1. What is scope? Your explanation should include the idea of global vs. local scope.

Scope is the range of influence of a given variable. Variables that are declared within a function only have jurisdiction within that function, and are considered to have "local scope". When a variable is declared outside of the function *or* inside the function with no keyword (let/const), then this is considered to be on the global scope and is accessible from anywhere in the code -- potentially even other files. Declarations made with the var keyword also result in global scope.

1. Why are global variables avoided?

The document-wide scope of global variables makes them unpredictable and difficult to control.

1. Explain JavaScript’s strict mode

Javascript’s strict mode prohibits the user from declaring variables without a keyword. If a keyword is skipped or forgotten, it will trigger an error which is an easy way to catch this problem before it’s created so global variables are easily avoided.

1. What are side effects, and what is a pure function?

When side effects occur, a function travels outside its own scope and alters something higher in the scope chain. When this happens, the code can be altered and other functions or attributes may misbehave. A pure function, on the other hand, is a function that is determinate and has zero side effects. It is predictable - it will return the same vale each time it’s given the same inputs.