

# JAVASCRIPT INDUCTION, BUGS AND CHRISTIANITY

*José Rodríguez*  
[jose@ejristos.com](mailto:jose@ejristos.com)  
*Bogotá, Colombia, 31/VIII/2020*

## *Abstract*

*The Reader is challenged to run a first program in a browser. It contains a simple Javascript program inside an HTML background plus some styling with CSS. Many errors or bugs expect anyone that aims at programming. The relation with Christianity is explained.*

## **INTRODUCTION**

To get connected to the Internet through cellular phone is a universal, unstoppable trend. On the other hand, cell phones have more computing power than most desktops in use. So, the dependence of programming on the mobile will tend to increase. This implies that programming on the mobile shall get expedite. Actually, the future is already here: with just one tap, one can run a program on a cell phone if only we use the extant machinery of the web on mobile, which at the side of the User is currently dominated by **HTML** (HyperText Markup Language), **JS** (JavaScript) and **CSS** (Cascading Style Sheets). HTML is responsible for the basic, static characteristics of a web page, JS adds dynamics and computer power, and CSS enables effective and elegant styling. Our purpose here is to show how to begin with these technologies but focusing on JS as a programming language.

All programs with HTML, JS, and CSS are encoded as plain text so, they can be typed and read with the most rudimentary text processors. All that is needed is to save programs with the correct suffix: **.HTML** for HTML, **.js** for JS and **.css** for CSS.

## HTML TEMPLATE

An HTML document has a basic immutable skeleton:

```
<!DOCTYPE html>
<HTML>

<head>
  <meta charset="utf-8" />
  <title></title>
</head>

<body>
</body>

</html>
```

Exercise. Verify that in the code above:

- For every <, there is a >. Text in within <> is called a **tag**. A tag contains signals for the HTML compiler.
- The tag <!DOCTYPE html> is single and denotes a global directive: the whole document is written in HTML language. This must be specified because HTML is the dominant language for the web but not the unique.
- The tag <meta charset="utf-8" /> is single and specifies that we publish a document in the ordinary encoding of the web. The 8 does not refer to the size of the symbols on a screen but to the memory size in bites that are allocated for each symbol.
- There are paired tags: <html> and </html>, <head> and </head>, <body> and </body>. The tag <html> is the overall cover for most documents, while <head> and <body> define separated sections.
- When a tag is paired, it is open with < > and closed with </>.

HTML is an extensible language because it admits insertions in other languages. Our immediate interest is JS which is the official programming language of the web at the User end. JS itself has many extensions with plugins but because we want to enable offline running of programs, we use pure, basic JS. The following is an HTML template with tags for CCS and JS that work altogether coherently:

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8" />

    <title> </title>

    <style>
    </style>

  </head>

  <body>

    <script type="text/javascript">
    </script>

    <script>
    </script>

  </body>

</html>
```

The former template can be interpreted as follows:

```
<!DOCTYPE html>
<!-- This is a HTML comment. It will not appear in the web page. -->
<!-- Declaration of HTML as the language of this page -->
<html>

<head>
  <!-- Beginning of heading -->
  <meta charset="utf-8" />
  <title>JS PROPAGANDA</title>

  <!-- Insert here the parameters of the used encoding and services,
say, the visibility for mobile and for what operating systems. -->

  <style>
    /* Put here the CSS style directrix of your HTML document, say,
font size. */
/*This is a comment in css */
  </style>

  <!-- All HTML instructions are enclosed by a beginning mark and an
ending mark. Marks are reserved words delimited by < > for the
beginning, and </ > for the ending. -->
  <!-- Example:      <title>JS PROPAGANDA</title>          -->

</head><!-- End of HTML heading -->

<!-- Beginning of programming in HTML and scripts in other languages-->
<body>

  <!-- Put here your HTML instructions -->

  <!-- Inside an HTML document, you can insert code in various
programming languages. So, a declaration of the language is necessary
but for JS it can be dispensed because JS is by 2020 the official
language of the web. -->

  <!-- Beginning of JS code -->
  <script type="text/javascript">
    //This is a JS comment. It will be ignored by the JS interpreter.
    //Put here your JS code
  </script>
  <!-- End of JS code -->

  <script>
    //You can add more JS code here, say, another section of a program.
  </script>

</body><!-- End of HTML programming -->

<!-- End of HTML document -->
</html>
```

## OUR FIRST HTML-CSS-JS PROGRAM FOR THE WEB

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />

  <!-- Title of this HTML document -->
  <title>My first CSS-JS program for the browser </title>

  <style>
    /* This is a comment in css */
    /* For an HTML document, Styling is done in css */
    /* If the background is not not declared, it is white,
       otherwise, it must be given a color */
    body {
      background-color: lightblue;
    }
  </style>
</head>

<body>
  <!-- Prints an important message into the page, a substring appears
in italic -->
  <h1>My first <i>html-css-js program </i> for the browser</h1>
  <!-- A declaration and a message is printed to the page -->
  <p id="demo">Click the button and see the effect here.</p>

  <script>
    //This JS script adds two numbers and reports the result to the
browser
    //declaration
    var a = 1;
    var b = 2;

    //This functions adds two numbers.
    function sum(a, b) {
      return a + b;
    }

    //Function sum(a,b) is called and output written to the browser
    function report() {
      let answer = a.toString() + " + " + b.toString() + " = " +
sum(a, b);
      document.getElementById("demo").innerHTML = answer;
    }
  </script>

  <!-- A button must be clicked on to run the script -->
  <button type="button" onclick="report()">Try it</button>

</body>

</html>
```

To run this program:

- Highlight the code and copy it to the clipboard.
- Paste it to a new document in any word processor.
- Save it with the suffix **.html** as: **MyFirstProgram.html**
- To run the code on Mobile: attach the file to a mail to yourself, open it on mobile and just tap on it. To run it on a desktop: Open it on a browser with the menu **Open file** under **Tools** or **Developer tools** or so. Or, open a new tab, click over it and use **Ctrl + O**. The browser will present a dialog for opening files, next you select the appropriate one, open it, and the program will run at once.

**You are guaranteed** to run this program before 2 hours of work. Otherwise, write to me to negotiate a functional set of instructions.

Exercise. Evolve the previous program to multiply two numbers.

Some word processors are specially targeted for program development. For mobile, you must search for the apps in the store of your system provider. For a desktop, there are various marvelous options: *Visual studio code*, *Atom*, *Bluefish*. All are free. *Sublime* is also free but compulsively ask for a subscription with some cost. Use all these editors for text editing and saving. While you discover the specific short cuts to connect your program with browsers, just save and open your file with the browser of your election and the program will run automatically. For recurrence, refresh the browser tab with F5 and that is all to it.

If one can run and understand this program, one is done because the web is filled in wonderful teaching free material that guides one from zero to a professional level. See below.

## **BUGS AND CHRISTIANITY**

An identified flaw in a computer program is called a **bug**. This naming comes from history:

ENIAC, the first electronic computer, was made of tubes and relays. It was made by the military to simulate ballistic trajectories under real conditions. One morning, the computer did not function. After revising everything, a bug, an insect, was found interfering with a relay. From that day on, every trouble in software is called a bug.

Bugs in self-contained programs have two main sources: misconceptions and ill encoding. You are doomed to commit many errors or bugs before you can make a program run and much more to make it output what it must. Yes: programming is difficult. But, make up your mind because this is a very important part of your formation. So, let us try to understand where the difficulty comes from.

The difficulty of programming stems from three very simple facts:

- There are too many options to make a modification.
- Most modifications render programs useless.
- Minor changes that work produce very diverse effects or otherwise the language would lack enough expressiveness.

Hence, it is extremely improbable to pick up one change that serves your purpose. That is why you must study a lot to succeed, be very careful, and be armed with unbeatable patience.

So, why is a recurrent failure in developing computer programs important for your formation?

Recurrent, unavoidable failure in developing programs is important because it speaks about our origin and destiny, where we come from, and where we go. In fact, DNA contains verbal instructions to continue life and to reproduce. Hence, DNA is a Programming Language whose compiler is the Genetic Code. Besides, by looking to your body, how different your nails from your skin are, and these from your muscles, you will learn that the DNA Language is highly expressive or equivalently that it has too many options to diversify and so you will wonder at the excel purposefulness of every detail of your body. At this moment you will exclaim: all that and more is explained by Evolution! If that scream convinces you that Evolution is the explanation of your being, you have worked with programming thus far very little. Keep working, impose yourself more challenges and your great suffering will rather soon than later lead you to reason like this:

If I, being a very smart person, commit so many bugs at developing simple programs, where are the bugs of evolution that it should mandatorily have committed to synthesizing my DNA programming?

Your left hand is the right place to look at the expected mandatory bugs: open and close it and decide: Is your left hand bugged, malformed, highly dysfunctional? If you find it bugged and representative of the species, report it to the world. This is an incredible discovery that it alone will irrefutably support the Evolutionary Theory. But if you find no bugs neither in your anatomy or physiology nor in the fossil record, then fear God your Creator and repent for you are heading to the abyss if you continue to oppress your wife, your junior staff, and the poor. And pray for your enemies in the name of Jesus.

## **TO KNOW MORE:**

### **HTML, JS and CSS in one place:**

- [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web)
- <https://www.w3schools.com/>
- <https://codepen.io/>
- <https://www.tutorialspoint.com/index.htm>
- <https://www.guru99.com/>
- <https://developers.google.com/web/tools>

## JS:

- <https://javascript.info/>
- <https://eloquentjavascript.net/>

## Running JS heavy-duty tasks (no HTML, no CSS):

<https://nodejs.org/en/>

<https://deno.land/>

## To see and study the code of a wonderful web page:

- In Firefox: After right-clicking the mouse on the page, a new tab will display the code if you select the menu *View Page Source*.
- In Chrome: *View page source* or *Control + U*.
- To see what a specific, self-contained portion of code does: highlight and copy it to the clipboard, and paste it into the following online HTML editor:  
<https://HTMLedit.squarefree.com/>

**Bugs.** Programming is an adventure: you are the chevalier and the bugs are the villains. There is too much epic literature in this respect:

- <https://developer.mozilla.org/en-US/docs/Tools/Debugger>
- <https://www.quora.com/What-is-the-likelihood-of-a-software-developer-creating-0-software-bugs-in-their-work-product-100-of-the-time>
- <https://link.springer.com/article/10.1007/s10664-019-09781-y>
- No one is forced to accept that your programs are correct. Moreover, we consider that debugging, the difficult art of cleansing programs from bugs, must be open and reproducible. That is why we add too many lines of code to our programs whose only purpose is to print the output of functions that these could be compared with expected outputs by everyone.

## DNA and the genetic code:

- <https://www.youtube.com/watch?v=dijqYyFY1GM>
- <https://www.youtube.com/watch?v=a48GfC0ygpq>
- <https://www.youtube.com/watch?v=ocAAkB32Hqs>
- <https://www.youtube.com/watch?v=igVWV8vzxYo>

**Molecular biology and computing.** A DNA molecule is a text that is written in the DNA Programming Language that is transcribed to machine code, the mRNA transcript, that is executed by the CPU, the ribosome, that executes a compiler hash table, the genetic code, that is written in the tRNAs, which are synthesized by the aminoacyl-tRNA synthetases. By looking at the marvelous complexity of all this machinery, my soul exults in God, my Creator, for his overwhelming power and intelligence. The Academy did its best to educate me in its atheistic ways but it was unable to silence my theistic instinct:

- <https://ejristos.com/dl/javaevo2v2.pdf>

### **Confronting the Evolutionary Theory with the Logic of Science:**

- <https://ejristos.com/dl/javaevo3v2main.pdf>
- <https://ejristos.com/dl/javaevol18main.pdf>

**To connect yourself with the True God that will judge you:** He is holy and jealous. So, before everything else, inform accurately about him. Next, clean yourself, and then speak to Him, verbally or in your mind. He will listen. The know-how is in the Bible. The Book of Luke in the New Testament is a good starting point:

- <http://bibletranslation.ws/trans/DRPbiblesofar.pdf>
- <https://www.bible.com/versions>
- <https://www.wordproject.org/bibles/>
- <https://www.biblegateway.com/versions/>

***All links were alive by 31/VIII/2020.***