UNB

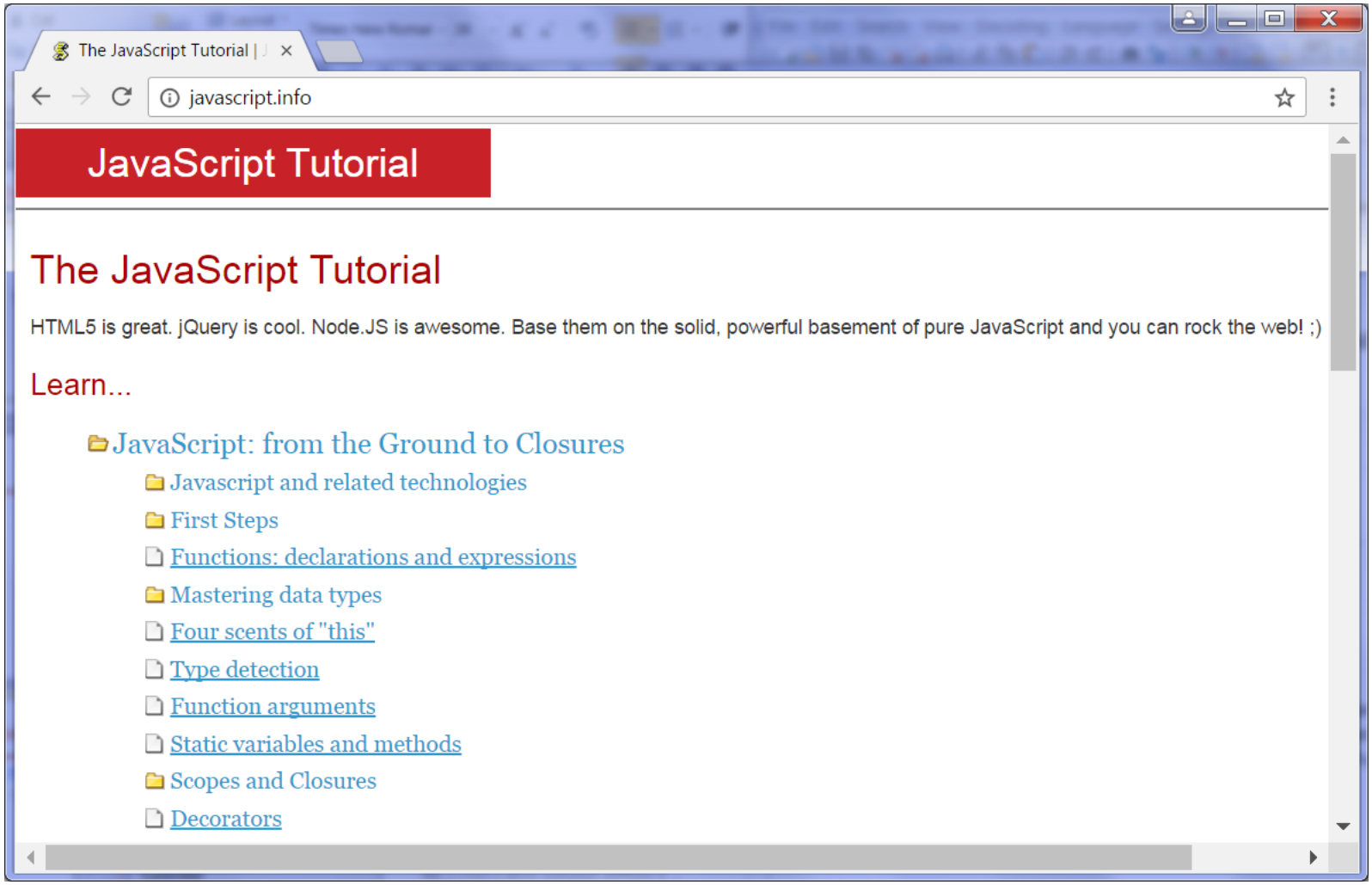
Introduction to JavaScript

Emmanuel Stefanakis
estef@unb.ca

HTML + JS

```
01 <html>
02 <body>
03   <h1>Counting rabbits</h1>
04
05   <script>
06     for(var i=1; i<=3; i++) {
07       alert("Rabbit "+i+" out of the hat!")
08     }
09   </script>
10
11   <h1>...Finished counting</h1>
12
13 </body>
14 </html>
```

Basics (code_1 … 8)



JS -- Where in the HTML ?

Moving scripts into HEAD

If the HTML may be large, where is the best place to put JavaScript? If you want a script to execute early, before the page is displayed, then the HEAD section is a good place.

```
01 <html>
02   <head>
03     <script>
04       function count_rabbits() {
05         for(var i=1; i<=3; i++) {
06           alert("Rabbit "+i+" out of the hat!");
07         }
08       }
09     </script>
10   </head>
11
12   <body>
13
14   <h2>Press the button to start</h2>
15
16     <input type="button" onclick="count_rabbits()" value="Count rabbits!"/>
17
18   </body>
19
20 </html>
```

Putting scripts into HEAD is a common and easy practice, but highly optimized sites use another method.

JS -- Where in the HTML ?

Scripts at the end of `BODY`

A script can also be at the bottom of page body. In this case it executes after the page is shown.

- ⇒ Good, because user doesn't have to wait for scripts.
- ⇒ Bad, because the functions become available after the HTML is loaded. A user has a chance to click on button which may not work. Usually adding special code that hides functionality until the script has loaded resolves the problem.

By the way, CSS styles must be declared in the HEAD according to the HTML standard. Only scripts are allowed to be placed anywhere.

JS -- Where in the HTML ?

External scripts

Usually, most JavaScript code is put into an external file, which is attached to HTML, like this:

```
<script src="/path/to/script.js"></script>
```

The `/path/to/script.js` is a relative path. If you have a specific location including the full URL that is the absolute path. Relative paths are relative to your current location on the site.

File `/path/to/script.js` contains JavaScript code, which will execute immediately after browser recieves the file.

This is very handy, because the same file may be used on many pages. If the web-server is configured correctly, the browser will cache the file and will not download it every time.

Here is how it looks like:

```
01 <html>
02   <head>
03     <script src="/files/tutorial/browser/script/rabbits.js"></script>
04   </head>
05
06   <body>
07     <input type="button" onclick="count_rabbits()" value="Count rabbits!"/>
08   </body>
09
10 </html>
```

Open the code in new window

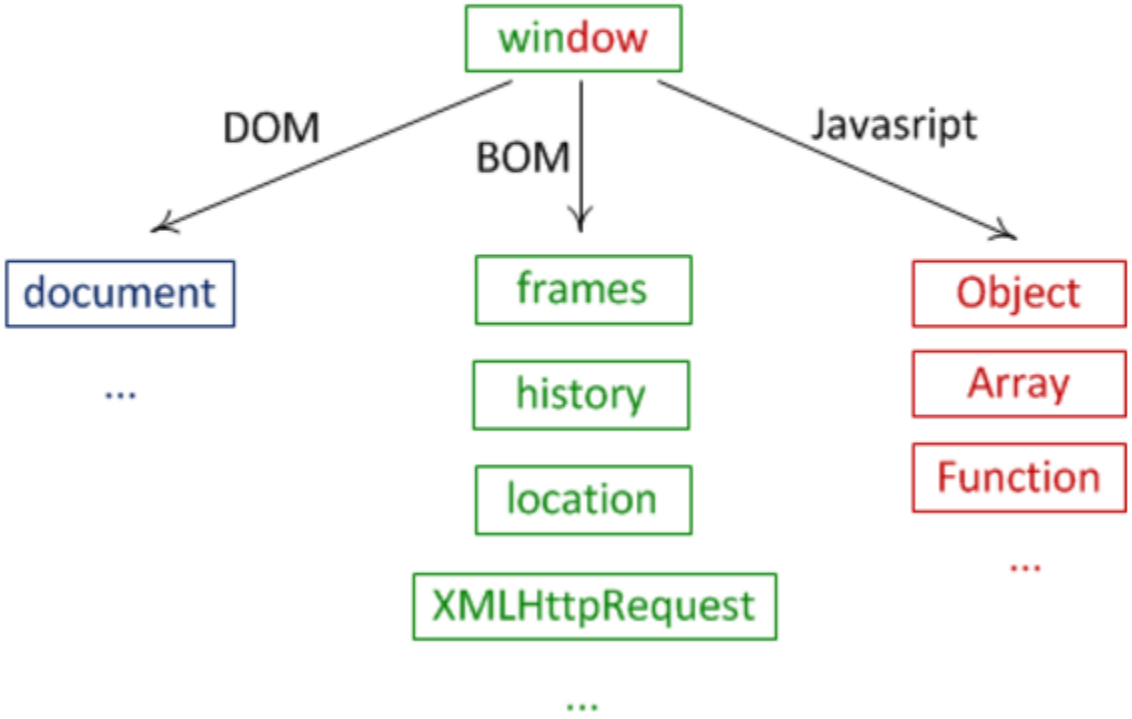
Here is the contents of `/files/tutorial/browser/script/rabbits.js`:

```
1 function count_rabbits() {
2     for(var i=1; i<=3; i++) {
3         // operator + concatenates strings
4         alert("Rabbit "+i+" out of the hat!")
5     }
6 }
```

Browser Environment

The global structure

The browser provides access to a large hierarchy of objects for developers to manipulate. You can see a part of it below:



Browser Environment

Document Object Model (DOM)

`document` and related objects allow to access contents of the page, modify elements etc. Most interaction with HTML is handled here.

There is a pack of standards for DOM, developed by W3C. You can find it at W3C DOM page. There are three levels of DOM, each level expands on the previous. Modern browsers generally support pre-W3C features from the browser dark-ages, called DOM 0.

Browser Object Model (BOM)

BOM is a pack of objects that allow to control the browser, e.g change current URL, access frames, do background requests to server with XMLHttpRequest etc. Functions like `alert`,`confirm`,`prompt` also belong BOM, they are provided by the browser.

Many BOM features are standartized in HTML5, but not all.

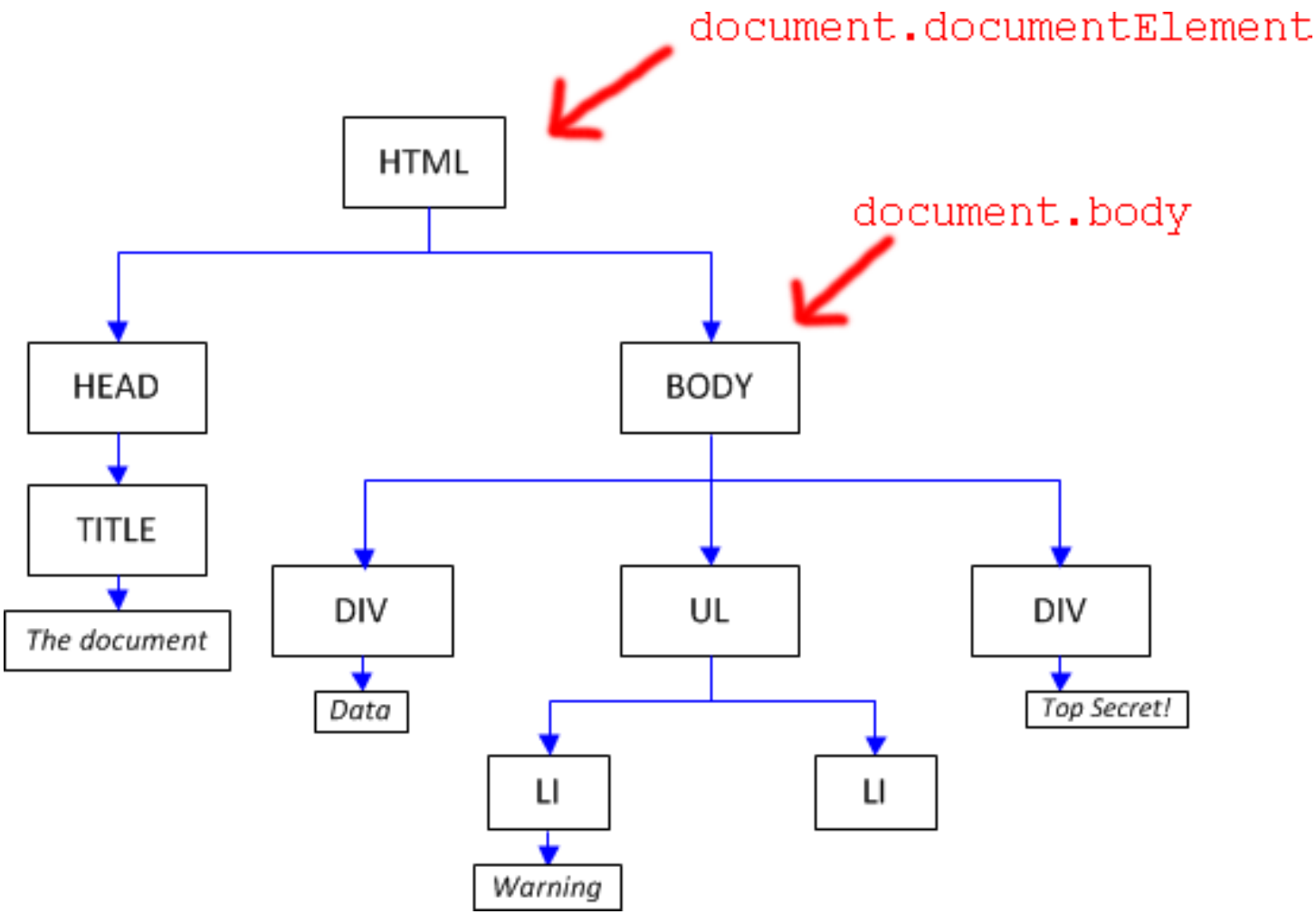
JavaScript objects and functions

JavaScript itself is a language which gives us access to DOM, BOM and provides objects and functions of its own.

JavaScript follows the ECMA-262 standard.

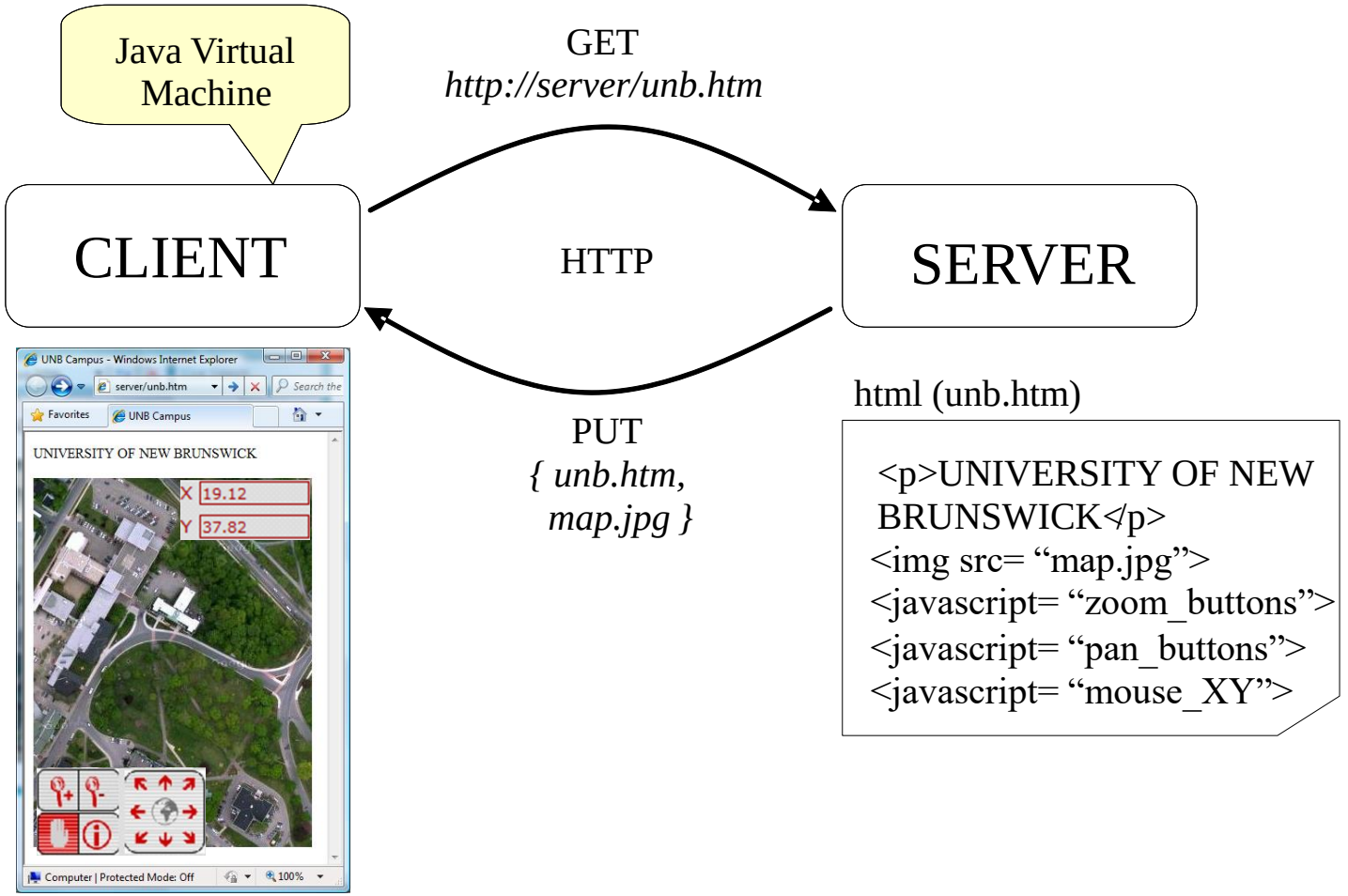
The global `window` object mixes browser window functionality (methods `focus()`,`open()` etc) with being a JavaScript global object. That's why it is both green and red.

DOM

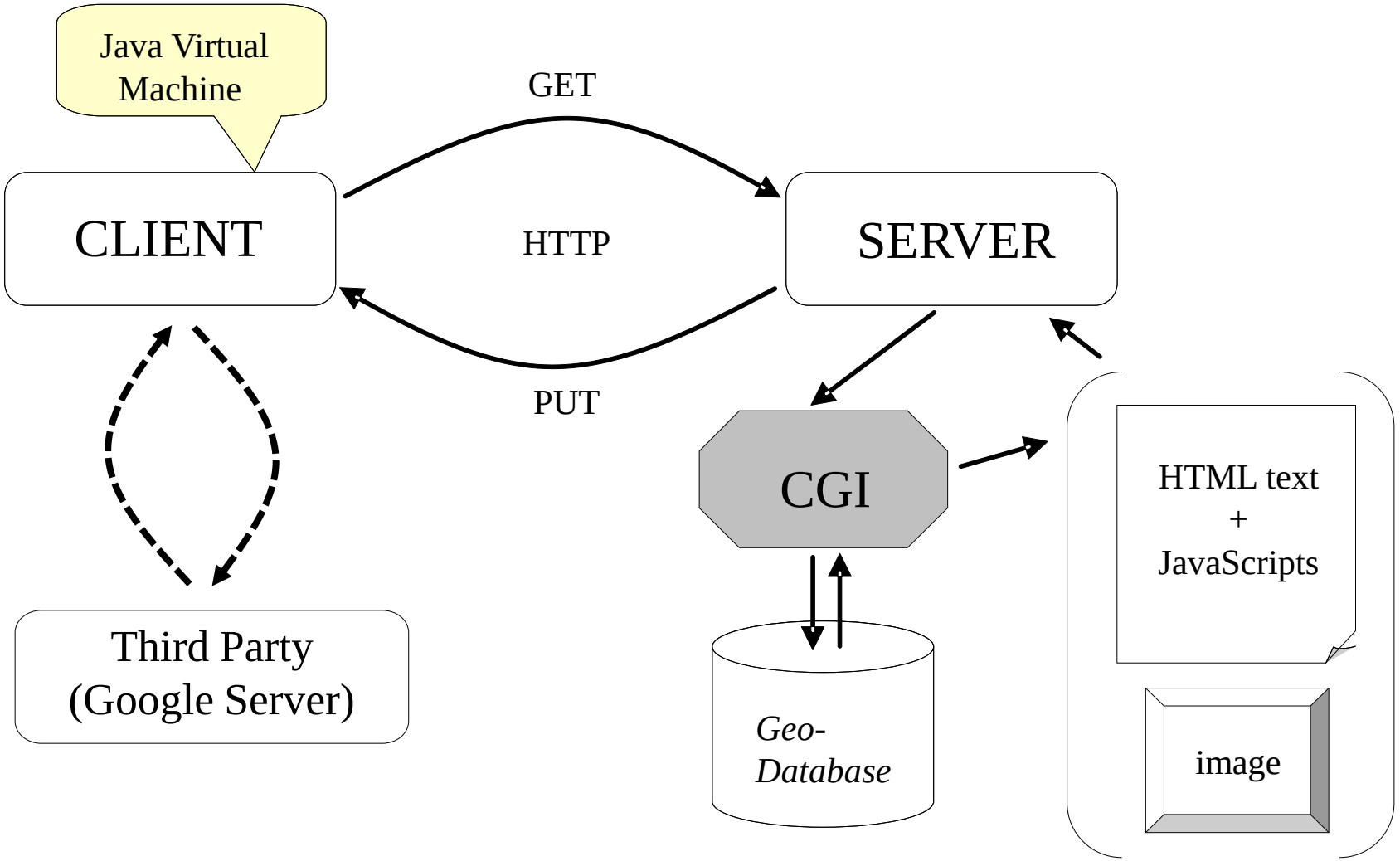

document.documentElement
HTML
document.body
HEAD
BODY
TITLE
DIV
UL
DIV
The document
Data
Top Secret!
LI
LI
Warning

Client-side…

- JavaScripts…



Web-based Mashups



References

- JS Tutorial
  - http://javascript.info/