|  |  |
| --- | --- |
| Tarea:  Estándares de HTM, CSS, Javascript, Angular Filtros & Directivas.  Miembros:  Natalie Mata, Ariana Aguilar, José Guillén, Germán García, Adrián Carmiol. |  |

Estándares de HTML:

**Atributos y etiquetas:**

Todas las etiquetas y atributos deben escribirse en minúsculas. Además, los valores de atributos deben estar en minúscula cuando el propósito es para que solo sea interpretada por máquinas. Para los casos en los que los datos tienen que ser legible por humanos, debe hacerse la capitalización apropiada.

**Indentación:**

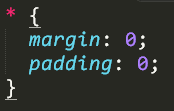
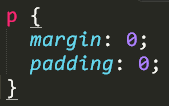
Se identará con la tecla tab preestablecida en 4 espacios.

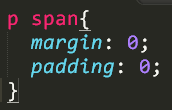
**Etiquetas con auto-cierre:**

Para las etiquetas que son auto-cierre, debe tener un espacio antes del slash

Estándares de CSS:

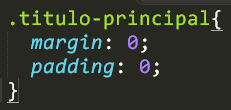
**Selectores:**

* Selector universal:
  + Llama a todos elementos de una página.  
    
* Selector tipo etiqueta:
  + Llama a los elementos según la etiqueta HTML mencionada.  
    
* Selector descendente:
  + Llama a los elementos dentro de otros elementos.



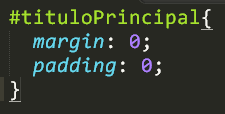
* Selector de clase:
  + Se declaran con un “.” al inicio, y le da estilo al elemento cuyo atributo “class” sea igual al declarado.





* Selector de ID:
  + Permite seleccionar un elemento del HTML según su atributo “id”, se declaran con ”#”.





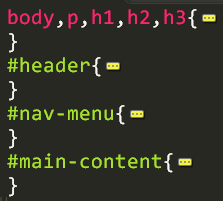
**Nombres de clases:**

Mantener el código legible, por lo cual darle nombres con sentido a las clases, también esto ayudará a comprender la función de estas.



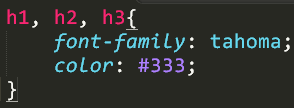
**Estructura:**

Se recomienda un formato de arriba hacia abajo, esto para que los estilos se parezcan al orden del código fuente, esto ayudará a identificar rápidamente partes de tu código.



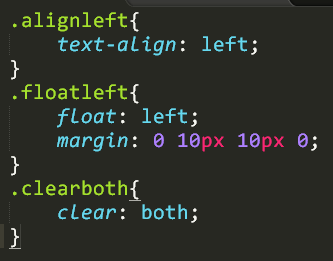
**Clases combinadas:**

Cuando algunos elementos utilizan las mismas propiedades, en vez de volver a escribir el mismo código, se pueden combinar y declararle las mismas propiedades.



**Clases genéricas:**

Se utilizan para alinear contenidos, flotar imágenes, limpiar los “floats”, y otras clases que se puedan tener en mente.



Estándares de JavaScript:

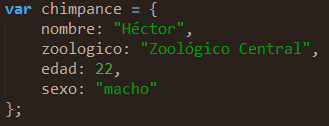
**Nombre de las variables:**

Los nombres de las variables se escribirán en **camelCase**, lo cual consiste en poner la primera palabra en minúscula y la segunda con la primera letra en mayúscula, si existiera una tercera palabra, también con la primera letra en mayúscula.

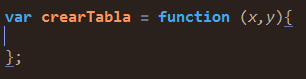
**Espacios entre los operadores:**

Siempre poner un espacio entre operadores ( = + - \* / ) y después de las comas.

**Reglas para las declaraciones:**

* Declaraciones simples:
  + Terminar siempre una declaración simple con un punto y coma.  
    
* Declaraciones complejas:
  + Poner el corchete de apertura al final de la primera línea.
  + Poner un espacio antes del corchete de apertura.
  + Poner el corchete de cierre en una nueva línea sin espacios antes de éste.
  + No terminar la declaración compleja con punto y coma.  
    

**Creación de funciones:**

Para crear una función se va a declarar como variable y se le asigna la función.  


**Indentación:**

Se indentará con la tecla tab preestablecida en 4 espacios.

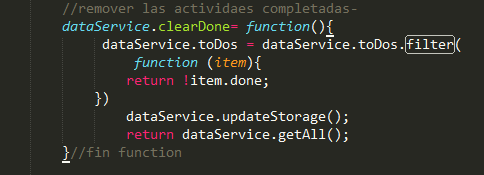
Filtros

**En nuestro código de la tarea anterior podríamos aplicar los siguientes filtros:**

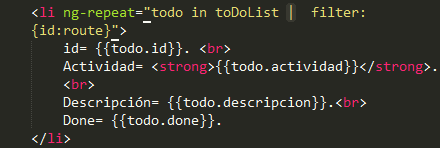
* Date = Para mostrar un formato mejor de fecha {{fecha | date}}
* UpperCase = Para eventualmente mostrar los títulos de las actividades en mayúscula {{hola | uppercase}}
* LimitTo = Para que no se hagan listas muy largas de tareas puede que después de unas 25 tareas se oculten el resto {{lista | limitTo:25}}

**Por otra parte se habían implementado los siguientes filtros:**

* Filtro para obtener solamente las actividades que aún no se habían completado.

x

* Filtro para obtener la actividad indicada en el id.



Directivas

**ng-repeat**

Esta directiva permite repetir una serie de tags HTML en función de un arreglo de datos de $scope. Su utilidad es para hacer tablas o similar, es decir para estructuras que se repiten.

**ng-submit**

Permite el binding de una expresion angular a eventos onsubmit.

**ng-href**

**<a ng-href="http://www.gravatar.com/avatar/{{hash}}">link1</a>**

**ng-class**

Esta directiva permite configurar las clases html de css al unir a la expresion que representa.

**ng-model**

Esta directiva une un input, select, textarea con alguna propiedad en el scope utilizando ngmodelcontroller, el cual es creado y presentado con esta directiva.

**ng-click**

La directiva ngClick permite especificar un comportamiento especifico de un elemento al hacerle click

Referencias adicionales:

http://codeguide.co/