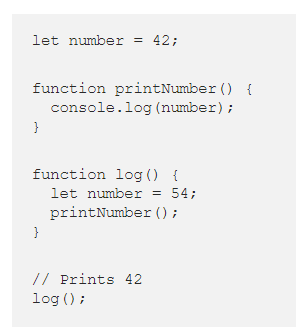
Lexical Scope (also known as Static Scope) literally means that scope is determined at the [lexing time](https://en.wikipedia.org/wiki/Lexical_analysis" \t "_blank) (generally referred to as compiling) rather than at runtime. For example:



Here the console.log(number) will always print 42 no matter from where the function printNumber() is called. This is different from languages with the **dynamic scope** where the console.log(number) will print different value depending on from where the function printNumber() is called.

If the above code was written in a language that supports dynamic scoping, the console.log(number) would have printed 54 instead.

Scope chain. In JS when variable is called first it is looked in a block scope, if not found it stars searching function scope, lastly it will look for global scope. If not found even in global scope reference error is thrown

Learn more: https://blog.bitsrc.io/understanding-scope-and-scope-chain-in-javascript-f6637978cf53