let, const and var

Introduction to Variable Declaration in JavaScript

In this section, we will examine the three different ways of declaring variables in JavaScript: **let**, **const**, and **var**.

The let Keyword

Up until this point, we have always used the **let** keyword to declare variables. However, there are also **const** and **var**. Now, **let** and **const** were introduced in ES6, making them modern JavaScript, while the **var** keyword is the old way of declaring variables.

We use the **let** keyword to declare variables that can change later, essentially during the execution of our program. For example, we used **let** to declare a variable and then changed its value later. In technical terms, this is called reassigning a value to a variable or mutating the variable.

Example: Mutating Variables with let

For instance, we can set the age to 30 at one point in the program and then change it to something else, such as when the person turns 31. The age changes from 30 to 31, and so we mutate the variable. It is perfectly acceptable to declare a variable with **let** at one point and then assign a new value to it later.

Declaring Empty Variables with let

We can also declare empty variables with **let** and assign values to them later, for example, based on some condition. This is useful when we want to declare all variables at the top of a file but only assign actual values later in the program.

The const Keyword

On the other hand, we use the **const** keyword to declare variables that are not supposed to change at any point in the future. The value in a **const** variable cannot be changed. For example, the birth year is a perfect example for **const** because it can never change, while the age can change.

javascript Code Sample

```
const birthYear = 1991;
```

javascript Code Sample

```
// Attempting to reassign a const variable will result in an error

birthYear = 1990; // TypeError: Assignment to constant variable.
```

If we try to reassign a **const** variable, we get a type error: assignment to constant variable.
The **const** keyword creates a variable that cannot be reassigned, or in technical terms, an immutable variable.

Initialization Requirement for const

Variables created with **const** are immutable, which also means that we cannot declare empty **const** variables. For example, this is not legal:

javascript Code Sample

```
const year;
```

This will result in an error: missing initializer in **const** declaration. When using **const**, we need an initial value.

Best Practices: When to Use let or const

With these two different ways of declaring variables, you might ask whether to use **let** or **const**. As a best practice for writing clean code, it is recommended to use **const** by default and **let** only when you are certain that the variable needs to change at some point in the future. If you have a variable that is never supposed to change, such as a birth year, always use **const**. If you are sure that the age variable is never changing inside your program, declare it using **const** as well.

Having as little variable mutation as possible is a good practice because changing variables introduces a potential to create bugs. By default, always use **const** and use **let** only when the value of the variable needs to change.

The var Keyword

There is also a third way in JavaScript of declaring variables, which is the **var** keyword. However, this should be completely avoided. We should still know how it works for legacy reasons because you may see this in older codebases or tutorials.

javascript Code Sample

```
var job = "programmer";

job = "teacher";
```

At first sight, **var** works similarly to **let**—you can change the value of the variable later. However, although **var** and **let** look similar, they are quite different beneath the surface. There are also many differences between **let**, **const**, and **var**, which will be covered in detail later.

For now, what matters is that you should never use **var**.

Declaring Variables Without Keywords

Some people might point out that it is not mandatory to declare variables at all. For example:

javascript Code Sample

```
lastName = "Schmedtmann";

console.log(lastName);
```

JavaScript will execute this script even without declaring a variable using **let**, **const**, or **var**. However, this is a terrible idea because it does not create a variable in the current scope. Instead, JavaScript creates a property on the global object. You should always properly declare variables and never write a variable like this without declaring it.

Conclusion

You are making great progress, even though the code does not do much yet. It is important to get the fundamentals down before moving on to more advanced topics. In the next video, we will discuss operators in JavaScript.

Key Takeaways

- Variables in JavaScript can be declared using **let**, **const**, or **var**, each with distinct behaviors.
- **let** allows variable mutation, while **const** creates immutable variables that require initialization.

- The **var** keyword is the old way of declaring variables and should be avoided in modern JavaScript.
- Declaring variables without a keyword is possible but leads to undesirable global properties.