CS 328 JavaScript Coding Standards so far

- last modified: 2025-05-02

- For CS 328, you are expected to use **unobtrusive-style JavaScript**, trying to keep it as separate from the content as you can by limiting it to within an HTML page's **head** element.

  – In particular, you are expected to put your JavaScript **script** elements (see below) at the **end** of a document's **head** element, after its CSS **link** element(s).

- For CS 328, you are expected to use only the following **two** types of tags for your JavaScript embedded within a document's head element:

```
<script type="text/javascript">
    //  JavaScript statements here
</script>
```

or

```
<script src="file-or-URL" type="text/javascript"> </script>
```

...for **external** JavaScripts.

  – NOTE that, for **external** JavaScripts, it is also USEFUL to add **ONE** of these ADDITIONAL attributes (written using the course-required strict style):

    async="async"

    defer="defer"

    – BOTH of these tell the browser it is SAFE to continue **parsing HTML** while this external JavaScript is being downloaded; (that is, both are assuring the browser you are using good modern practice of NOT modifying the DOM until the page is loaded)

    – **async="async"** means that this external JavaScript can be downloaded at the same time as HTML parsing **and** other JavaScript downloads (for example, it does not depend on anything from a previous JavaScript)

    – **defer="defer"** means that, while it can be downloaded at the same time as HTML parsing, the JavaScripts need to be executed in the order their script elements appear (for example, because one uses a function from an earlier one)

- External JavaScript should be in a file with the suffix **.js**

- For CS 328, you are expected to use **let** or **var** when you begin using a new variable name.

  – Use **let** for block-scoped variables, and **var** for variables "[local] to an entire function regardless of block scope".

  – from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let:

    "**let** allows you to declare variables that are limited to a scope of a block statement, or expression on which it is used, unlike the **var** keyword, which defines a variable globally, or locally to an entire function regardless of block scope. The other difference between **var** and **let** is that the latter is initialized to a value only when a parser evaluates it ...."

- For CS 328, you are expected to follow the JavaScript coding convention mentioned in zyBooks

Chapter 7 to name JavaScript variables using **camelCase** ("where the identifier starts with a lowercase letter, and subsequent words begin with a capital letter").

- That is, `animalSound` is preferred over `AnimalSound` or `animal_sound`.

- BUT: for named **constants** (declared using the keyword `const`), CS 328 class style is to name them in ALL-UPPERCASE.

• For CS 328, we'll follow the good practice recommended by zyBooks Chapter 7 with regard to statements and semicolons:

- You are to **avoid** having two or more statements on the same line.

- JavaScript does not require that statements end with a semicolon (although it requires that they be *separated* by a semicolon if they are on the same line).

  You may choose whether you end your JavaScript statements with semicolons, BUT class style and good practice is to make that decision and then **be consistent** and either use semicolons or not **throughout** your code.

• CS 328 class style is that `break` statements may be used **only** in `switch` statements, and that `continue` statements may **not** be used.