Variable Declarations in JavaScript

This lesson teaches how variables are declared in JavaScript and what's the difference between var, let and const keywords.

Earlier before 2015, the only keyword available to declare variables was the `var` keyword. With JavaScript ES6 introduced, the two new keywords that came into existence are `const` and `let`. Let's see how each of these keywords differ from one another and what's the best time to use them.

The different variable declarations with `var`, `let` and `const` can be confusing for newcomers to React even though they are not React specific. Maybe it is because JavaScript ES6 was introduced when React became popular. In general, I try to introduce `let` and `const` very early in my workshops. It simply starts with exchanging `var` with `const` in a React component:

```
import React from 'react';
require('./style.css');

import ReactDOM from 'react-dom';
import App from './app.js';

ReactDOM.render(
  <App />,
  document.getElementById('root')
);
```

Also, JavaScript didn't have a dedicated keyword to declare constants only which is why `const` was introduced. The variables assigned using `const` are read-only which means that once they are initialized using `const`, they cannot be reassigned.

Scope of var, let and const

The primary reason to introduce two more keywords in the new JS version was to allow programmers to decide scope options for their defined variables.

The scopes of all these keywords are mentioned below:

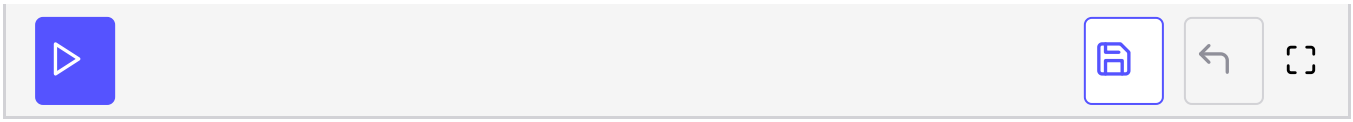
- `var`: Function in which the variable is declared
- `let`: Block in which the variable is declared
- `const`: Block in which the variable is declared

> *Block scopes are what you get when you use if statements, for statements or write code inside curly brackets.*

Run the following example and see what's the result.

```
function foo(){
  for (var i=0; i<5 ; i++){
    console.log(i);
  }
  console.log(i);
}

foo();
```



If you replace `var` with `let` and then run the code again you will get the error `ReferenceError: i is not defined`. This shows that `i` is only accessible within the curly brackets.

Rules of Thumb:

- Don't use `var`, because `let` and `const` is more specific
- Default to `const`, because it cannot be re-assigned or re-declared
- Use `let` when you want to re-assign the variable in future
- Always prefer using `let` over `var` and `const` over `let`

When to use what?

While `let` is usually used in a `for` loop for incrementing the iterator, `const` is normally used for keeping JavaScript variables unchanged. Even though it is

possible to change the inner properties of objects and arrays when using

`const`, the variable declaration shows the intent of keeping the variable unchanged.