(Notes) Call, Bind, and Apply

Question 1: What is Call?
Explanation:
The `call` method in JavaScript is used to invoke a function with a specified `this` context and arguments individually. It accepts the context object as the first argument followed by individual arguments.

Question 2: What is Apply?
Explanation:
The `apply` method in JavaScript is similar to `call` but accepts arguments as an array. It is used to invoke a function with a specified `this` context and an array of arguments.

Question 3: What is Bind?
Explanation:
The `bind` method in JavaScript is used to create a new function with a specified `this` context. It doesn't immediately execute the function but returns a new function that can be invoked later.

Question 4: Output
Explanation:
The first `console.log` outputs **"Piyush is 24"** because `call` immediately invokes the function with the specified context and arguments. The second `console.log` outputs the bound function because `bind` creates a new function but doesn't execute it.

Question 5: Call with function inside object
Explanation:
The `call` method is used to invoke the `getAge` function of the `person` object with a different context (`person2`). It returns the `age` property of `person2`, which is **24**.

Question 6: Output
Explanation:
The first `console.log` outputs **"🥑"** because `data.getStatus()` returns the `status` property of the `data` object. The second `console.log` also outputs **"🥑"** because

`data.getStatus.call(this)` uses `call` to explicitly set the context to the global object, accessing the same `status` property.

Question 7: Call printAnimals such that it prints all animals in object

Explanation:

The `printAnimals` function is invoked using `call` within a loop to print each animal's species and name from the `animals` array with the correct index.

Question 8: apply to append an array to another

Explanation:

The `apply` method is used to push all elements of the `elements` array into the `array` using the spread operator (`...elements`).

Question 9: Using apply to enhance built-in functions

Explanation:

The `apply` method is used to find the maximum number in the `numbers` array by passing the array as arguments to `Math.max`.

Question 10: How will you Create a bound function

Explanation:

A bound function is created using the `bind` method. In this case, `f` is bound to `null`, and the resulting function is assigned to `user.g`.

Question 11: Bind Chaining?

Explanation:

Bind chaining involves calling `bind` multiple times on a function. However, only the first `bind` call is effective in setting the `this` context. Subsequent `bind` calls have no effect.

Question 12: Fix the code

Explanation:

The `checkPassword` function prompts for a password and calls either `loginSuccessful` or

`loginFailed` based on the input. `bind` is used to ensure that `this` inside `loginSuccessful` and `loginFailed` refers to the `user` object.

Question 13: Partial application for login

Explanation:

The `checkPassword` function is modified to accept two callbacks, `ok` and `fail`. The `login` method in the `user` object is then passed as arguments to `askPassword`, but the specific partial application is not provided.

Question 14: Explicit Binding with Arrow Function

Explanation:

Arrow functions do not have their own `this` context and instead inherit it from the surrounding lexical scope. Therefore, using `call` or `bind` with arrow functions doesn't change the context.

Question 15: Call Method Polyfill

Explanation:

A polyfill for the `call` method is provided, allowing a function (`purchaseCar` in this case) to be invoked with a specific context (`car3`) and arguments (`'₹'` and `'60,00,000'`).

Question 16: Apply Method Polyfill

Explanation:

A polyfill for the `apply` method is provided, allowing a function (`purchaseCar`) to be invoked with a specific context (`car2`) and an array of arguments (`['₹', '50,00,000']`).

Question 17: Bind Method Polyfill

Explanation:

A polyfill for the `bind` method is provided, allowing a function (`purchaseCar`) to be bound to a specific context (`car1`) and arguments (`'₹'` and `'1,00,00,000'`). The bound function is then invoked to make a purchase.
