

## EJERCICIOS DE ASINCRONISMO

Dada la siguiente función:

```
function exampleAsync(valor) {  
  return new Promise((resolve) => {  
    setTimeout(() => {resolve(valor + 1)}, 1000)  
  });  
}
```

Resolver la operación de manera que llegue a “10” el valor retornado, tanto con async/await como con promesas.

Dado las siguientes variables e invocación:

```
const piValidValue = 3.14  
const piWrongValue = 3.10  
  
operation(piValidValue)  
.then((res) => console.log("Válido! ", res))  
.catch((err) => console.log(err))  
  
operation(piWrongValue)  
.then((res) => console.log(res))  
.catch((err) => console.log("No válido! ", err))
```

Crear la función “operation”, que retorne una promesa con un caso de éxito y otro caso de error según corresponda.

Dado el siguiente bloque de código:

```
const data =  
await fetch("https://restcountries.com/v3.1/name/argentina")  
  
async function countryInformation(data) {  
  const myCountry = data.json()  
  console.log("Datos: ", myCountry)  
}  
  
countryInformation(data)
```

Qué modificación debo hacer en el código para que no retorne un objeto “Promise { <pending> }”?