Advanced JavaScript Coding Examples

What is the difference between var, let, and const in JavaScript?

```
// Example
var a = 1;
let b = 2;
const c = 3;
```

Explanation:

var is function-scoped and can be redeclared within the same function or globally.

let and const are block-scoped and cannot be redeclared within the same block, but let can be reassigned while const is read-only.

What is a closure in JavaScript?

```
// Example
function outerFunction() {
  let a = 1;
  function innerFunction() {
    console.log(a);
  }
  return innerFunction;
}
```

```
const func = outerFunction();
func(); // 1
```

Explanation:

A closure is a combination of a function and the lexical environment within which that function was declared. In this example, innerFunction has access to a even though it's not within its own scope because it was declared within outerFunction and is able to "remember" the value of a through the closure.

What is the difference between synchronous and asynchronous JavaScript?

Explanation:

Synchronous JavaScript executes code in a blocking manner, meaning that each line of code must be executed before moving on to the next line. Asynchronous JavaScript executes code in a non-blocking manner, meaning that multiple lines of code can be executed at the same time. Asynchronous code uses callbacks, promises, or async/await to manage the execution flow.

What is a higher-order function in JavaScript?

```
// Example
function doSomething(func) {
```

```
  func();
}

function hello() {
  console.log('Hello');
}

doSomething(hello); // Hello
```

Explanation:
A higher-order function is a function that takes a function as an argument or returns a function as its result. In this example, doSomething is a higher-order function because it takes the hello function as an argument.

What is event bubbling and how can it be prevented?

Explanation:
Event bubbling is a phenomenon where an event that occurs on an element in the DOM tree will also trigger the same event on all of its parent elements. To prevent event bubbling, you can use the stopPropagation() method within the event listener on the child element.

What is the difference between == and === in JavaScript?

Explanation:
== is a loose equality comparison that converts the operands to a common type before comparing them, whereas === is a strict equality comparison that compares the operands without converting their types.

What is a promise in JavaScript and how does it work?

```
// Example
const promise = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve('Hello');
  }, 1000);
});

promise.then((result) => {
  console.log(result); // Hello
});
```

Explanation:

A promise is a JavaScript object that represents the eventual completion or failure of an asynchronous operation and its resulting value. The Promise constructor takes a function with two parameters, resolve and reject, which are functions used to fulfill the promise or reject it with an error. The then() method is used to handle the fulfilled promise and returns a new promise that can be further chained with more then() or catch() methods.