esolutions

Test de Nivelación

1. ¿Cómo funciona `this` en JavaScript?

   Respuesta:

   Con la palabra `This` podemos acceder a las propiedades y claves de un objeto, en algunos casos hace referencia al objeto global. En el caso de utilizarlo en una función, va a depender como lo invoque la función

2. ¿Cuál es la diferencia entre variables `null`, `undefined` y `undeclared`?

   Respuesta:
   En el caso de `null`, la variable existe, pero representa valor nulo o está vacío. En el caso de `undefined` la variable no existe, representa que el valor o variable no está definida. En el caso de undeclared, no está declarado.

3. ¿Qué es un «*clousure*» y cuándo o por qué usaría uno?

   Respuesta:

   Un closure es una función que encapsula una serie de variables y definiciones que únicamente serán accesibles si se devuelve con return.

   Aplicando closures podemos crear funciones utilizando diferentes contextos, por ejemplo en el ejercicio práctico de ordenamiento de menor a mayor por edad, y luego por nombre se utilizan.

```
ejercicio.sort( (a, b) => {
    if(a.edad < b.edad) {
      return -1;
    }
    if(a.edad > b.edad) {
      return 1;
    }
    if (a.nombre.toLowerCase() < b.nombre.toLowerCase()) {
      return -1;
    }
    if (a.nombre.toLowerCase() > b.nombre.toLowerCase()) {
      return 1;
    }
    return 0;
  });
```

4. Escriba un programa en JavaScript que dada la entrada (por ejemplo):

```
let input = [
    {nombre: "Pedro", edad: 20, ciudad: "Córdoba" },
    {nombre: "Paricia", edad: 22, ciudad: "Córdoba" },
    {nombre: "José", edad: 23, ciudad: "Mendoza" },
    {nombre: "María", edad: 20, ciudad: "Córdoba" },
    {nombre: "Juan", edad: 20, ciudad: "Córdoba" },
    {nombre: "Ana", edad: 22, ciudad: "Córdoba" },
];
```

El valor de la propiedad «nombre» de cada objeto debe extraerse, sólo en el caso que el valor de la propiedad «ciudad» sea «Córdoba» y almacenarse en un array ordenado por edad (de menor a mayor) y por orden alfabético.

Es decir, la salida debe ser:

`["Juan", "María", "Pedro", "Ana", "Patricia"]`

Respuesta:

```
// Matias G. Soruco
// Ejercicio técnico ESolutions


let ejercicio = [
    {nombre: "Pedro", edad: 20, ciudad: "Cordoba"},
    {nombre: "Patricia", edad: 22, ciudad: "Cordoba"},
    {nombre: "José", edad: 23, ciudad: "Mendoza"},
    {nombre: "Maria", edad: 20, ciudad: "Cordoba"},
    {nombre: "Juan", edad: 20, ciudad: "Cordoba"},
    {nombre: "Ana", edad: 22, ciudad: "Cordoba"},
];

// Utilice metodo sort para ordenar primero por edad, luego por el nombre
// Si A < B, devolver -1, si A > B devolver 1
// Si la edad no es ni mayor ni menor (es igual) entonces lo que hacemos es compArar el nombre de las personas alfabeticamente
ejercicio.sort( (a, b) => {
    if(a.edad < b.edad) {
      return -1;
    }
    if(a.edad > b.edad) {
      return 1;
    }
    if (a.nombre.toLowerCase() < b.nombre.toLowerCase()) {
      return -1;
    }
    if (a.nombre.toLowerCase() > b.nombre.toLowerCase()) {
      return 1;
    }
    return 0;
  });

// Creamos el array donde van a estar las personas que cumplan con ciertas condiciones
var newArray = [];

ejercicio.forEach(object =>{

    var nombre = object.nombre;
    var ciudad = object.ciudad;
// Si la persona es de Cordoba, se pushea al array
    if(ciudad === "Cordoba"){
        newArray.push(nombre);
    }
});

// Devuelve el array ordenado por edad, y luego alfabeticamente
console.log(newArray);
```

Link [Repositorio] para acceder al codigo

5. ¿Qué función cumple el `doctype` y cuántos tipos conoce?

Respuesta:

`doctype` es una declaración del tipo de documento, especificamos el lenguaje y la versión del mismo. Debe incluirse en la primera línea de la pagina, incluso antes del HTML, por ejemplo, ya que definiremos el estándar que trabajaremos.
Ejemplo:

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ESolutions</title>
</head>
<body>

</body>
</html>
```

Actualmente solo trabaje con `doctype` HTML.

6. ¿Cuál es la diferencia entre `HTML` y `XHTML`?

   Respuesta:

   Yo creo que la mayor diferencia entre HTML y XHTML esta en la sintaxis, XHTML es como mas “estricto”, como por decir una diferencia: HTML no distingue entre mayúsculas y minúsculas, mientras que XHTML distingue entre ambas, y los atributos y las etiquetas deben estar en minúsculas.

   Por ejemplo:

   Este seria valido en HTML y XHTML:

```html
<p class="esolutions">Estoy realizando la prueba técnica!</p>
```

Pero esto no seria valido en XHTML, pero si en HTML (ya que contiene mayúscula en ‘eSolution’:

```html
<p class="eSolutions">Estoy realizando la prueba técnica!</p>
```

7. Generalmente, ¿por qué es buena idea agregar la etiqueta `<link>` dentro de la etiqueta `<head>` y la etiqueta `<script>` justo antes de cerrar la etiqueta `<body>`?

   Respuesta:

   Primero recordemos que tienen un comportamiento en cascada. Significa que por ejemplo, si tenemos dos definiciones/propiedades para un valor, tomará el ultimo que se declaró.

   En el caso de la etiqueta `<link>`, debemos poner en el `<head>` porque es donde generalmente cargaremos los estilos a utilizar (css, fonts, icons) y los necesitamos funcionando antes de que nuestra página inicie su contenido

   En algunos casos, la etiqueta `<script>` es buena idea ponerlo al final de `<body>` porque de otra manera podría interrumpir lo que seria la carga de algunos datos importantes de nuestra web. Pero en otros casos, necesitamos que estén dentro de la

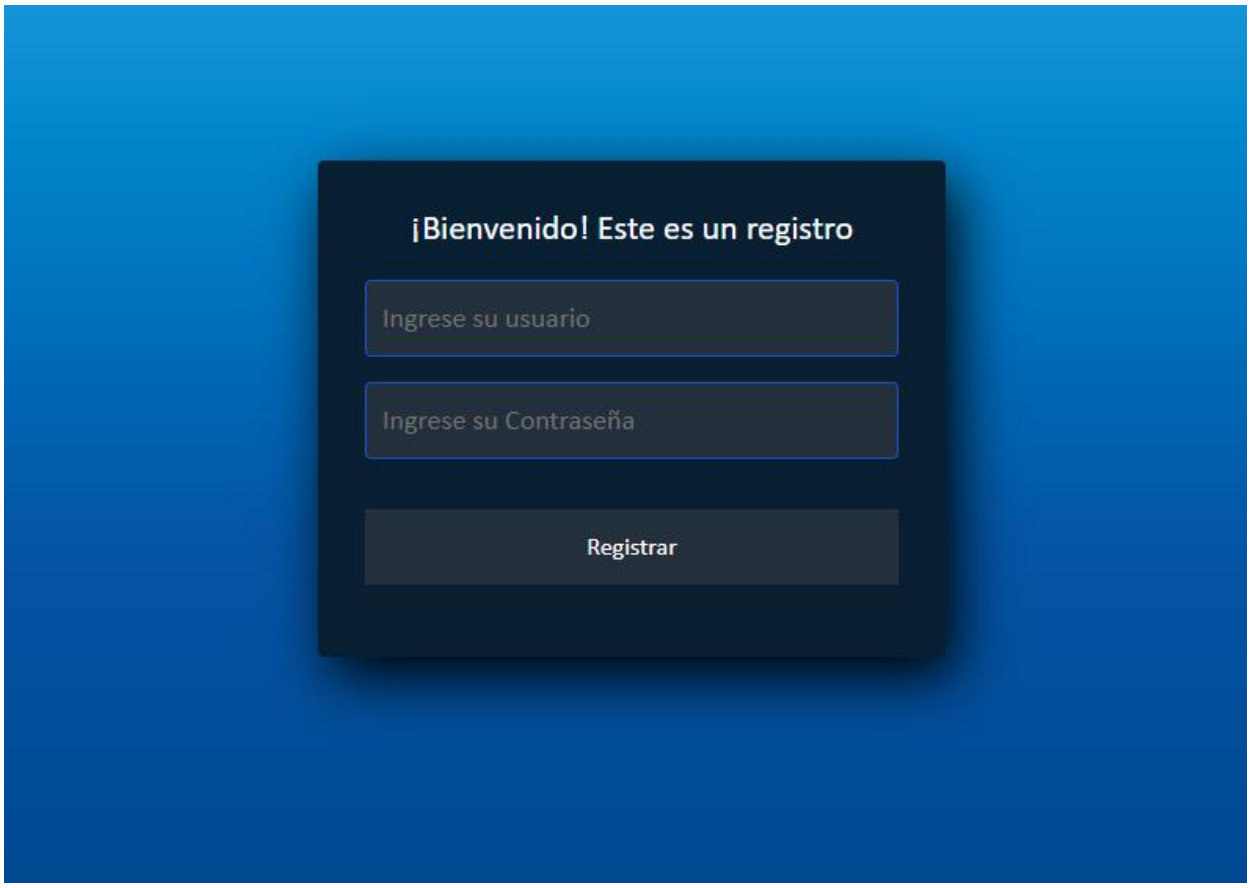
etiqueta head por si debemos utilizar/modificar DOM, por ejemplo

8. Cree una página con dos preguntas propias. Su encuesta debe primero recopilar los datos personales de los participantes (nombre, dirección, sexo, fecha de nacimiento, una contraseña, etc.) seguida de cuatro preguntas para responder. Se valora la variedad de elementos usados en la página. Use CSS para mejorar el aspecto de su formulario.

   Respuesta:

   Cree un simple formulario, solo con HTML y CSS.

   Donde su index.html sería



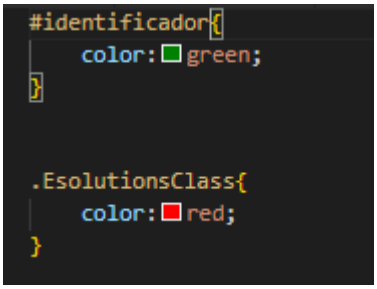
   Y su indexTec.html es una single page donde contiene una presentación y formulario

   Link Git para acceder a la pagina o Repositorio para acceder al código

9. ¿Cuál es la diferencia entre clases e identificadores en CSS?

   Respuesta:

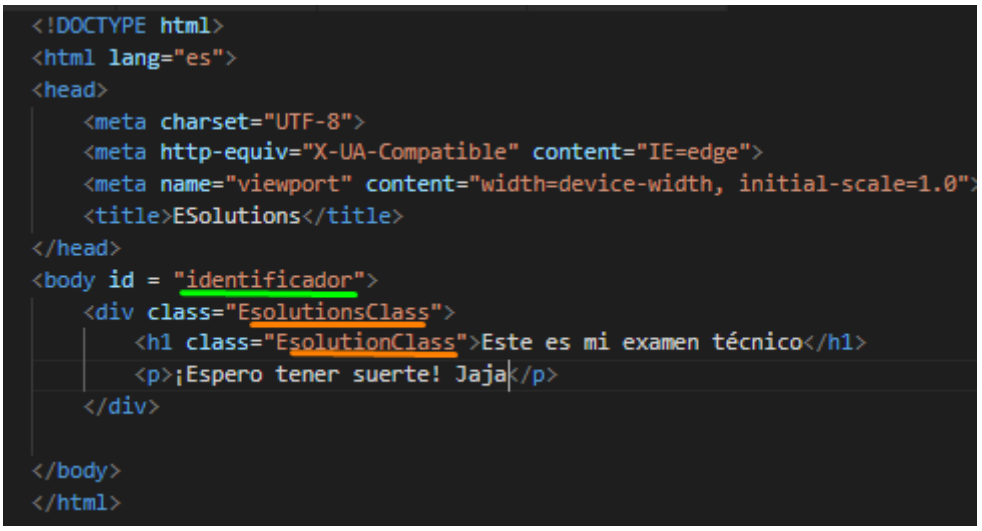
Tenemos varias diferencias entre las clases e identificadores, empezaría por la forma en la que se llaman a cada una.
A los ID (identificador, en este caso) se le llaman de las siguientes maneras (con un #):



Mientras que a las clases (EsolutionsClass) se los llaman con un '.'.

Por otro lado, un elemento HTML puede tener más de una clase, mientras que los identificadores son UNICOS. Es decir, no puede haber mas de un #identificador (en este caso) definido en todo el documento HTML.

Sería correcto utilizarlo así:



10. ¿Cuál es la diferencia entre las posiciones `relative`, `fixed`, `absolute` y `static` para un elemento dado?

Respuesta:

En el caso de static, esté tomará el valor que le corresponde y no hará caso a los valores de top, left, right, down.

En el caso de relative, podremos utilizar el valor que le corresponde por defecto (como static) pero podremos posicionarlo con los valores de top, left, right, down.

En el caso de absolute, también podremos posicionarlos con los valores de top, left, right, down, pero no comenzará en el valor que le corresponde por defecto, si no que tomara como referencia el elemento posicionado previamente o la ventana del navegador.

Y, por último, pero no menos importante, en el caso de fixed, siempre permanece en el

mismo lugar incluso si se scrollea la página. Utilizamos los valores de top, left, right, down para posicionar el elemento