

Actividad 2

A continuación se muestra el resultado de los ejercicios de la actividad 2.

Los cambios se subieron a mi fork en el siguiente link: https://github.com/engmmcasas/Javascript-Course/tree/main/activity/2_session

Para correrlo, basta con descargar el proyecto, acomodar los archivos en una carpeta individual, correr el live server, abrir consola de navegador (preferentemente chrome) y observar los resultados.

Respuesta Ejercicio 1.

La implementacion del ejercicio 1 se dividio en index.js dentro de la carpeta 2_session, y en una carpeta interior llamada ejercicio1 que alberga un archivo ejercicio1.js que tiene la funcion requerida de aceptar un objeto y regresar sus propiedades. Aqui abajo el codigo tanto de index.js y ejercicio1.js, y una captura de pantalla de la consola.

Codigo index.js:

```
// Ejercicio 1

import { ejercicio1 } from "../ejercicio1/ejercicio1.js";
console.log("#####");
console.log("#####");
console.log("Ejercicio 1");
const threePointsStrategy = {
  diasTrabajo: "fin de semana",
  horarioTrabajo: "12pm a 5pm",
  motivacion: "completar mi formacion como data scientist/machine learning engineer",
  softwareComplementarios: "django/flask en python y, herramientas big data en java/scala"
}

let ejercicio1_array = ejercicio1(threePointsStrategy);
console.log(ejercicio1_array);
```

Codigo ejercicio1.js

```
function ejercicio1(objt){

  let output_array = [];
  for (let property in objt){
    output_array.push(property);
  }
  return output_array;
}

export { ejercicio1 };
```

Respuesta en consola:

```
#####  
#####  
Ejercicio 1  
▶ (4) ['diasTrabajo', 'horarioTrabajo', 'motivacion', 'softwareComplementarios']
```

Respuesta Ejercicio 2.

Para el ejercicio dos se creo codigo en un .js aparte llamado ejercicio2.js y que esta dentro de una carpeta igualmente llamada ejercicio2. Este script se llamo dentro del index.html para que se desplegara dentro de la consola y exponga los ejemplos de cada valor de this. Aqui abajo respuestas en consola y resultado de los ejemplos, y captura de pantalla de codigo para ejemplos.

Respuestas en consola:

#####	ejercicio2.js:1
#####	ejercicio2.js:2
Ejercicio 2	ejercicio2.js:3
##### En modo global #####	ejercicio2.js:11
En el contexto global, this es igual a window: true	ejercicio2.js:15
##### En una funcion sin especificar modo #####	ejercicio2.js:17
En una funcion simple, this es igual a window: true	ejercicio2.js:23
##### En una funcion en modo estricto #####	ejercicio2.js:27
En una funcion en modo estricto, this es igual a sin especificar: true	ejercicio2.js:34
##### Metodo dentro de un objeto #####	ejercicio2.js:39
En un metodo dentro de un objeto, se refiere a la propiedad dentro del objeto, en esta funcion this es igual al valor dentro de la propiedad 'propiedadMuestra': true	ejercicio2.js:49
##### Metodo arrow dentro de un objeto #####	ejercicio2.js:57
En un metodo arrow dentro de un objeto, se refiere al contexto de la funcion, en esta funcion this es igual al elemento variable propiedadMuestra: true	ejercicio2.js:66
##### En un evento #####	ejercicio2.js:73
##### Usando call #####	ejercicio2.js:83
call puede modificar el this. En una funcion normal, this seria el contexto o window en este caso, pero aqui se paso el objeto.	ejercicio2.js:93
Se pudieron multiplicar los dos numeros dentro la funcion multiplicarNumeros, en este ejemplo se introdujo 4 a la funcion y el objeto contiene el numero 3, por lo que el resultado es 12: true	ejercicio2.js:94
##### Usando apply #####	ejercicio2.js:104
apply funciona igual que call pero ahora se pasan los datos en lista, igual el resultado es 12: true	ejercicio2.js:97
##### Usando bind #####	ejercicio2.js:113
Con bin, regresa una funcion donde se une otra funcion y un objeto, y la funcion puede hacer uso de los elemento dentro del objeto	ejercicio2.js:122

Captura pantalla codigo con ejemplos:

```
console.log("#####");
console.log("#####");
console.log("Ejercicio 2");

//Enumera los distintos valores que puede tener "this" y pon un ejemplo de cada uno.
//se utiliza en objetos (contextos globales y de funcion), dependiendo si se ejecuto en modo estricto o no.
//en el contexto global sera window

//en modo global es igual al window.
console.log("##### En modo global #####");

let isThisWindow = this === window;
let valor1 = `En el contexto global, this es igual a window: ${isThisWindow}`;
console.log(valor1);

console.log("##### En una funcion sin especificar modo #####");

//en una funcion sin especificar que modo es, es window.
function f1(window_input){
    let isThisWindow = this === window_input;
    let valor1 = `En una funcion simple, this es igual a window: ${isThisWindow}`;
    console.log(valor1);
}
f1(window);

console.log("##### En una funcion en modo estricto #####");

//en una funcion en modo estricto, es sin especificar
function f2(){
    'use strict';
    let isThisWindow = this === undefined;
    let valor1 = `En una funcion en modo estricto, this es igual a sin especificar: ${isThisWindow}`;
    console.log(valor1);
}
f2();
```

```
console.log("##### Metodo arrow dentro de un objeto #####");

var propiedadMuestra = "Propiedad en contexto";

let pruebaObjeto2 = {
    propiedadMuestra: "Propiedad muestra",

    metodoMuestra: () => {
        let valor = `En un metodo arrow dentro de un objeto, se refiere al contexto de la funcion, en esta funcion this es igual al elemento variable propiedadMuestra: ${this.propiedadMuestra}`;
        console.log(valor);
    }
};

pruebaObjeto2.metodoMuestra();

// en un evento, se refiere al lemento que recibe el evento.
console.log("##### En un evento #####");

function actionButtonReceiver(element){
    console.log("En un evento, se refiere al lemento que recibe el evento");
    console.log(element);
}
```

```
// call puede modificar el this. En una funcion normal, this seria el contexto o window en este caso, pero aqui se paso el objeto
// y se pudieron multiplicar los dos numeros dentro la funcion multiplicarNumeros.
console.log("##### Usando call #####");

let objeto = {
  numero: 3
};

let multiplicarNumeros = function(otroNumero, type_input){
  let resultado = this.numero * otroNumero;
  let comparativa = resultado === 12;
  if (type_input === "call"){
    console.log("call puede modificar el this. En una funcion normal, this seria el contexto o window en este caso, pero aqui se paso el objeto.");
    console.log("Se pudieron multiplicar los dos numeros dentro la funcion multiplicarNumeros, en este ejemplo se introdujo 4 a la funcion y el objeto contiene el numero 3, por lo que el resultado es 12: " + resultado);
  }
  else{
    console.log('apply funciona igual que call pero ahora se pasan los datos en lista, igual el resultado es 12: ${comparativa}');
  }
}

multiplicarNumeros.call(objeto, 4, "call");

console.log("##### Usando apply #####");

// apply funciona igual que call pero ahora se pasan los datos como arreglo

let numeros = [4, "Apply"]
multiplicarNumeros.apply(objeto, numeros);
```

```
// Para finalizar, Bind
console.log("##### Usando bind #####");

let objetoToBind = {
  texto1: "Con bin, regresa una funcion donde se une otra funcion y un objeto",
  texto2: " ", y la funcion puede hacer uso de los elemento dentro del objeto"
};

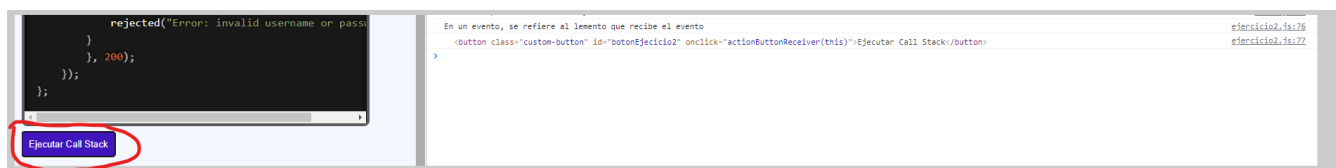
let functionToBind = function(){
  console.log(`${this.texto1}${this.texto2}`);
}

let funcionRegresada = functionToBind.bind(objetoToBind);
funcionRegresada();
```

Para evento:

Se creo un boton dentro de index.html, hasta el final que imprime en consola el objeto regresado. Aqui abajo la captura.

En rojo, el boton nuevo creado y a la derecha el nombre de este boton impreso en consola.



Respuesta Ejercicio 3.

Igualmente, se hizo una carpeta llamada ejercicio3 que tiene dentro un archivo llamado ejercicio3.js que realiza lo pedido para este ejercicio.

Aqui captura de pantalla de la consola demostrando lo que se solicito y captura de pantalla de codigo.

Respuesta consola:

#####	index.js:26
#####	index.js:27
Ejercicio 3	index.js:28
String a invertir: ThreePoints me ayuda a mi formacion profesional	index.js:30
lanoiseforp noicamrof im a aduya em stnioPeerhT	index.js:32

Captura pantalla codigo index.js:

```
// Ejercicio 3

import {ejercicio3} from "../ejercicio3/ejercicio3.js";

console.log("#####");
console.log("#####");
console.log("Ejercicio 3");
let string_input = "ThreePoints me ayuda a mi formacion profesional";
console.log(`String a invertir: ${string_input}`);
let string_reversed = ejercicio3(string_input);
console.log(string_reversed);
```

funcion en ejercicio3.js:

```
1 let ejercicio3 = (input_string) => {return input_string.split("").reverse().join("")}
2
3 export { ejercicio3 };
```

Respuesta Ejercicio 4.

La clase pedida se hizo dentro un archivo ejercicio4.js dentro de la carpeta ejercicio4 y fue importada dentro de index.js para poder usar dentro de este script. Solo se tiene captura de pantalla dentro de esta seccion pero el resultado es usado en ejercicio 5.

Captura pantalla codigo en ejercicio4.js:

```
class ejercicio4_Login {  
  
  constructor(username, password){  
  
    this.username = username;  
    this.password = password;  
  
  }  
  
  login(){  
  
    if (this.username === "admin" && this.password === "passwd"){  
      alert("User logged in");  
    }  
    else{  
      alert("user or password incorrect");  
    }  
  
  }  
  
}  
  
export {ejercicio4_Login};
```

Codigo importado en index.js:

```
// Ejercicio 4  
  
import { ejercicio4_Login } from "../ejercicio4/ejercicio4.js";
```

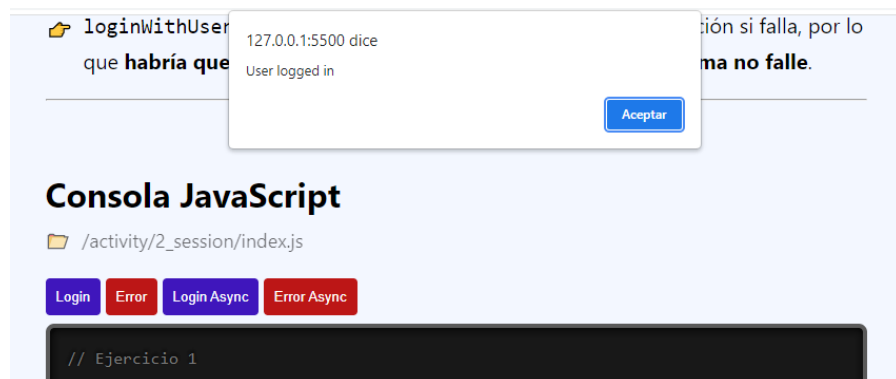
Respuesta Ejercicio 5.

En este ejercicio se hizo uso del codigo hecho en ejercicio 4 por el script ejercicio4.js y el codigo implementado en index.js donde se traen los elementos botones de index.html para posteriormente crearles funciones eventos en caso de click. El resultado es el esperado: para el boton Login sale una alerta comentando que el "log in" fue hecho correctamente, mientras que en error muestra una alerta diciendo que el usuario o password son incorrectos.

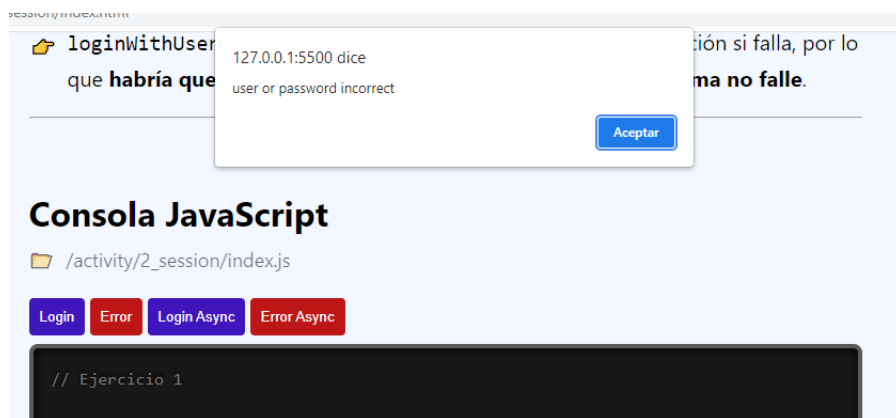
Aqui la respuesta despues de presionar los dos botones y abajo el codigo para este ejercicio.

Captura pantalla para dos botones:

Login:



Error:



Codigo index.js:

```
// Ejercicio 5

const button1 = document.getElementById("loginSuccess");
const button2 = document.getElementById("loginFailure");

button1.addEventListener("click", event => {
  let login = new ejercicio4_Login("admin", "passwd");
  login.login();
});

button2.addEventListener("click", event => {
  let loginWrong = new ejercicio4_Login("Manuel", "el_mero_password_123");
  loginWrong.login();
});
```

Respuestas Ejercicio 6.

Para el siguiente ejercicio se aprovecho el código ya implementado en ejercicio 5 dentro de index.js para traerse los elementos botones, se agrego la palabra reservada “async” dentro del callback de metodo “addEventListener” para que pueda aceptar una funcion asincrona y tomando el código ejemplo del manual del curso, se agrego las líneas de código que corren la función asincrona, esperan por la respuesta y en caso de error lo imprimen, esto dentro de un try-catch.

Aqui abajo captura de pantalla del código y respuesta en consola.

Captura pantalla código.

```
57
58 // Ejercicio 6
59
60 const button3 = document.getElementById("loginSuccessAsync");
61 const button4 = document.getElementById("loginFailureAsync");
62
63
64 let loginWitUsername = (username, password) => {
65     return new Promise(function (resolve, rejected) {
66         setTimeout(() => {
67             if (username === "admin" && password === "passwd") {
68                 resolve("User logged in");
69             } else {
70                 rejected("Error: invalid username or password");
71             }
72         }, 2000);
73     });
74 };
75
76
77 button3.addEventListener("click", async () => {
78
79     try{
80         let response = await loginWitUsername("admin", "passwd");
81         console.log(response);
82     }
83
84     catch (err){
85         console.log(err);
86     }
87 });
88
89 button4.addEventListener("click", async () => {
90
91     try{
92         let response = await loginWitUsername("admin2", "passwd2");
93         console.log(response);
94     }
95
96     catch (err){
97         console.log(err);
98     }
99 });
```


Captura pantalla respuesta en consola.

[Unreadable message: an object is returned]		index.js:81
User logged in		index.js:97
Error: invalid username or password		index.js:97
User logged in		index.js:81
Error: invalid username or password		index.js:97
User logged in		index.js:81
Error: invalid username or password		index.js:97
User logged in		index.js:81
Error: invalid username or password		index.js:97
>		