SIT120 Practical

Week 3 –JavaScript Functions

GOALS:

- Start to gain better understanding of the web using basic JavaScript applications.
- Learn definitions of some new concepts/components.
- Note 1-2 sentences reflections for each tasks

Please do not use examples from the web or unit site for your tasks and try to be as creative as possible.

Task 1 – JavaScript String Methods

- String methods help us to work with strings.
- Visit our Unitsite or https://www.w3schools.com/js/ for more information on JS details
- Study examples, implement and use ideas from the web

The length property returns the length of a string: **Example**

```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = txt.length;
```

Task 2 – JavaScript Number and Array Methods (15 mins)

- Number methods help us to work with numbers.
- Visit our **unitsite or** https://www.w3schools.com/js/ for more information on JS details
- Study examples, implement and use ideas from the web

The toString() method returns a number as a string.

All number methods can be used on any type of numbers (literals, variables, or expressions):

Example

```
var x = 123;
x.toString();           // returns 123 from variable x
(123).toString();        // returns 123 from literal 123
(100 + 23).toString();   // returns 123 from expression 100 + 23
```

Task 2 .. – JavaScript Number and Array Methods

- The join() method also joins all array elements into a string.
- It behaves just like toString(), but in addition you can specify the separator:
- **Example**

-
  ```
  var fruits = ["Banana", "Orange", "Apple", "Mango"];
  document.getElementById("demo").innerHTML = fruits.join(" * ");
  ```
- Result:
- Banana * Orange * Apple * Mango

Task 3 – JavaScript Get /Set Methods

- **The getTime() Method**
- The getTime() method returns the number of milliseconds since January 1, 1970:
- **Example**
- var d = new Date();
document.getElementById("demo").innerHTML = d.getTime();
- **The getFullYear() Method**
- The getFullYear() method returns the year of a date as a four digit number:
- **Example**
- var d = new Date();
document.getElementById("demo").innerHTML = d.getFullYear();

Study examples, implementations and use ideas from the web

Task 4 – Get started with some new concepts/components

Important theories: We will spend around 30 minutes in this task

- □ **Computed Properties and Watchers**
- □ **Class and Style Bindings**
- □ **Conditional Rendering**
- □ **List Rendering**
- □ **Event Handling**
- □ **Form Input Bindings**
- □ **Components Basics**

Click and try to understand their definitions. Why we need these new concepts? Let's write one sentence each. For example

A **computed property** is the concept in Vue used to implement complex logic and operations, as embedding substantial logic into the templates makes them bloated and hard to maintain.

- □ Component Registration
- □ Props
- □ Custom Events
- □ Slots
- □ Dynamic & Async Components
- □ Handling Edge Cases

We will be learning how to code and apply these concepts/components in the upcoming weeks.

Continue from Week 2
Create one responsive page (Proof of Concept for Assessment 1)

- Create a responsive web page with the following requirements
  - It should be a business idea like online shopping, services, etc.
  - Your application should contain HTML hyperlinks between the pages, provide necessary information about the web page.
  - It should have a search function, query function as well as contact information.
- Use advanced HTML, CSS and JavaScript features properly.
- Create a page and discuss different requirements with your demonstrator

Hint: https://vuejs.org/v2/examples

Discuss your Assignment -1 with tutor and peers

- Discuss (think how to implement) the current state of your project proposal , user stories and UX design with your tutor and classmates. Note down insightful ideas and revise it.

Important for next week: JavaScript Debugging

Errors can (will) happen, every time we write some new code.

Code Debugging

Searching for (and fixing) errors in programming code is called code debugging.

- Coding might contain syntax errors, or logical errors.

Many of these errors are difficult to diagnose.

Often, when programming code contains errors, nothing will happen. There are no error messages, and you will get no indications where to search for errors.

JavaScript Debuggers

- Debugging is challenging and sometimes intractable. But fortunately, all modern browsers have a built-in JavaScript debugger.

Study a few examples, implement and use ideas from the web,
we will study in details next week