# Estructura de la clase 6:

# Primera Parte

## Repaso general

* Temas que quedan: Programación funcional, Ajax, proyecto final.

## Identificación de componentes en páginas

* Acordión
* <http://www.murphyoilsoap.com/home>

## Ejercitación jQuery

* Hacer un formulario con tooltip – Ej jquery – ej 6
  + Mouseenter, mouseleave, mouseover, mouseout. (on y multiples eventos)
  + Hover: $(selector).hover(handlerIn, handlerOut) is shorthand for: $(selector).mouseenter(handlerIn).mouseleave(handlerOut);
* TP11: Curriculum Vitae
  + each, formulario, fade in, fadeOut (Callback)
  + seleccionar cada campo
  + data-attr , apendear información
  + Botón de imprimir.
* Ejercicio del teléfono.
  + .html, utilities
  + Renderizar contactos
  + Agregar un contacto a la base
  + Recuperar datos del local storage
* Ejercio 6 - Autos
  + Componentes varios
  + Ver Each
  + Change
* Ejercicio 7 - Can’t touch me
  + Diferencia entre posición y offset.

**TP13: Blog**

* Ejercicio 8 – center element y manejo de imágenes
* Ejercicio 9 –Make a read More

## Implementación de librerías de Jquery

* nicescroll
* BxSlider

**TP14: Ajax**

Ejercicio 8 – Hacer un submenú.

Stop propagation; usar Firefox 3d view.

## Effectos jQuery

* Stop
* Bind y unBind.
* toggleClass / slideToggle

Implementacion de librerías externas de jQuery

Evento on resize. Onbreakpoint change

Posibles implementaciones:

CenterElement

ReadMore

Manejo de imágenes con load, cargar imagen dependiendo del tamaño de la pantalla.

Ejemplo banderas de países. Con Ajax.

Ajax- > traer jSon

Post get

Hash

window.location

Window.hash.

Window.href

# jQuery utilities.

Dar TP12 de Tarea

# Segunda Parte

## Programación Funcional

Nuevo patrón de diseño en el que el programa es considerado como una función matemática. Solo tenemos una entrada y una salida.

**Diferencia entre los valores por referencia**

**Por valor**

El paso de parámetros por valor consiste en copiar el contenido de la variable que queremos pasar en otra dentro del ámbito local de la subrutina , consiste pues en copiar el contenido de la [memoria](https://es.wikipedia.org/wiki/Memoria_principal) del argumento que se quiere pasar a otra [dirección de memoria](https://es.wikipedia.org/wiki/Direcci%C3%B3n_de_memoria), correspondiente al argumento dentro del ámbito de dicha subrutina. Se tendrán dos valores duplicados e independientes, con lo que la modificación de uno no afecta al otro.

**Por referencia**

El paso de parámetros por referencia consiste en proporcionar a la subrutina a la que se le quiere pasar el argumento la [dirección de memoria](https://es.wikipedia.org/wiki/Direcci%C3%B3n_de_memoria) del dato. En este caso se tiene un único valor referenciado (o apuntado) desde dos puntos diferentes, el programa principal y la subrutina a la que se le pasa el argumento, por lo que cualquier acción sobre el parámetro se realiza sobre la misma posición de memoria.

Algunos tipos de variables se pasan por valor y otros por referencias:

* **Por valor:** string, boolean, number , null,undefined
* **Por referencia:** Arrays, Objects y funciones.

***Realizar ejercicio 1 de Programación funcional***

**Funciones puras**

Una función pura es aquella que no tiene efectos secundarios. Significa que no CAMBIA ningún dato externo, pero si puede acceder. Se hace encapsulado el universo entero de la función (Que no se modifiquen variables globales o de otro scope). La idea de programación funcional es tratar de hacer todas las funciones puras. Todas la funciones puras deben retornar algo, un array o un objeto ya que solo podes retornar un valor.

**Inmutabilidad**

Una variable inmutable significa que sus valores son inmutables. En la programación funcional se utiliza el método de la inmutabilidad, es decir que no se cambie el valor de las variables.

Se podría asegurar que no cambie utilizando const

const pepe = 20;

pepe++ -> no puedo hacer esto

const no quiere decir que tenga VALORES inmutables, sino que es una AGINACION inmutable (o inmutable binding). Primitive strings y numbers son inmutables. no se los puede cambiar. Usar const es una eleccion. Elegímos no cambiar la referencia.

const z = [4,5,6];

z = 10; no se puede

z[0] = 8 si se puede porque estoy apuntando al mismo objeto.

***Realizar ejercicio 2 de Programación funcional***

Operación en listas de Elementos:

**Map (List transformation) Transformaciones de listas:** A partir del método Map de los arrays obtendremos una nueva lista con la misma cantidad de valores pero con una transformación aplicada.

**Foreach(List iteración, solamente iteración:** No cambio nada? La única forma que tendría sentido es si tuviera un side effect. Por eso te dicen que no las uses, pero a veces quiero map effects. También podría iterar con un map.

***Explicaren pizarrón. A que se llama mapear***

***Realizar ejercicio 3 y 4;***