# Speeding Up Enum

In this lesson, you will see how it is possible to improve the performance of enum.

WE'LL COVER THE FOLLOWING



## Setting `enum` as `const` to increase speed

`enum` can be set as a constant to speed up the performance. This way, during execution, instead of referencing the function generated by TypeScript to JavaScript, it will use the value.

For example, without constant `enum`, the value set to a direction with `Orientation.East` will be equal to a function that looks for the value in the map to get the value. However, with a constant, the value is set in the transpiled code to `0` directly – no more function or mapping.

## Drawbacks

There are not a lot of drawbacks to this. You can still use the `enum` with the dot notation and the name of one of the entries. You can also use the name of the `enum` with the square brackets and the name of one of the entries.

However, you won't be able to use the square brackets with the value.

The only difference is that constant `enum` doesn't allow for redefining values once they're initialized, which is allowed with default non-constant `enum`. However, in both cases, it's possible to add an entry using the square brackets.

## The JavaScript output

As before, if the `preserveConstEnums` is `false`, the JavaScript generated is:

```
let Orientation;
(function (Orientation) {
    Orientation[Orientation["East"] = 0] = "East";
    Orientation[Orientation["West"] = 1] = "West";
    Orientation[Orientation["North"] = 2] = "North";
    Orientation[Orientation["South"] = 3] = "South";
})(Orientation || (Orientation = {}));
let directionInNumber = Orientation.East;
```

With `preserveConstEnums` set to `true`, the JavaScript generated looks like the following:

```
let directionInNumber = Orientation.East;
```

This feature has been available since version 1.4 and can be turned off by using the compiler option `preserveConstEnums` and setting it to `true`.