Lesson15

Javascript Basics

Created by Aleksander Apostolidis
PHP Fullstack course, Code Academy 2020

Javascript coding conventions

- Variables and functions should use **camelCase**
- All variable and function/objects names start with a **letter**
- Always declare variable (**var, let, const**)
- Its better if you will end a simple statement with a semicolon “;”
- Always put spaces around operators (= + - * /), and after comma
- Because you are beginners always use { … } for Loops and if/else Conditionals
- Be consistent!

DOM manipulations

Get elements

- querySelector / querySelectorAll
- Example: document.querySelector(‘#id’)/querySelector(‘.class’) / querySelector(‘p’) (get first <p> element
- document.getElementById(‘id’);
- document.getElementByTagName(“HTMLtag”)
- document.getElementsByClassName(“className”)

querySelector
Examples

- var container = document.querySelector(“#test”);
- var matches = container.querySelectorAll(“div.highlighted > p”);
- document.querySelector(“div span”);
- document.querySelector(“input[type=‘text’]”);
- document.querySelector(“button.save”);

Create and append and remove Elements

DOM Manipulations

- Var para = document.createElement(‘p’) - creates html element
- Var div = document.createElement(‘div’) -
- element.appendChild(div) - append/add created element to other element
- Element.remove() and element.removeChild(childElement) - removes element from DOM
- DO NOT USE document.write(“Text”) / document.writeln(“some text/ html”)

Working with elements
DOM Manipulations

- element.textContent = “text” - set text in element
- element.innerHtml = “ Inner html ” - set “html” in element
- element.className = “my-class” - set class for element
- element.setAttribute(name, value) - set element attribute
- element.getAttribute(name) - get element attribute
- element.removeAttribute(name) - get element attribute
- element.style.color = ‘white’ set inline CSS style on element

Switch statements

- Switch statement take a single expression/value as an input and then look through a number of choices until they find one that matches that value, executing the corresponding code that goes along with it.

```
switch (expression) {
  case choice1:
    run this code
    break;

  case choice2:
    run this code instead
    break;

  // include as many cases as you like

  default:
    actually, just run this code
}
```