JS Course:

* Only one number type in JS (**Number**), also used as a converting function
* Text type in JS is **String**
* Comparison in JS is done using **===** (without type coercion)
* Two none types in JS:
  + **undefined** for unassigned variable
  + **null** for deliberately missing value
* If statement in JS needs parenthesis around condition
* Variable is declared (plus assigned) using **let x = 2**; Reassignment is done without let.
* Const is declared using **const x = 3**
* Variables and constants follow the scope rules, where variable defined within a scope (delimited by {}) is not available outside this scope. This rule is not followed by **var** keyword, therefore it’s not used anymore.
* Unless necessary, all variables should be defined as constants.
* F-string in JS uses template string (backticks quotes) and using **`${2+2}`**
* Case in JS is camelCase for nearly everything, e.g. function and variable names
* Printing in JS is done using console object: **console.log(“Test”)**
* There are multiple objects available in JS code in HTML page:
  + Math
  + JSON
  + Console
  + localStorage
  + document
* **Object** data type in JS is like a **dict** in python and uses keys without quotes. If the key has dash in its name, quotes can be used, such as “key-2”: 2.
* **localStorage object** is used for persisting values even after page reload. It only accepts string key:value pairs, therefore JS object types must be JSON.stringified first (and the JSON.parsed when loaded from storage) before calling **setItem()**
* There are **event listeners** in HTML elements, which can be assigned JS code and expose **event object**, which can be used in the respective JS code to distinguish e.g. which key was pressed (**event.key=”Enter”**):
  + onclick
  + onkeydown
  + onmouseenter
  + …
* **JS code in HTML page** is located in <script> element which can be located in <body> element.
* **Function parameters** can have default vales, but not type hints
* **Default value** can be specified on assignment using OR operator: **let x = value || 0**
* Objects are references to heap memory address similarly to python
* Objects are compared using identity (reference/memory address)
* Ternary operator in JS: **condition ? what-to-do-if-true : what-to-do-if-false**
* HTML elements can be queried as JS objects and saved as JS variable using:

**const** **buttonElement = document.querySelector(“.js-button”)**

* Text within HTML element can be updated using **buttonElement.innerHTML = “text”**
* JS code in <script> element is executed at the load of the webpage, so it can be used (apart from function definitions used in HTML elements) to also initialize variables and content of some HTML elements, such as <p>
* CSS are defined in <style> element in <head>
* CSS can be applied to HTML elements using element’s class attribute or element name. Class attributes (e.g. **<button class = “cost-button”>** can be referenced in CSS style definition using dot notation: **.cost-button** selector:

**<style>**

**.cost-button {**

**border: 10;**

**}**

**</style>**

* CSS styles are applied sequentially, overriding previous definitions if other applicable definitions follow previous ones in a list of definitions. Therefore, more specific CSS (e.g. for specific paragraph) should be defined after more generic ones (e.g. for body element).
* HTML elements (when selected as JS objects) have **classList** property which is also an object whose method **add** can be used (with new class name) to change the appearance of element.
* **Padding is a space inside the element, from text to border**
* **Margin is a space outside the element, to next element**
* Padding is defined
  + with four values in order: top, bottom, left, right,
  + with single value specifying all four values
  + with two values specifying vertical padding and horizontal padding
* Button HTML element can have **img** sub-element defined within button element specifying image that can be clicked. This img element is void, without closing tag. Img element has **src** attribute defining source of the image. **Img** elements can have class attributes to modify its appearance, e.g. height. Also the button element can have class attribute to e.g. set **background color:** **transparent;** and **border**: **3px solid white;** and **border-radius: 10px;**
* **Input** HTML element can have **type** attribute with value date, i.e. <input type=”date”> , which makes input field as date.
* **Div elements do not have default styling.**
* **CSS grid** is a layout that allows table-like display of e.g. <p> elements in rows and columns, with multiple properties to modify appearance.
* **Object destructuring:** const {x, y} = {x: 2, y: 3}