Bienvenido al test de evaluación de chicfy HTML, CSS y JAVASCRIPT.

Por favor, intenta responder a todas las preguntas sin extenderte demasiado. Algunas preguntas pueden ser respondidas googleando, pero intenta responder personalmente.

1. **¿Qué frameworks Javascript has usado? ¿Qué destacarías de cada uno de ellos?**He usado angular, angualr2, react y jQuery  
     
   De angualr destaco la idea de llavar al front MVC que era una idea muy usada en back pero no mucho en front. Los servicios, las factories, las directives, los filtros.  
   Esto hace que se separen bien los modules html quede muy limpio, que se puedan testear los controladoes y los servicios por separado.  
   La verdad es que está muy bien  
     
   De angular2, pues que decir es una puta pasada (perdón) con el typescript los decoradores queda todo mucho más lipio y se mueven servicios y demás y ahora todos son classes.  
     
   jQuery lo he incluido aun que no se mucho un framework, es más bien una librería. Destacaría la manipulación del dom.

En general todos los frameworks de front han hecho que el back se pueda mover a una APIRest quitando todo el código de “back” en las capas de presentación

**2. ¿Cómo crearías una clase en Javascript? Pon el ejemplo que quieras.**

En javascript plano no hay “tipos” por lo tanto todos son objetos no “existe” mucho la idea de clase. Apartir de EC6 si aparecen los tipos

**En javascript**

**var chicfyClass = {**

**}**

**En EC6**

class chicfy {

hola: string;

}

**3. Hay un error Javascript en una de las webs, ¿qué pasos darías para solucionarlo?**

Primero mirar la consola para ver el detalle del error.  
Escrbir el test unitario o e2e apartir del erorr detectado en consola.  
Una vez escrito el test corregir el error, pasar los testos y comit a dev.

**4. ¿Qué tecnologías de las propues****tas e incluidas en HTML5 has usado?**

He utlizado los tags semánticos habituales como <header> <aside> <content> etc.  
Canvas, figure, data , audio, video

**5. ¿Qué tecnologías y/o herramientas has usado para crear/generar/optimizar estilos CSS? Cita ventajas e inconvenientes de cada una.**

He utilizado precompiladores de css less y sass.  
Normalmetne me gusta más less por que quita la dependencia en todos los entornos hay javascript.

**6. ¿Has usado tests unitarios y/o de secuencias de usuario? ¿Conoces algún framework? ¿Cuáles son los principales beneficios de usar tests?**

Si he usado tests unitarios la mayoría de ellos escritos en Jasmine o mocha.  
Para lanzarlos normalmente he usado karma o Jenkins.  
En cuanto a beneficios de usar los tests unitarios. Voy a citar lo mismo que he escrito para el back, y es que si tienes una buena suite de testos te da una tranquilidad al desarrollar. Si rompes algo y la suite de tests es correcta puedes detectar el bug antes de subir a producción.

**7. Indica el resultado de la ejecución del siguiente código javascript y la razón.**

var fullname = 'John Doe';

var obj = {

fullname: 'Colin Ihrig',

prop: {

fullname: 'Aurelio De Rosa',

getFullname: function() {

return this.fullname;

}

}

};

console.log(obj.prop.getFullname());

var test = obj.prop.getFullname;

console.log(test());

Resultado  
Aurelio De Rosa  
John Doe  
  
El primer console saca “Aurelio De Rosa” por que estás llamando a la función getFullname() y el scope del this es el obj.  
  
Y el segundo console saca “John Doe” por que getFullName hace un return de this.fullname y aquí el scope de this no es obj sino es window.

**8. Ejercicio Javascript**

Escribe una función spacify que, dada una cadena de texto, devuelva la misma cadena con cada carácter separado por un espacio. Ejemplo: spacify(“chicfy mola”) devuelve “c h i c f y m o l a”.

Punto adicional: Escribe el código necesario para hacer que el método spacify sea parte String.

function spacify(aString) {

return aString.split("").join(" ");

}

**9. Ejercicio CSS**

Imagina que tienes el siguiente código HTML:

<div id=“container”>

<div id=“content”>Chicfy. Compra, vende y diviértete.</div>

</div>

Si la capa “container” mide 200px de alto, indica todas las formas que sepas de poder **centrar verticalmente** la capa “content” dentro de “container”. Escribe el código CSS necesario para cada forma.

#container {

height : 200px;  
}

#content {

position: absolute;

height: 100px  
top: 50px;

}

#container {

height : 200px;  
}

#content {

position: absolute;

height: 100px  
bottom: 50px;

}