#Zadatak 11 (Adnan Nukić / adnannukic3@gmail.com)

\* Element

In the initial exercise "Basics HTML and HTML5", the element is:

**<h1> some text </h1>**.

Already on the second task, in the description, the element is only h2, e.g. **h2 element**.

*„There are also h3,h4,h5 and h6 elements”.*

It is noticeable further through the FCC that paragraph, img, main, anchor, input, form, label, div, metadata, etc. are said to be contextual elements, e.g., main element.

Further through the exercise it is possible to notice that the text is an integral part of the element. It is requested to modify the text within the p element. So, it is possible to say that **<p> text </p>** is an element, or only a part of open and closed tags **<p> </p>**.

This last part is noticeable in the section "Nest an Anchor Element within a Paragraf" where the anchor element is said to be **<a>** which requires a closed tag **<a>**.

The "Create a Bulleted Unordered or Ordered List" states that the element is **<ul>,** **<ol**> and **<li>**.

Most often through the FCC element is the **"something" element**.

\*Attribute

The first mention of attributes is in the task of adding img to our HTM, where **src** and **alt** attributes are added to the img element.

Also, for the anchor element we have **href** attribute (HypertextREFerence).

When adding an internal link, the **id** attribute for the element we want to link internally appears.

It is not specified for the attribute whether it is the whole part, e.g. **type = "text"** or just **type**. Namely, there are attributes that do not have the ability to set values ​​such as **required**.

Certain attributes have an input type, e.g. radio, checkbox, etc.

Through the FCC there is no confusion when it comes to attributes.

\*Declaration

The declaration is used for the first time in the FCC when adapting our HTML to a newer version for supporting via browser, where it is necessary to declare doctype.

**<!DOCTYPE html>**

\*Property

A property through FCC is the ability to influence something, for example to change the color of a text via a color still property directly on a particular element (as an attribute).

e.g. <h1 style=”**text-color:** red;”>

Property is actually and font-size in e.g. class:

h1 {**font-size:** 50px;}

\*Selectors

Selectors through the FCC are a way to act on groups or individual elements through CSS. We choose for example some element and edit it for the whole document:

**h1 { color:red;}**

Selectors are also **classes**, **ids,** **elements**, etc.

\*Block

A block is a set of selectors and a set of other commands, functions, ids, etc. within an open and closed tag, e.g. style block:

**<style>**

**h2 {**

**color: red;**

**}**

**</style>**

Through the FCC there is no confusion when it comes to a block.

\*Class

A class is a multi-purpose table that can be added to HTML elements. The CSS class declaration is done in the block <style> ... </style> in the following way:

**.class name {**

font-color: blue;

**}**

In the attribute of the element "class" is not dotted as in the declaration.

e.g. <h1 **class:”class-name”**> ....... <h1>

\*Variable

We use variables for the purpose of simplifying and shortening the time of changing css styles at once for all elements with which we want to change the value, by changing only one value.

**--variable-name:** gray;

Variable must be define inside a selector.

We can use variable as an property value:

e.g. background: **var(--variable-name)**; <!--so there we have example what people in FCC think when they speak about property and its value-->

\*Function

„Use the CSS Transform scale Property to Scale an Element” step leads us to use transform properties with scaling function:

transform: **scale(1.5);** <!-- we see here that they are value for the property-->

Further through the steps, the FCC instructs us that we have different functions for different properties, which perform certain actions on elements through properties.

So functions are some operations, in the background they are mathematical operations, over elements through properties.