# Estructura de la clase 5:

# Primera Parte

## Preguntas de ejercicios pasados.

* Revisar temas de la clase anterior

## LocalStorage

Realizar ejemplo de crear un objeto super héroe.

* window.localS*t*orage - stores data with no expiration date
* window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)
* localStorage.setItem, get item, clear.
* tostring / parseJson. // no se pasan los métodos

Diferencia con las cookies:

<https://www.w3schools.com/js/js_cookies.asp>

* From object – delete -> Preguntar quién averiguó sobre el tema.

Hacer ejercicio – 8 de arrays para ir retomando la clase.

## Delete elements

* From array – splice ->

## Modular

* IIFE­, Encapsulamiento – Scope
* Patron Modular – Crean do una función anónima que se autoinvoca y retorna un objeto literal.
* Metodos privados y públicos.
* Function declaration
* Ejercicio modulo diario

**Ver modulo de netflix. Ejercicio 8**

Tp Crear modulo Diario: TP patrones.

# Segunda Parte

## *SCRUM*

* Introduccion
* Roles: ProductOwner, scrumMaster, Equipo de desarrollo.
* Actividades:
  + Utilizar ejemplo de Volver al futuro.
  + Selección de requisitos
  + planificación de la iteración (Sin el cliente)
  + Ejecución de la iteración (Reuniones diarias) y no se pierde el contacto con el PO.
  + Inspección y adaptación -> Demostración y restrospectiva

## Comienzos de DOM

* Que es el dom?
* Explicacion de manipulación de elementos
* Tareas habituales con el dom en js.
* Arbol de nodos. Familia de nodos: Padres, hijos, hermanos.
* Mostrar la consola y una pagina moviéndonos por los distintos elementos.

## Accesso

* Getelementby id
* GetElementbytagName
* GetElementsbyclassname
* Array de nodos.

## Propiedades de los nodos

* .innerHTML

Propiedades de los

Creacion de nodos

* document.createElement(‘p’) ->**createElement(etiqueta)**:
* document.createTextNode(“Hola Mundo!”);**createTextNode(contenido)**:
* parrafo.appendChild. **nodoPadre.appendChild(nodoHijo)**:
* removeChild
* .setAttribute("class", "democlass");
* varparrafo = document.getElementById("provisional");
* parrafo.parentNode.removeChild(parrafo);
* <p id="provisional">...</p>

**Ejercicio del payaso**

**Ejercicio Clase 7:**

**TP Clase 7.**

## Eventos

* Hacen posible que los usuarios transmitan información a los programas.
* Multiples eventos: click, mover el mouse, presionar una tecla. Al ocurrir estos eventos, se ejecuta una función asociada a la que de llama eventHandler (Manejadores de eventos).
* Algunos eventos están ligados a una etiqueta. Otros se pueden aplicar a todos los elementos.

## http://i.imgur.com/8F7hhnl.jpg

## Formas de agregar un eventhandler.

* Como atributos de html // ej: onclick o onmuseover .(FEO)
* Como funciones externas ->onclick -> mostrar mensaje. (this). Style.bordercolor (FEO)
* Manejadores de eventos semánticos- > .onclick = function () {
* Se tiene que esperar a que termine de cargar la pagina.
* Si los llamo por clase o tag tengo que hacer un ciclo.

EJEMPLOS CON EVENTOSSSS

Ej: 1, 2, 3 . Nombrar onload, mouse y resize

Ejercicio y TP.