1. `foreach()` iterates over an iterable object the same way a `for...of` loop does. A `for` loop is more flexible where it can be a `for...of`, `for..in`, or a typical `for(Initialization ; Condition; Operation)` loop. The `for` statement allows for more complex loop logic to be used.
2. Creates a new object and binds it to `this` since it is a constructor function
3. 101
4. Hoisting is putting a `var` variable as close as possible to the top of the function. For example:

```
function foo(){
      var x = 10
      console.log(x)
}
// is the same as:
function bar(){
      console.log(x)
      var x = 10
}
```

`x` will be pulled to the top of the function before execution and both the functions will have the same behaviour. in this example:

```
function foo(){
      var x = 10
      function bar(){
            var y = 20
            console.log(x) // 10 (x is relatively global to bar)
            console.log(y) // 20 (y is in scope)
      }
      bar()
      console.log(x)  // 10 (in scope)
      console.log(y)  // ReferenceError (y is in the scope of bar)
}
```