tag) {
        std::cout << "InputIterator used" << std::endl;
    while (n--) ++i;
}

template <typename BidirectionalIterator, typename Distance>
void advance_impl(BidirectionalIterator& i, Distance n, std::bidirectional_iterator_tag) {
        std::cout << "BidirectionalIterator used" << std::endl;
    if (n >= 0)
        while (n--) ++i;
    else
        while (n++) --i;
}

template <typename RandomAccessIterator, typename Distance>
void advance_impl(RandomAccessIterator& i, Distance n, std::random_access_iterator_tag) {
        std::cout << "RandomAccessIterator used" << std::endl;
    i += n;
}

template <typename InputIterator, typename Distance>
void advance_(InputIterator& i, Distance n) {
    typename std::iterator_traits<InputIterator>::iterator_category category;
    advance_impl(i, n, category);
}
```

```
int main(){

}
```



Explanation

The expression `std::iterator_traits::iterator_category` category in line 32 determines the iterator category at compile-time. Based on the iterator category, the most specific variant of the function template `advance_impl(i, n, category)` is used in line 33. Each container returns an iterator of the iterator category which corresponds to its structure.

In the next lesson, we'll solve an exercise for tag dispatching in idioms and patterns.