***let* and *const* Exercise**

**ES5 Global Constants**

var PI = 3.14;

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

Const PI = 3.14.

Variables using const can not be redeclared or renamed. They can however be mutated

**ES2015 Global Constants**

/\* Write an ES2015 Version \*/

Const name = Kristina

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

The difference between var and let is that var can be reassigned, redeclared, and/or mutated. Var is global scoped. You can also access var variables in the window object on the console. Var can be block scoped as well, especially if it is declared within a function. some built in methods/objects can be overwritten, overrided or removed completely if we use var with a conflicting name like (var = console) Let can be reassigned but not redeclared, it is block scope it cannot be accessed in the window object. They both use the same syntax. They can be initialized but they do not have to be assigned with assignment happening later on

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

Var as noted above can be redeclared, reassigned and mutated. It can be accessed in the window object. Const is only allowed to mutate. It is block scoped , it cannot not be redeclared or reassigned(this will result in a typeError), it is not accessible through the window object. If you reassign the same const (for example const PI = 3.14, and const PI = 4.56) to two different values it will result in a syntax error. Var can be initialized with assignment happening later while const requires that you have an assignment for the variable upon initialization of the const variable

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

Let and Const are both block scoped. Let can be reassigned but not redeclared, while const is only allowed to mutate. Neither can be accessed in the window object. Let variables can be initialized without a value assigned, while const requires a variable to be assigned upon its declaration

What is hoisting?

It is the human description of how the JS compiler works. variables using the let/const method are not hoisted, meaning they will result in a reference error. While variables using the keyword var will simply return undefined. They are lifted to the top of whatever scope they are declared in. Function declarations are also able to be hoisted. They can be invoked long before they are ever defined.