JAVASCRIPT

OBJECT ORIENTED PROGRAMMING (OOP) WITH JAVASCRIPT

WHAT IS OBJECT-ORIENTED PROGRAMMING?

Object-oriented programming (OOP) is a programming paradigm based on the concept of objects

We use objects to model (describe) real-world or abstract features

Objects may contain data (properties) and code (methods). By using objects, we pack data and the corresponding behavior into one block

In OOP, objects are self-contained pieces/blocks of code

Objects are building blocks of applications, and interact with one another

Interactions happen through a public interface (API): methods that the code outside of the object can access and use to communicate with the object

OOP was developed with the goal of organizing code, to make it more flexible and easier to maintain

THE 4 FUNDAMENTAL OOP PRINCIPLES

ABSTRACTION

Ignoring or hiding details that don't matter, allowing us to get an overview perspective of the thing we're implementing, instead of messing with details that don't really matter to our implementation

ENCAPSULATION

Keeping properties and methods private inside the class, so they are not accessible from outside the class. Some methods can be exposed as a public interface (API)

INHERITANCE

Making all properties and methods of a certain class available to a child class, forming a hierarchical relationship between classes. This allows us to reuse common logic and to model real-world relationships

POLYMORPHISM

A child class can overwrite a method it inherited from a parent class

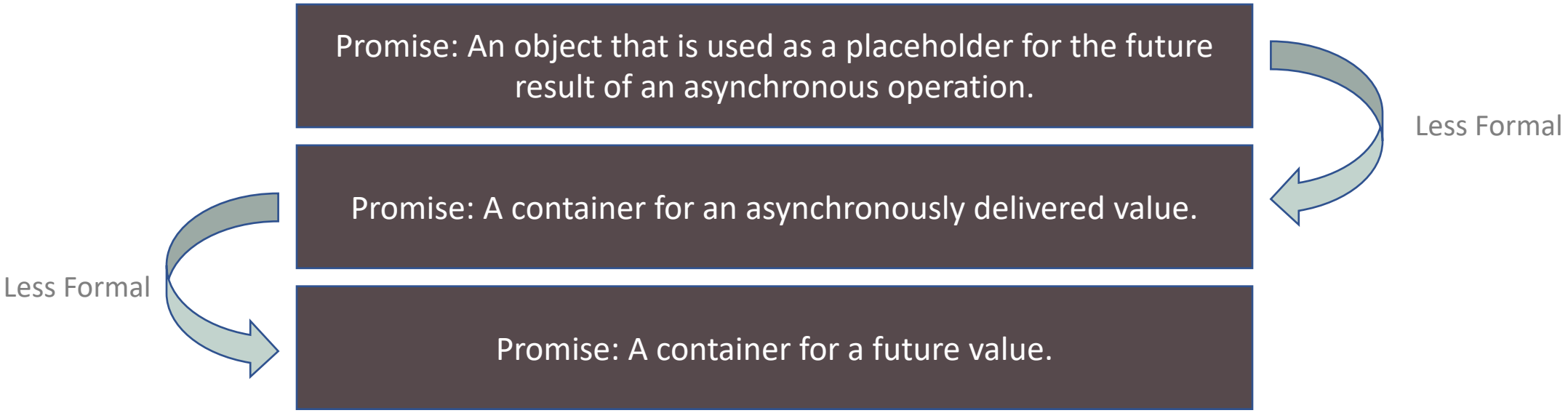
ES6 & MORE – NEW FEATURES

ARROW FUNCTION	DESTRUCTURING	REST / SPREAD
TEMPLATE LITERALS	BLOCK SCOPING	MAP/SET
CLASSES	DEFAULT PARAMETERS	PROMISE

ASYNCHRONOUS JAVASCRIPT

PROMISES, ASYNC/ AWAIT AND MORE

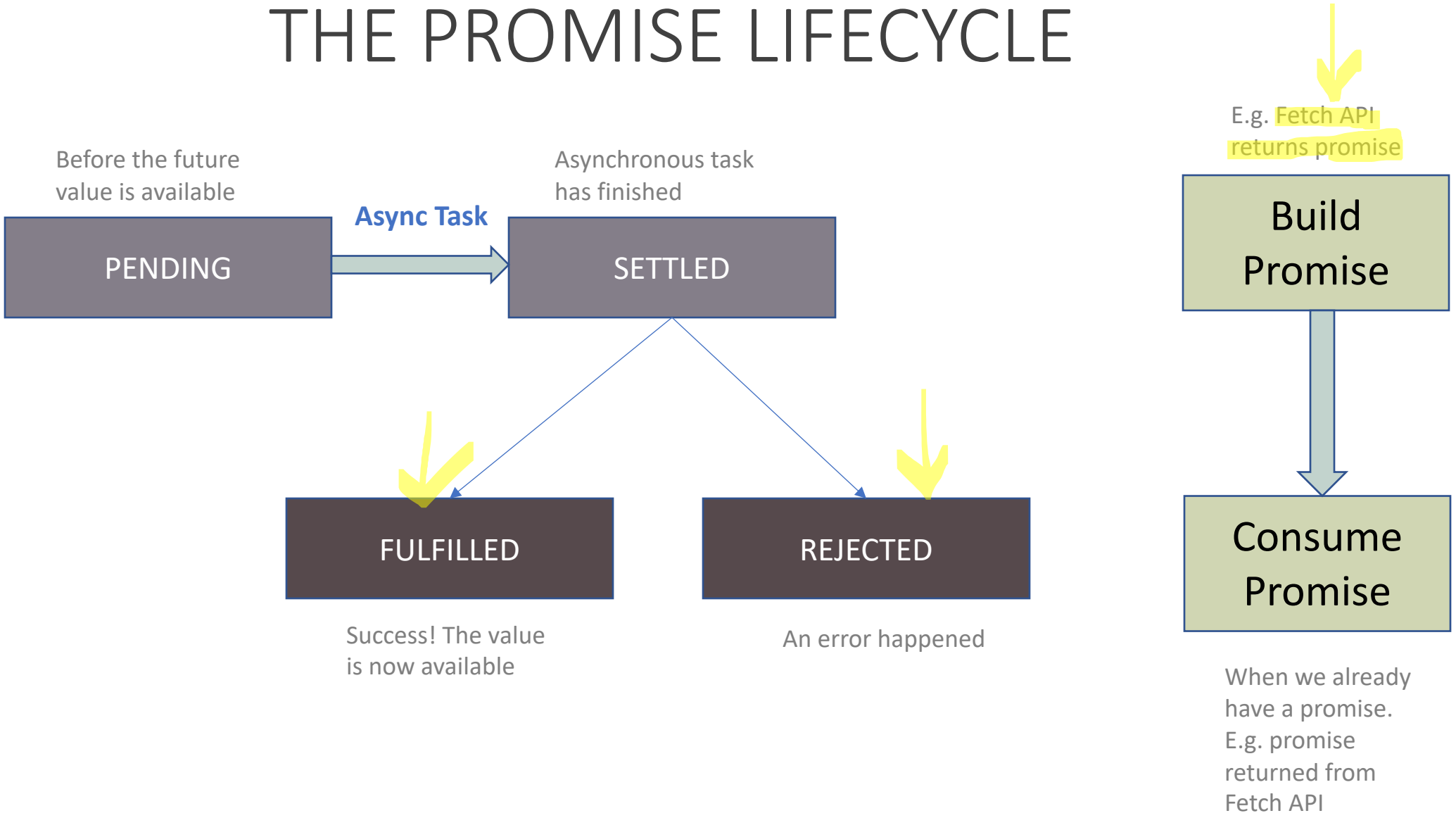
WHAT ARE PROMISES?



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

THE PROMISE LIFECYCLE



MODERN JAVASCRIPT DEVELOPMENT

WRITING CLEAN AND MODERN JAVASCRIPT

REVIEW: MODERN AND CLEAN CODE

READABLE CODE

- Write code so that others can understand it
- Write code so that you can understand it in 1 year
- Avoid too “clever” and overcomplicated solutions
- Use descriptive variable names: what they contain
- Use descriptive function names: what they do

FUNCTIONS

- Generally, functions should do only one thing
- Don’t use more than 3 function parameters
- Use default parameters whenever possible
- Generally, return same data type as received
- Use arrow functions when they make code more readable

REVIEW: MODERN AND CLEAN CODE

GENERAL

- Use DRY principle (refactor your code)
- Don’t pollute global namespace, encapsulate instead
- Don’t use var
- Use strong type checks (=== and !==)

OOP

- Use ES6 classes
- Encapsulate data and don’t mutate it from outside the class
- Implement method chaining
- Do not use arrow functions as methods (in regular objects)

REVIEW: MODERN AND CLEAN CODE

AVOID NESTED CODE

- Use early return (guard clauses)
- Use ternary (conditional) or logical operators instead of if
- Use multiple if instead of if/else-if
- Avoid for loops, use array methods instead
- Avoid callback-based asynchronous APIs

ASYNCHRONOUS CODE

- Consume promises with async/await for best readability
- Whenever possible, run promises in parallel (Promise.all)
- Handle errors and promise rejections

REFERENCES

READING MATERIAL

- https://javascript.info/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript

VIDEO LINKS

- https://www.youtube.com/watch?v=NCwa_xi0Uuc
- https://www.youtube.com/watch?v=nZ1DMMsyVyI