

Otro concepto nuclear que está en el "core", en el núcleo, en el corazón de Javascript y hay que entenderlo para trabajar bien con Javascript, es la programación asíncrona. La posibilidad de trabajar con una secuencia de código que no se detiene a esperar un resultado, así trabaja Javascript.

Otros lenguajes, por ejemplo, pueden decir "voy a buscar datos a un sitio" o, mejor dicho, el programa podría decir la instrucción "ve a buscar los datos a un sitio", a una base de datos, eso no es instantáneo como leer de una, de un "array", porque esa información viene de fuera, viene de un servidor. Ustedes dirán "bueno, pero es muy rápido", sí, probablemente sea muy rápido tal vez, a veces, va a durar menos de un segundo, es imperceptible, pero no es instantáneo.

Hay lenguajes de programación que cuando decimos "ve a buscar información a un sitio", se quedan esperando la respuesta, pero Javascript no se queda esperando la respuesta, Javascript nunca nos espera, por lo tanto, si nosotros decimos "ve a buscar la respuesta" y muestra la respuesta, como no se va a quedar esperando, la respuesta que nos muestra será incorrecta, porque en realidad, en el mundo Javascript todavía no los tienen, los está buscando, pero el código se sigue ejecutando y ustedes dirán "bueno, así gana, fin del juego, no sirve". No, obviamente que esto no es así. Hay maneras de decirle a Javascript "ve a buscar los datos y muéstame dos cuando los tengas" y luego, si el programa sigue avanzando con otras cosas, con otras cuestiones, básicamente está explicado, tal vez, de una manera muy primaria, pero es nuestra primera aproximación, así que permítame la simplificación. De esto se trata de la programación asíncrona.

Para manejar el código asíncrono, Javascript utiliza "promesas". Supongan que tengo una función que dice "voy a buscar los datos" y luego muestra datos, bueno, Javascript lo ejecuta y esa función no devuelve los datos, devuelve una "promesa" que dice "te daré los datos o le mostraré un error, pero tranquilo, porque cuando esto está resuelto yo le voy a avisar" y efectivamente, así es como funciona. ¿Cuerda entenderlo un poco?, sí. Si yo fuera un novato y estuviera escuchando esta explicación, necesitaría ejemplos. Aquí tienen un ejemplo, creo que bien detallado.

Vamos a entender el concepto de "promesa", para luego ir a buscar información como haríamos en la vida real, con una API que se llama "fetch".

Para primero, para que entiendan cómo funciona "fetch" hagamos nuestro propio "fetch", vamos a construir uno para que ustedes entiendan cómo funcionan las "promesas", cómo trata Javascript el código asíncrono.

Seguimos haciendo aquí nuestro JSON con empleados, simulando que éste es una base de datos o una API externa que nos entrega esta información. Sería ideal si pudiéramos construir una función "get data", está, por ejemplo, que recibe un recurso y luego, simplemente, bueno, lo convierte a formato Javascript válido, para finalmente retornarlo. Podríamos hacer aquí, por ejemplo, "const data, igual, JSON.parse de recurso", está ahí por parámetros, ¿verdad? Es la manera en la que le diremos a esta función "escuchame, ve a buscar la información a tal sitio, ¿bien?" y luego de hacer esto podría devolver, mostrar esa información" Muy bien.

Y ahora podríamos llamar a la función "get data" e indicarle, permítame jugar juntos, vamos a suponer que JSON, como hacemos hasta ahora es, bueno, pues, la dirección externa