# let and const Exercise

http://curric.rithmschool.com/springboard/exercises/js-let-const/

In this exercise, you’ll refactor some ES5 code into ES2015.

ES5 Global Constants

**var** PI = 3.14;

PI = 42; *// stop me from doing this!*

ES2015 Global Constants

*/\* Write an ES2015 Version \*/*

**const** PI = 3.14;

PI = 42; *// Error*

const pi = Math.PI;

const radius = 10;

const areaOfCircle = (radius, pi) => {

return (pi \* radius \* radius).toFixed(2);

};

console.log('The area of the circle is ' + areaOfCircle(radius, pi));

Quiz

1. What is the difference between **var** and **let**?

The difference between var and let is the scope of the variable, and whether they can be redeclared. Variables declared by **let** are only available inside the block where they're defined. Variables declared by **var** are available throughout the function in which they're declared.

**var** variable can be reassigned, redeclared, and mutated, and are available within the scope of the function. Variables declared with **let** can be reassigned a value but cannot be redeclared.

1. What is the difference between **var** and **const**?

**var** and **let** create variables that can be reassigned another value. **const** creates "constant" variables that cannot be reassigned another value.

1. What is the difference between **let** and **const**?

Variables declared with **const** cannot reassigned. Variables declared with **let** may be reassigned or mutated, such as a counter in a loop, or a value swap in an object or an array. The **let** variable can be used only in the block it's defined in, which is not always the entire containing function.

1. What is hoisting?

JavaScript Hoisting refers to the process whereby the interpreter appears to move the declaration of functions, variables, or classes to the top of their scope, prior to execution of the code.

Hoisting is JS's default behavior of defining all the declarations at the top of the scope before code execution. One of the benefits of hoisting is that it enables us to call functions before they appear in the code. JavaScript only hoists declarations, not initializations.