TypeScript Enum

Summary: in this tutorial, you'll learn about the TypeScript enum type and how to use it effectively.

What is an enum

An enum is a group of named constant values. Enum stands for enumerated type.

To define an enum, you follow these steps:

- First, use the `enum` keyword followed by the name of the enum.
- Then, define constant values for the enum.

The following shows the syntax for defining an enum:

```
enum name {constant1, constant2, ...};
```

In this syntax, the `constant1`, `constant2`, etc., are also known as the members of the enum.

TypeScript enum type example

The following example creates an enum that represents the months of the year:

```
enum Month {
    Jan,
    Feb,
    Mar,
    Apr,
    May,
    Jun,
    Jul,
    Aug,
    Sep,
    Oct,
    Nov,
    Dec
};
```

In this example, the enum name is `Month` and constant values are `Jan`, `Feb`, `Mar`, and so on.

The following declares a function that uses the `Month` enum as the type of the `month` parameter:

```
function isItSummer(month: Month) {
  let isSummer: boolean;
  switch (month) {
    case Month.Jun:
    case Month.Jul:
    case Month.Aug:
      isSummer = true;
      break;
    default:
      isSummer = false;
      break;
  }
  return isSummer;
}
```

And you can call it like so:

```
console.log(isItSummer(Month.Jun)); // true
```

This example uses constant values including `Jan`, `Feb`, `Mar`, ... in the enum rather than magic values like `1`, `2`, `3`,... This makes the code more obvious.

How TypeScript enum works

It's a good practice to use the constant values defined by enums in the code.

However, the following example passes a number instead of an enum to the `isItSummer()` function. And it works.

```
console.log(isItSummer(6)); // true
```

This example uses a number (`6`) instead of a constant defined by the `Month` enum. And it works.

Let’s check the generated Javascript code of the Month enum:

```
var Month;
(function (Month) {
    Month[Month["Jan"] = 0] = "Jan";
    Month[Month["Feb"] = 1] = "Feb";
    Month[Month["Mar"] = 2] = "Mar";
    Month[Month["Apr"] = 3] = "Apr";
    Month[Month["May"] = 4] = "May";
    Month[Month["Jun"] = 5] = "Jun";
    Month[Month["Jul"] = 6] = "Jul";
    Month[Month["Aug"] = 7] = "Aug";
    Month[Month["Sep"] = 8] = "Sep";
    Month[Month["Oct"] = 9] = "Oct";
    Month[Month["Nov"] = 10] = "Nov";
    Month[Month["Dec"] = 11] = "Dec";
})(Month || (Month = {}));
```

And you can output the `Month` variable to the console:

```
{
  '0': 'Jan',
  '1': 'Feb',
  '2': 'Mar',
  '3': 'Apr',
  '4': 'May',
  '5': 'Jun',
  '6': 'Jul',
  '7': 'Aug',
  '8': 'Sep',
  '9': 'Oct',
  '10': 'Nov',
  '11': 'Dec',
  Jan: 0,
  Feb: 1,
  Mar: 2,
  Apr: 3,
  May: 4,
  Jun: 5,
  Jul: 6,
  Aug: 7,
  Sep: 8,
  Oct: 9,
  Nov: 10,
  Dec: 11
}
```

The output indicates that a TypeScript enum is an object in JavaScript. This object has named properties declared in the enum. For example, `Jan` is `0` and `Feb` is `1`.

The generated object also has number keys with string values representing the named constants.

That's why you can pass a number into the function that accepts an enum. In other words, an enum member is both a number and a defined constant.

Specifying enum members' numbers

TypeScript defines the numeric value of an enum's member based on the order of that member that appears in the enum definition. For example, `Jan` takes 0, `Feb` gets 1, etc.

It's possible to explicitly specify numbers for the members of an enum like this:

```
enum Month {
    Jan = 1,
    Feb,
    Mar,
    Apr,
    May,
    Jun,
    Jul,
    Aug,
    Sep,
    Oct,
    Nov,
    Dec
};
```

In this example, the `Jan` constant value takes 1 instead of 0. The `Feb` takes 2, and the `Mar` takes 3, etc.

When to use an enum

You should use an enum when you:

- Have a small set of closely related fixed values.
- And these values are known at compile time.

For example, you can use an enum for the approval status:

```
enum ApprovalStatus {
    draft,
    submitted,
    approved,
    rejected
};
```

Then, you can use the `ApprovalStatus` enum like this:

```
const request =  {
    id: 1,
    status: ApprovalStatus.approved,
    description: 'Please approve this request'
};

if(request.status === ApprovalStatus.approved) {
    // send an email
    console.log('Send email to the Applicant...');
}
```

Summary

- A TypeScript enum is a group of constant values.

- Under the hood, an enum is a JavaScript object with named properties declared in the enum definition.
- Do use an enum when you have a small set of fixed values that are closely related and known at compile time.