

## 5 Vocabulary around variables and scope

```
var a;
```

### Variable Declaration

The creation of the variable.

```
a = 12;
```

### Variable Initialization

The initial assignment of value to a variable.

```
a = "me";
```

### Variable Assignment

Assigning value to a variable.

```
console.log(a);
```

```
var a = "me";
```

### Hoisting

Variables are declared at the top of the function automatically, and initialized at the time they are run.

### Scope

The limits in which a variable exists.

#### Global scope

The outer most scope is called the Global scope.

#### Functional scope

Any variables inside a function is in scope of the function.

#### Lexical Environment (Lexical scope)

The physical location (scope) where a variable or function is declared is its lexical environment (lexical scope).

#### Rule:


(1) Variables in the outer scope can be accessed in a nested scope; But variables inside a nested scope CANNOT be accessed by the outer scope. (a.k.a private variables.)

(2) Variables are picked up from the lexical environment.

```
var a = "global";

function first(){
  var a = "fresh";

  function second(){
    console.log(a);
  }
}
```



### Scope chain

The nested hierarchy of scope is called the scope chain. The JS engine looks for variables in the scope chain upwards (it its ancestors, until found)