15.3.4.3 Function.prototype.apply(thisArg, argArray)

When the `apply` method is called with two arguments `thisArg` and `argArray`, it performs a function call using the `[[Call]]` property of the object. If the object does not have a `[[Call]]` property, a runtime error is generated. If `argArray` is supplied but is not an array or an arguments object (see 10.1.8), a runtime error is generated. The called function is passed `ToObject(thisArg)` as the `this` value and the elements of `argArray` as the arguments.

When `argArray` is not supplied, no arguments are passed. When `thisArg` is not supplied, the called function is passed the global object as the `this` value.

When `argArray` is supplied, the function is passed the (ToUint32(`argArray`.length)) arguments `argArray[0]`, `argArray[1]`, ..., `argArray[ToUint32(`argArray`.length)-1].

15.3.4.4 Function.prototype.call(thisArg, arg1, arg2, ...)

When the `call` method is called with one or more arguments `thisArg` and (optionally) `arg1`, `arg2` etc, it performs a function call using the `[[Call]]` property of the object. If the object does not have a `[[Call]]` property, a runtime error is generated. The called function is passed `ToObject(thisArg)` as the `this` value and `arg1`, `arg2` etc. as the arguments.

When `thisArg` is not supplied, the called function is passed the global object as the `this` value.