5 Lessons I Learned from Eloquent Javascript & 1 Walkthrough of a Problem & It's Solution

By Henry van Wagenberg

Based on Research & Learning from Oct. 25 - Nov. 22 2017

1st Lesson: How do Loops Work in Javascript?

* There are 3 types of loops:

* while loop

```
var number = 3, counter = 0;
while(counter <= 3) {
  number = number + 2;
  counter++;
}
console.log(number);
```

* do loop

```
do
  {
    var yourName = prompt("Who are you?");
  }
  while(!yourName)
    console.log(yourName);
```

* for loop

```
for(var number = 0; number <= 6;
  number = number + 2) {
    console.log(number);
}
```

* Question for the audience: What is the equivalent to a "while" loop in Ruby? Is there even one?

2nd Lesson: What are Values? What is state?

A basic piece of learning for me is the fact that the computer experiences all data as empty 0s and 1s until we define a value by assigning it a "role" . In Javascript there are 6 types of roles: functions, objects, strings, numbers and undefined values. When we call that role, when we give it a name, we create it as a value: an island of meaning in an otherwise identical ocean of meaningless data.

State is all the variables that are assigned. Variables are how you catch and grasp values. Once you create a variable grasping a value, the variable name becomes an expression whose value is what the variable grasps.

Lesson 3: In JS You can put stuff like anywhere... sometimes will produce subtly different outputs

```
var number = 3, counter = 0;
while(counter <= 3) {
  number = number + 2;
  counter++;
}
console.log(number);
```

* Works! But outputs only once for final evaluation after loop is done

```
var number = 3, counter = 0;
while(counter <= 3) {
  console.log(number);
  number = number + 2;
  counter++;
}
```

* Works! Outputs while the loop is in operation.

Lesson 4: What are Unary and Binary and Ternary Operators in Javascript?

Functions that do something to a value producing either a side effect or a change in value: for example "-" turns a value negative or "+" can be used as a binary operator between "7" and "5" "typeof" is an operator to tell you what role the value has.

A ternary operator (while also exists in Ruby) picks from between the 2nd and 3rd arguments depending on whether the 1st argument is true or false.

```
console.log(henry_net_worth >= 1000 ? "amazing" : "sad")
```

Interesting fact: || and && are also technically "operators." These so-called logical operators "convert" the left-side into a Boolean role type of "true" or "false."

Lesson 5: JS variables defined with "var" are always global unless defined within a function... this is misleading and tricky

* Rational people would assume that a variable assigned inside a block is local to that block. For example:

* { var henry = "great" }

* not in JS! Haha. It's global. Only inside a var henry = function() { var henry = "secret" }

* In ECMAScript 6 there is a new way though:

  * { let henry = "great" } makes the variable henry local to the block.

A problem walk-through.

Problem:

FizzBuzz

Write a program that uses `console.log` to print all the numbers from 1 to 100, with two exceptions. For numbers divisible by 3, print `"Fizz"` instead of the number, and for numbers divisible by 5 (and not 3), print `"Buzz"` instead.

When you have that working, modify your program to print `"FizzBuzz"`, for numbers that are divisible by both 3 and 5 (and still print `"Fizz"` or `"Buzz"` for numbers divisible by only one of those).

(This is actually an interview question that has been claimed to weed out a significant percentage of programmer candidates.

A problem walk-through. Solution:

```
countNumber = 1;
while(countNumber <= 100) {
    if (countNumber % 3 === 0 && countNumber % 5 === 0)
      console.log("FizBuzz");
    else if (countNumber % 3 === 0)
      console.log("Fizz");
    else if (countNumber % 5 === 0)
      console.log("Buzz");
    else
      console.log(countNumber);
  countNumber++;
};
```

Bigger Picture Lessons

I like learning
Javascript is fun
Grundlagenwissen: sie lohnen sich!
Ich bin dankbar für Eruditus
I wish I'd done this years ago
This is fun
Anki is a powerful learning tool