

DWA_04.3 Knowledge Check_DWA4

1. Select three rules from the Airbnb Style Guide that you find **useful** and explain why.

Destructuring - It saves you from creating temporary references for those properties, and from repetitive access of the object. Repeating object access creates more repetitive code, requires more reading, and creates more opportunities for mistakes. Destructuring objects also provides a single site of definition of the object structure that is used in the block, rather than requiring reading the entire block to determine what is used.

Classes & Constructors - Classes and constructors are important concepts in object-oriented programming (OOP), and they provide several benefits that contribute to the structure, organization, and reusability of code. It provides a way to structure code, promote code reuse, enable inheritance, and improve overall code organization. They are essential tools for creating maintainable, modular, and scalable software.

Variables - Is used to store and manage data in a program. They allow programmers to assign names to memory locations, making it easier to reference and manipulate values. This facilitates the storage of information like numbers, strings, objects, and more. They contribute to code readability, reusability, and the dynamic nature of programs, making them a foundational concept in software development.

2. Select three rules from the Airbnb Style Guide that you find **confusing** and explain why.

Destructuring - It allows you to extract values from arrays or objects and assign them to variables in a concise and readable way. While it is a powerful feature, it can be perceived as confusing for several reasons, especially for beginners or those who are not familiar with its syntax and use cases. To overcome the confusion associated with destructuring, developers can benefit from studying examples, working on hands-on exercises, and gradually incorporating destructuring into their coding practice.

Arrow Functions - While it provides concise syntax and powerful features, developers need to be aware of their specific behaviors, especially regarding this binding, lack of the arguments object, and the nuances of implicit returns. Familiarity with these characteristics helps in avoiding potential pitfalls and using arrow functions effectively.

Iterators and generators - It introduces powerful features to JavaScript, but their syntax, concepts, and use cases may initially be confusing for developers who are not yet familiar with them. It's recommended to practice and experiment with iterators and generators to gain a better understanding of their capabilities and use cases.
