¡nter*f*eud!

the game that gets YOU hired

funktional

1

What is scope?

Scope is the current context of your code, in other words, the space of variables that your code currently has access to. Global scope is the highest level of scope. All local scopes are created through function declarations.

2

What is a **closure?**

A closure is a function in conjunction with the variables in the scope where that function was created.

3

What is partial function application?

Take a function that accepts multiple arguments, bind values to some of them, and return a new function that only accepts the remaining arguments. When invoked, a partially applied function will invoke the original function and merge passed-in arguments.

What is **currying**?

Currying transforms a function of N arguments so that it can be called as a chain of N single-argument functions. A curried function is "primed" for partial application. Only once all N functions have been called will the original be invoked with all N arguments.

4

What does eval do?

eval takes a string and runs it as JavaScript. If the string represents an expression, it evaluates the expression. If the string represents one or more statements, it executes the statements.

5

What is **arguments** inside of a function's body?

The arguments object is an Array-like object corresponding to the arguments passed to a function at execution time. This object is automatically available inside any function.

6

What is **this** by default?

By default, the `this` keyword refers to the outermost global object—`window` in a browser context and `global` in node.

Under what circumstances is **this** not the global obj.?

1. When a function is called as a method on an object, `this` refers to that object.
2. The `new` operator will creates an object and binds it to `this` in the ensuing function invocation.
3. The context can be explicitly set by `call`, `apply`, or `bind`.

7

What is the difference between call and apply?

Both call and apply are methods that invoke a function with a given context and arguments. Apply is exactly like call except that call expects its arguments in the parameter list while apply expects its arguments as a single array.

8

What is the difference between call and bind?

As opposed to call, bind returns a partially applied copy of the function. This copy will be bound with the given context and any additional arguments provided.

9

What is the difference between a function declaration and a function expression?

A function declaration is a line that begins with the "function" keyword followed by the function name. A function expression can be anonymous or named and is used somewhere within a statement.

10

What is a getter?

A function bound to an object property that runs whenever the property is accessed. The return value of the function is the value supplied when accessing the property.

★ ★ ★

¡10!

How can you define a getter?

```javascript
var person = {
  name : 'Gabriel',
  get greeting () { return 'hi, I am ' + this.name; }
};
console.log(person.greeting); // hi, I am Gabriel
// on pre-existing object:
Object.defineProperty(obj, "initial", {
  get: function () { return this.name[0]; }
});
console.log(person.initial); // G
```

11

Describe hoisting.

Variable declarations are initialized at the top of a scope, but their assignment stays in place. On the other hand, function declarations (but not function expressions) are moved entire to the top. This means you can invoke a function "before" declaring it.

12

Define **IIFE.**

IIFE stands for Immediately Invoked Function Expression—a function that is created and then invoked with no intermediate code running.

¡12!

Valid IIFE: (function (){})()
Invalid IIFE: function (){}()
Why?

The function to be invoked must be wrapped in parentheses so the JS parser knows to treat the func. as an expression and NOT a declaration. You cannot invoke a function declaration.

13

What does this log?

```
function f () {
    this.name = 'me';
}
console.log(f.name);
```

f

How do you determine a function's arity in JavaScript?

You can access the arity of a function via its `length` property.

15

What are higher order functions?

Functions that operate on other functions, either by taking them as arguments or by returning them. For example, forEach, reduce, or map.

16

[].**sort()** can take a **comparator function**. What should the comparator return, and how will that val. affect element order?

A number. If comparator(a, b) returns...
<u>negative number</u>: "a" should come first
<u>positive number</u>: "b" should come first
<u>zero</u>: neither takes precedence

17

Name three things a spy typically tracks.

A spy will usually record how many times the function was called, what parameters were passed in, and the return values. There are other possible values spies can track, however!

18

What is a **stub**?

A stub is used in automated testing to simulate a function—it gives a "canned response". For example, it could replace a slow operation or force an edge case.

⭐ ⭐ ⭐

¡18!

What is a **mock**?

A mock is a stub with expectations built in. The expectations specify the calls it is expected to receive. A mock will cause a test spec to fail if these expectations are not met.

19

What is functional programming?

☆☆☆☆☆