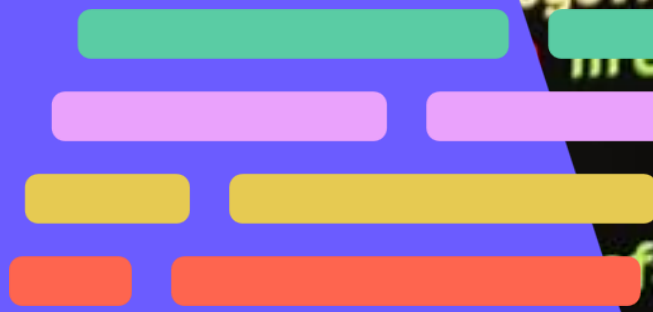
Somos un ecosistema de desarrolladores de software

Java Script POO (Document Object Model)



SYNCHRONUS CODE

</Riwi>

- ❑ Most code is synchronous;
- ❑ Synchronous code is executed line by line;
- ❑ Each line of code waits for previous line to finish;
- ❑Long-running operations block code execution.

SYNCHRONUS

BLOCKING

</Riwi>

```
const p = document.querySelector('.p');
p.textContent = 'My name is
Johny¡';alert('Text set¡ ');
p.style.color = 'red';
```

SYNCHRONUS CODE

</Riwi>

ASYNCHRONOUS

CALLBACK WILL RUN AFTER TIMER

```
</Riwi>

const p = document.querySelector('.p');
setTimeout(function(){
p.textContent = 'My name is Johny¡ ';
}, 5000);
p.style.color = 'red';
```

- ❑ Asynchronous code is executed after a task that runs in the“background” finishes;
- ❑ Asynchronous code is non-blocking;
- ❑ Execution doesn’t wait for an asynchronous task to finish its work;
- ❑ Callback functions alone do NOT make code asynchronous!

EXECUTED AFTER ALL OTHER CODE

```
</Riwi>

[1, 2, 3].map(v⇒ v*2);
```

callback does NOT automatically make code asynchronous

SYNCHRONUS CODE

</Riwi>

Asynchronous

```
</Riwi>

const img = document.querySelector('.dog');
img.src = 'dog.jpg';
img.addEventListener('load', function(){
    img.classList.add('fadeIn');
});
p.style.width = '300px'
```

CALLBACK WILL RUN AFTER IMAGE LOADS

addEventListener does NOT automatically make code asynchronous¡

❑ Example: Asynchronous image loading with event and callback Image loading Asynchronous
❑ Other examples: Geolocation API or AJAX calls

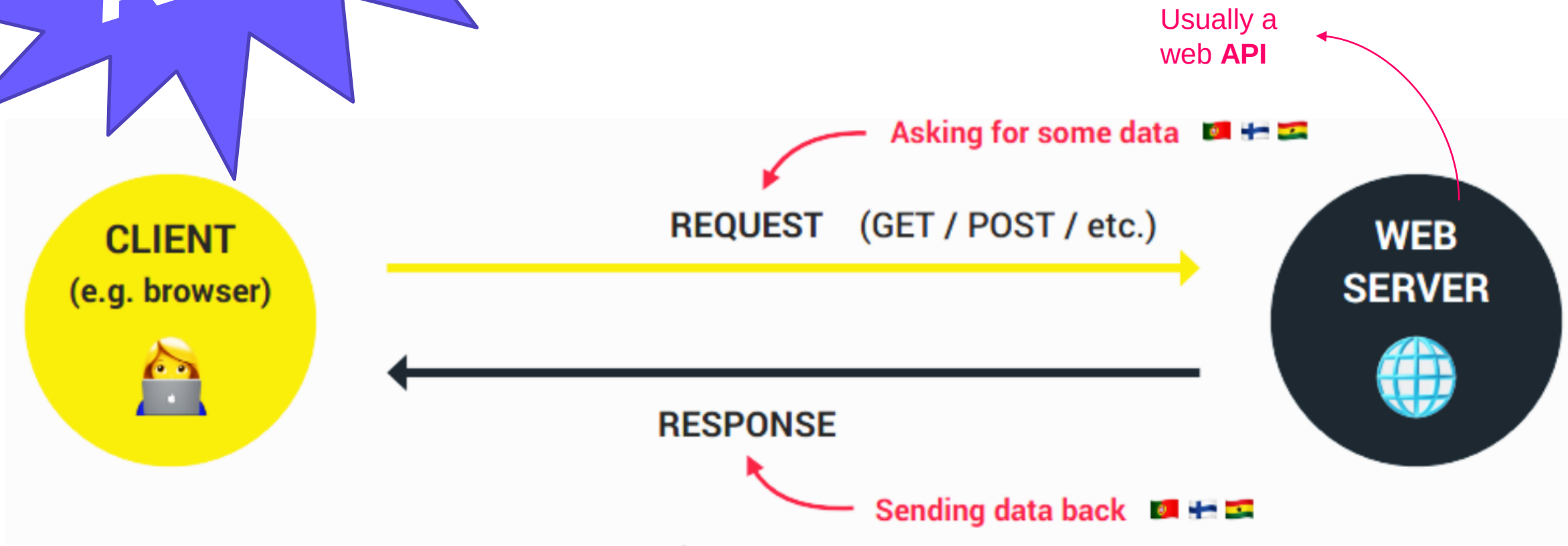
➢ **Asynchronous code is executed after a task that runs in the “background” finishes;**
➢ **Asynchronous code is non-blocking;**
➢ **Execution doesn’t wait for an asynchronous task to finish its work;**
➢ **Callback functions alone do NOT make code asynchronous!**

SYNCHRONUS CODE

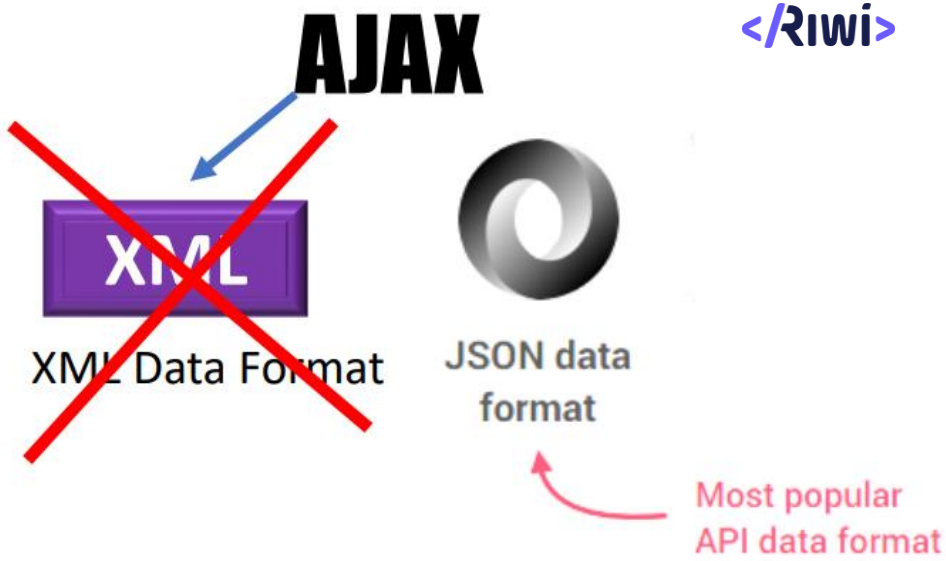
</Riwi>

AJAX

Asynchronous JavaScript And XML:
Allows us to communicate with remote web servers in an asynchronous way.
With AJAX calls, we can request data from web servers dynamically



What is a an API?



Application Programming Interface: Piece of software that can be used by another piece of software, in order to allow applications to talk to each other;

✓ There are be many types of APIs in web development:

✓ "Online" API: Application running on a server, that receives requests for data, and sends data back as response;

✓ We can build our own web APIs (requires back-end development, e.g. with node.js) or use 3rd-party APIs.

There is an API for everything

- Weather data
- Data about countries
- Flights data
- Currency conversion data
- APIs for sending email or SMS
- Google Maps
- Millions of possibilities..

PROMISE

❖ Promise: An object that is used as a placeholder for the future result of an asynchronous operation.

❖ Promise: A container for an asynchronously delivered value.

❖ Promise: A container for a future value.

❖ We no longer need to rely on events and callbacks passed into asynchronous functions to handle asynchronous results;

❖ Instead of nesting callbacks, we can chain promises for a sequence of asynchronous operations: escaping callback hell

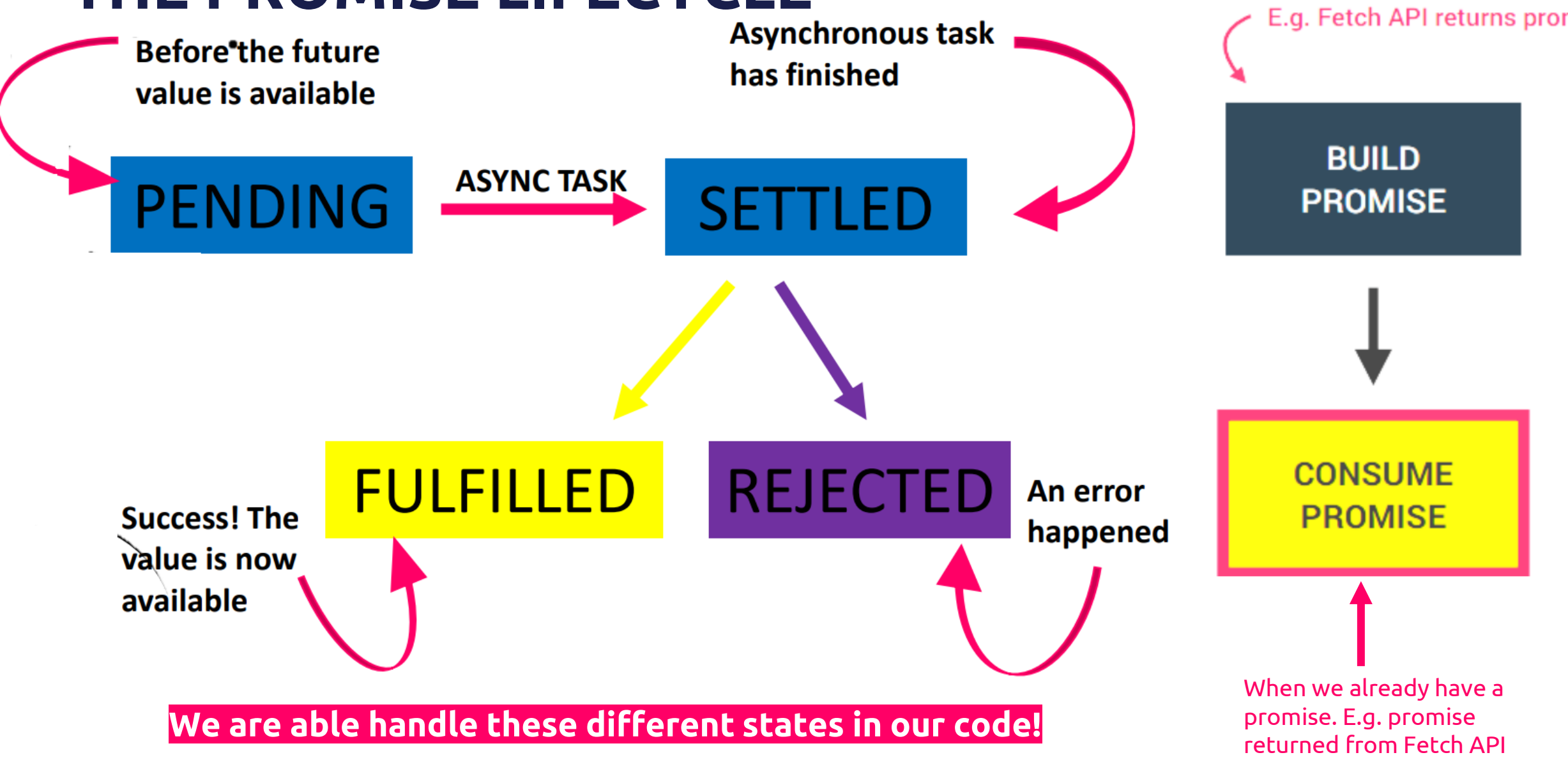
Promise that I will receive money if I guess correct outcome.

❑ I buy lottery ticket (promise) right now

❑ Lottery draw happens asynchronously

❑ If correct outcome, I receive money, because it was promised

THE PROMISE LIFECYCLE



We are able handle these different states in our code!

</Be a coder>