interfeud!

the game that gets YOU hired

Prep material:
- http://www.joshwright.com/tips/13-javascript-tips-and-tricks
- http://developer.telerik.com/featured/seven-javascript-quirks-i-wish-id-known-about/
- http://en.wikipedia.org/wiki/Functional_programming#Concepts
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions
- http://martinfowler.com/articles/mocksArentStubs.html#TheDifferenceBetweenMocksAndStubs

1

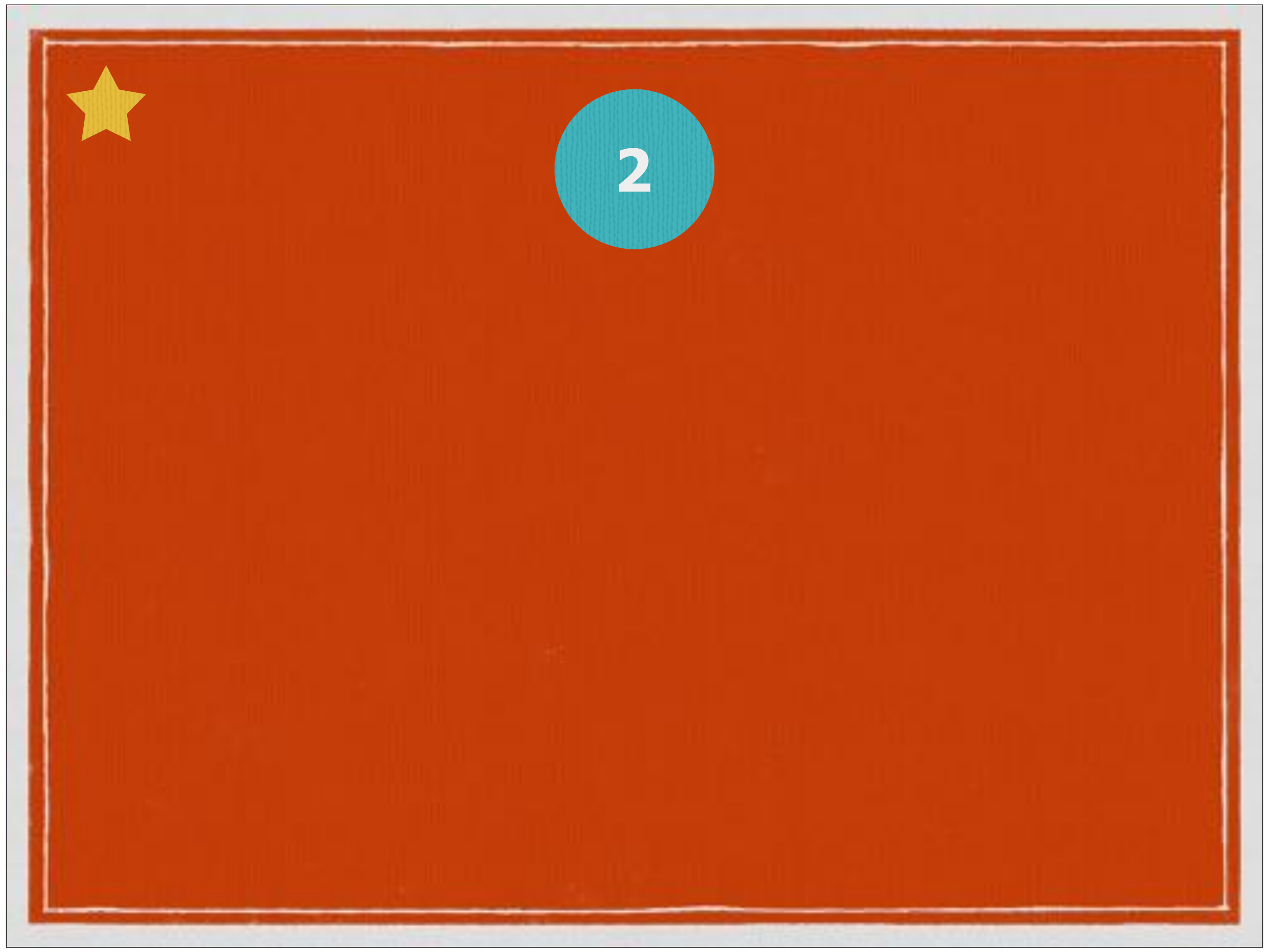
1

What is scope?

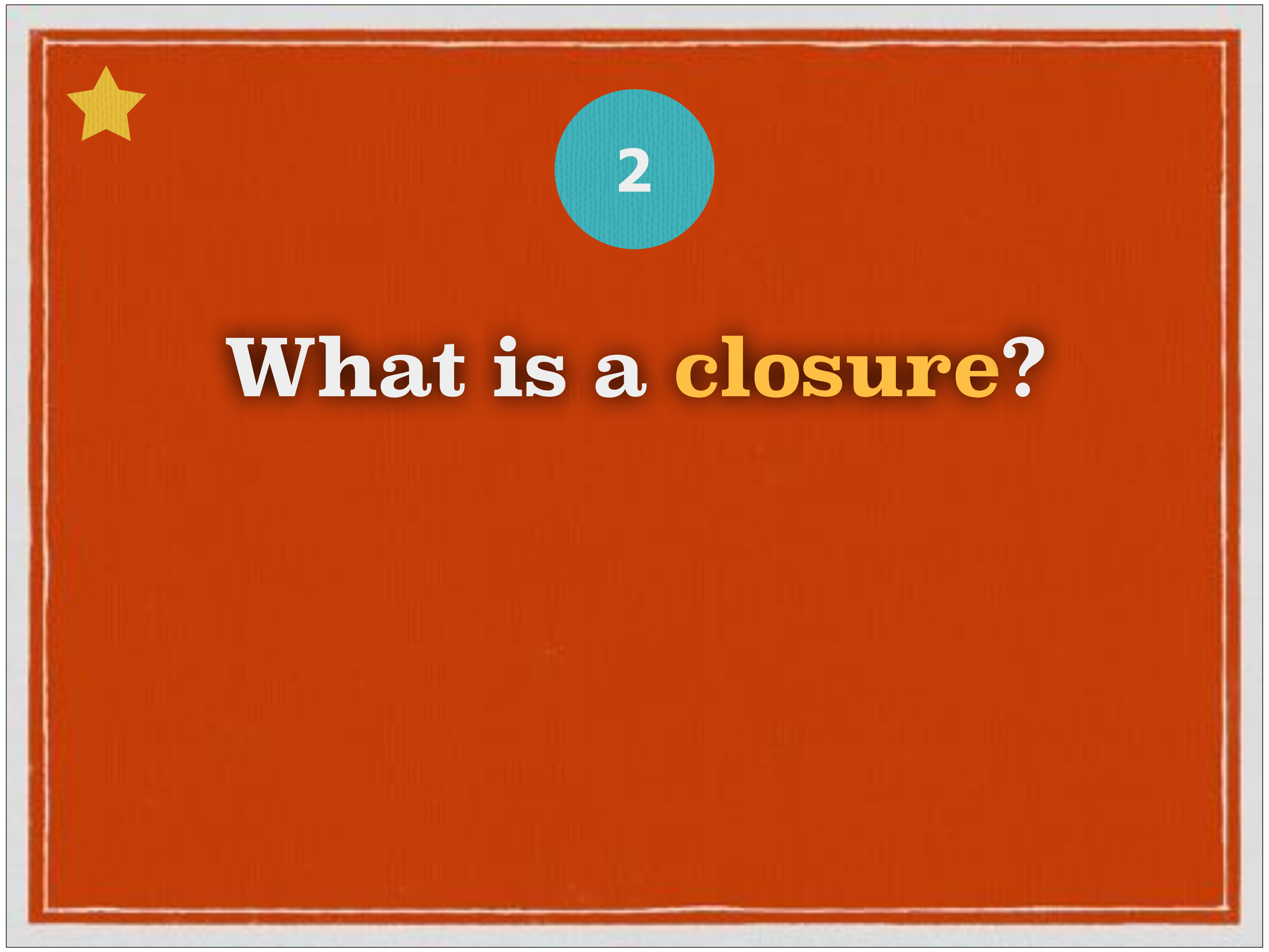
1

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.



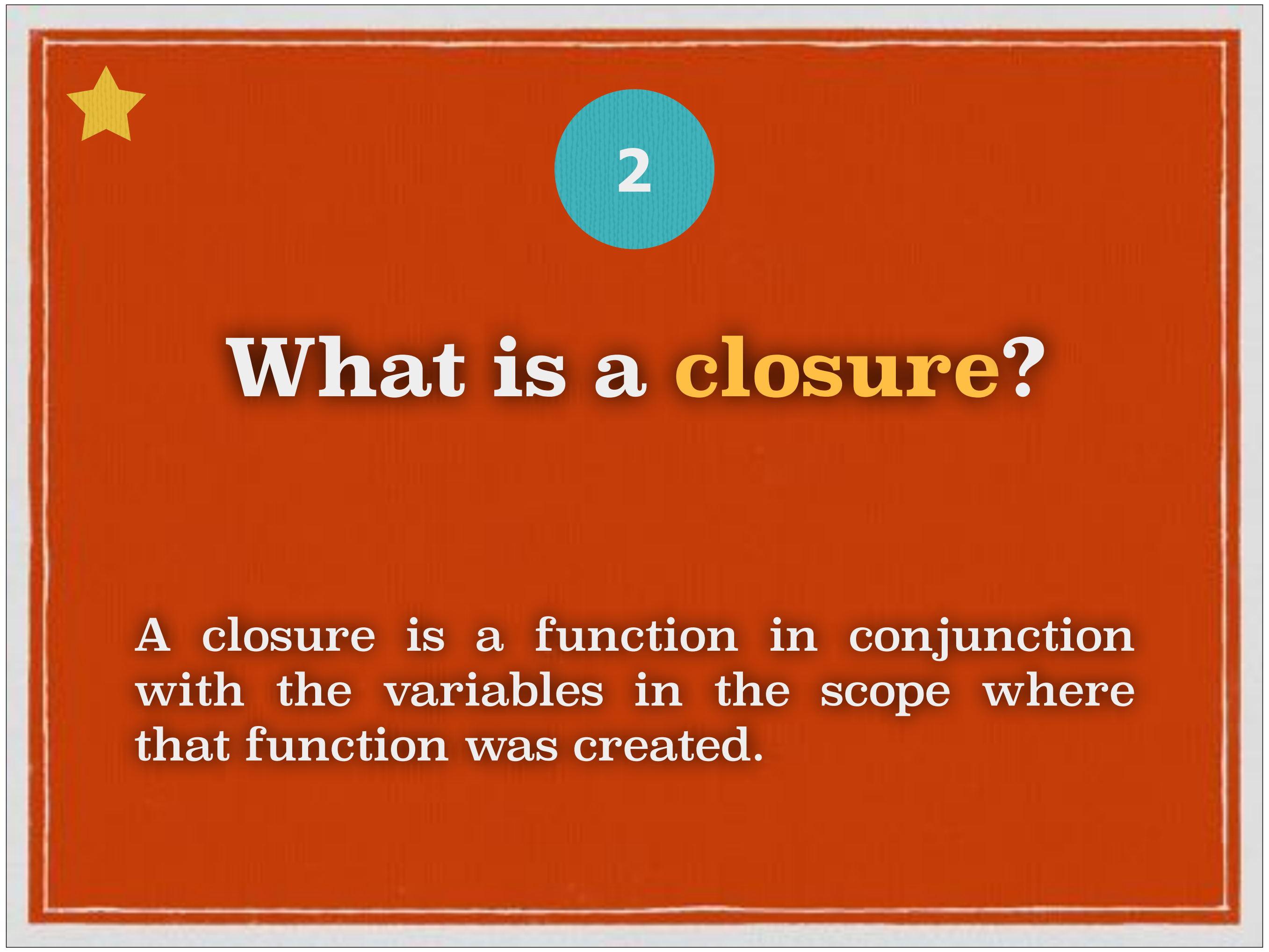
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures
http://repl.it/nLW/2



2

What is a closure?

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures
http://repl.it/nLW/2



2

What is a closure?

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

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures
http://repl.it/nLW/2

3

3

What is partial function application?

3

What is partial function application?

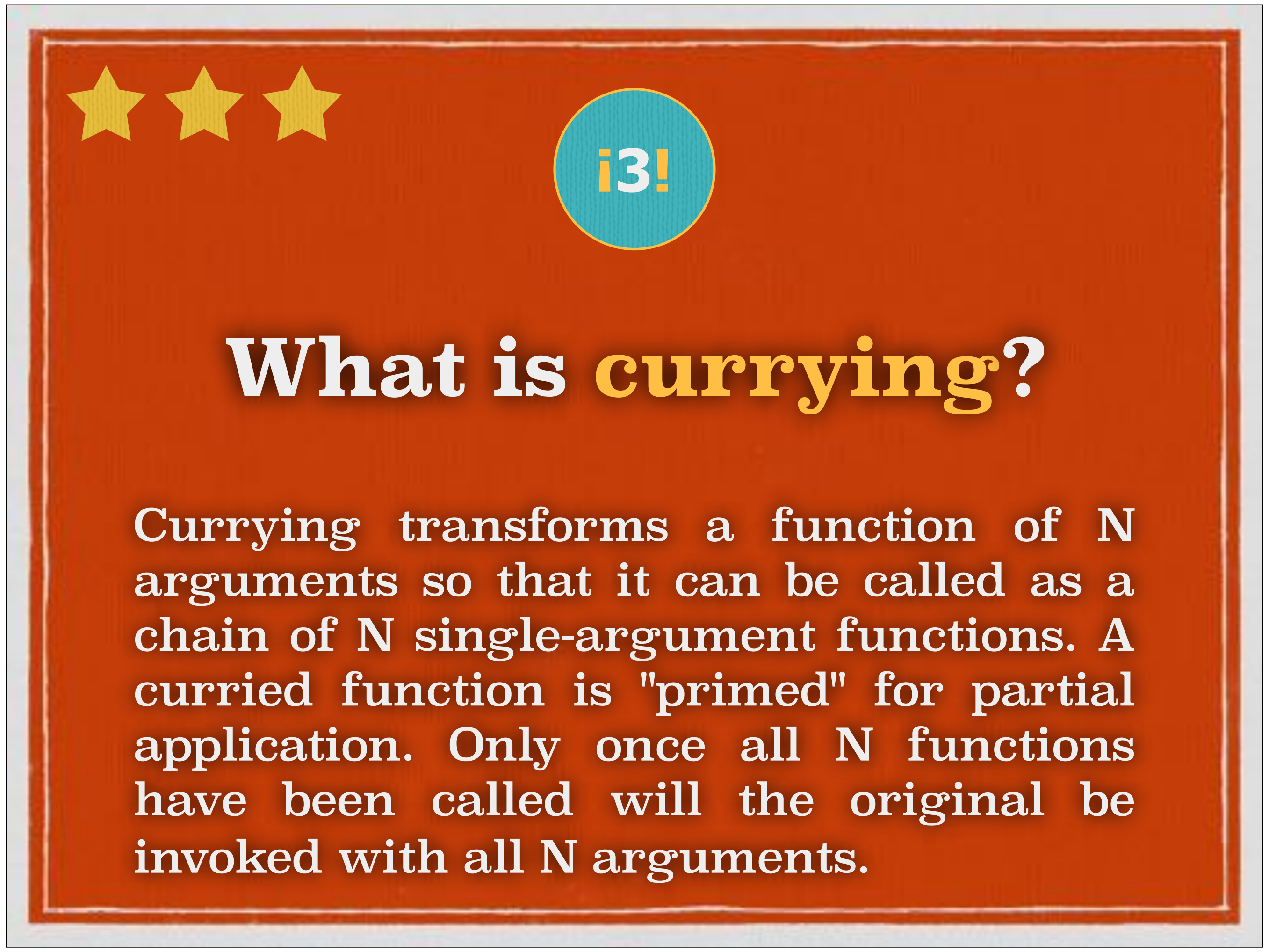
Take a function that accepts multiple arguments, bind values to some of them, and return a new function that only accepts the remaining arguments. When invoked, a partially applied function will invoke the original function and merge passed-in arguments.

http://tech.pro/tutorial/2011/functional-javascript-part-4-function-currying
http://repl.it/nVh/1

¡3!

What is currying?

http://tech.pro/tutorial/2011/functional-javascript-part-4-function-currying
http://repl.it/nVh/1

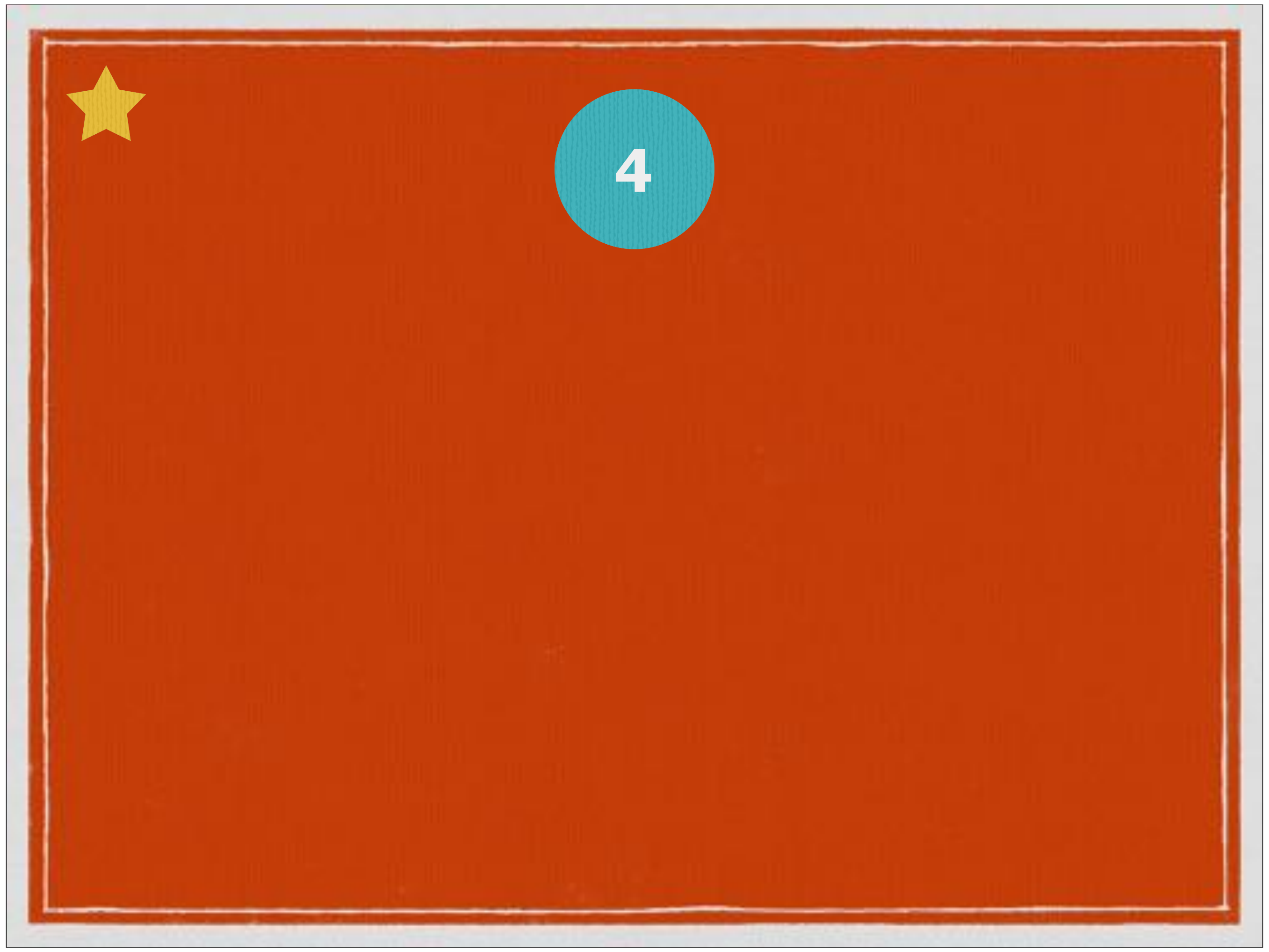


¡3!

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.

http://tech.pro/tutorial/2011/functional-javascript-part-4-function-currying
http://repl.it/nVh/1

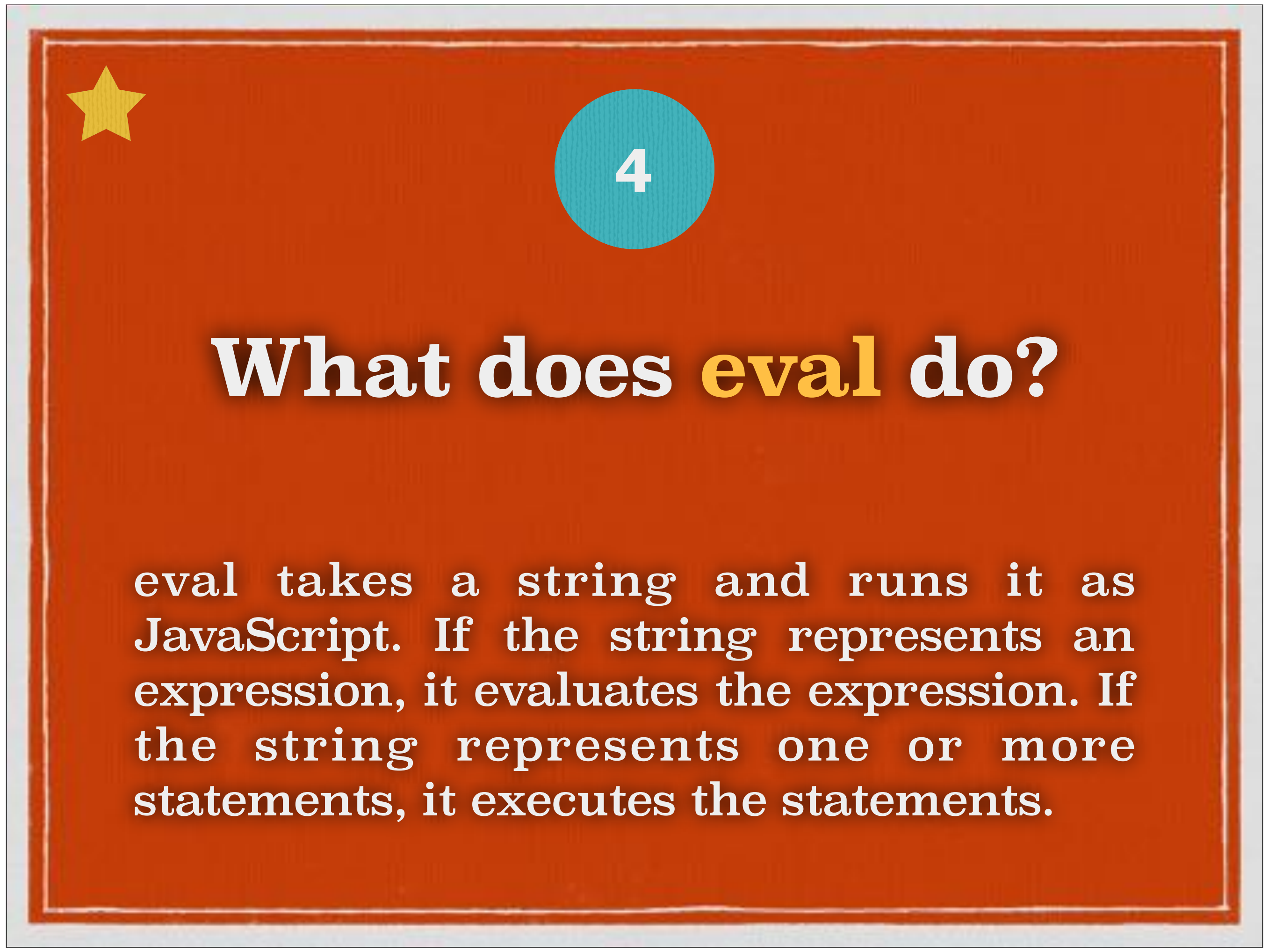


https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval
http://repl.it/nVq

4

What does eval do?

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval
http://repl.it/nVq

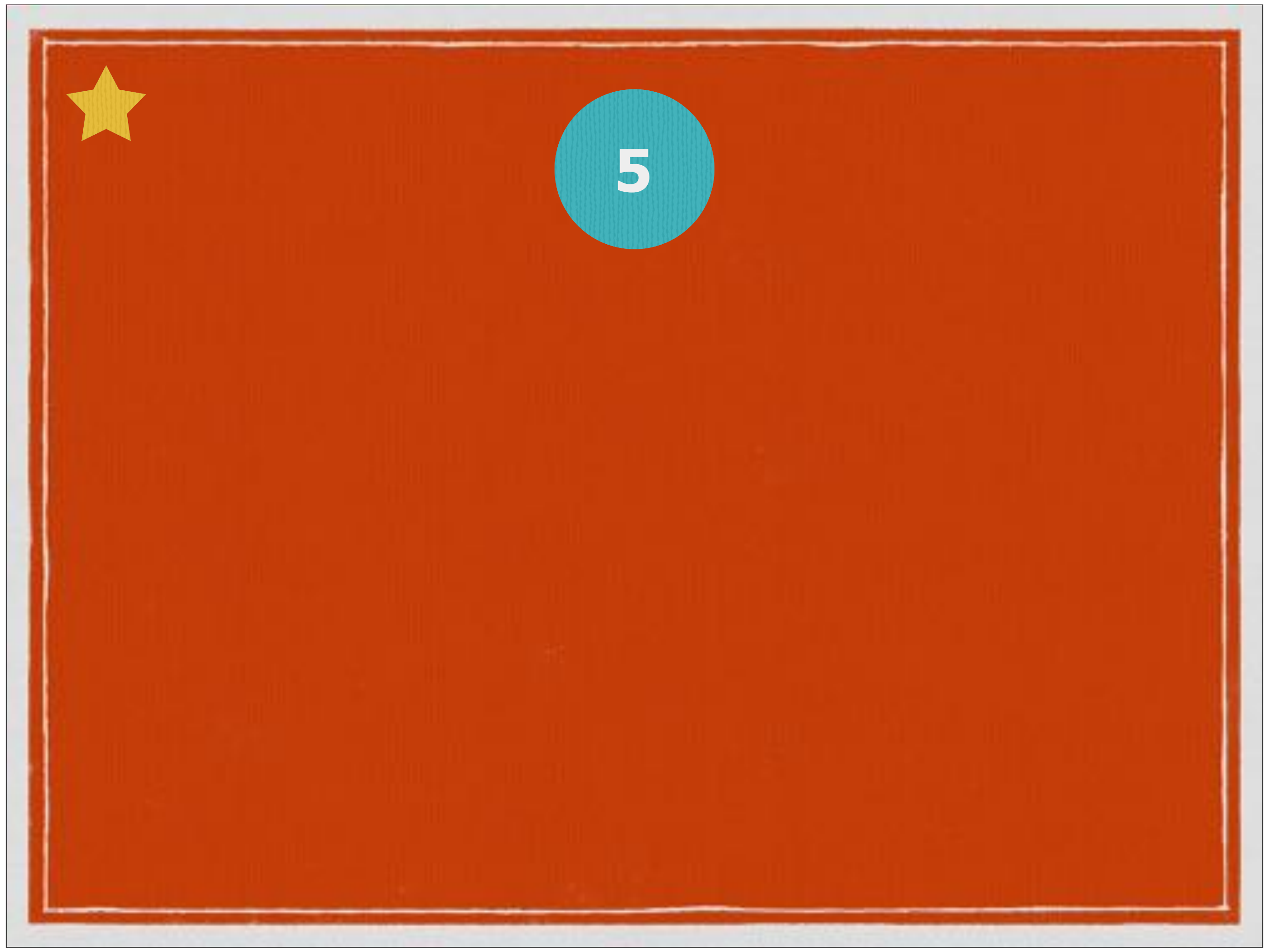


4

What does eval do?

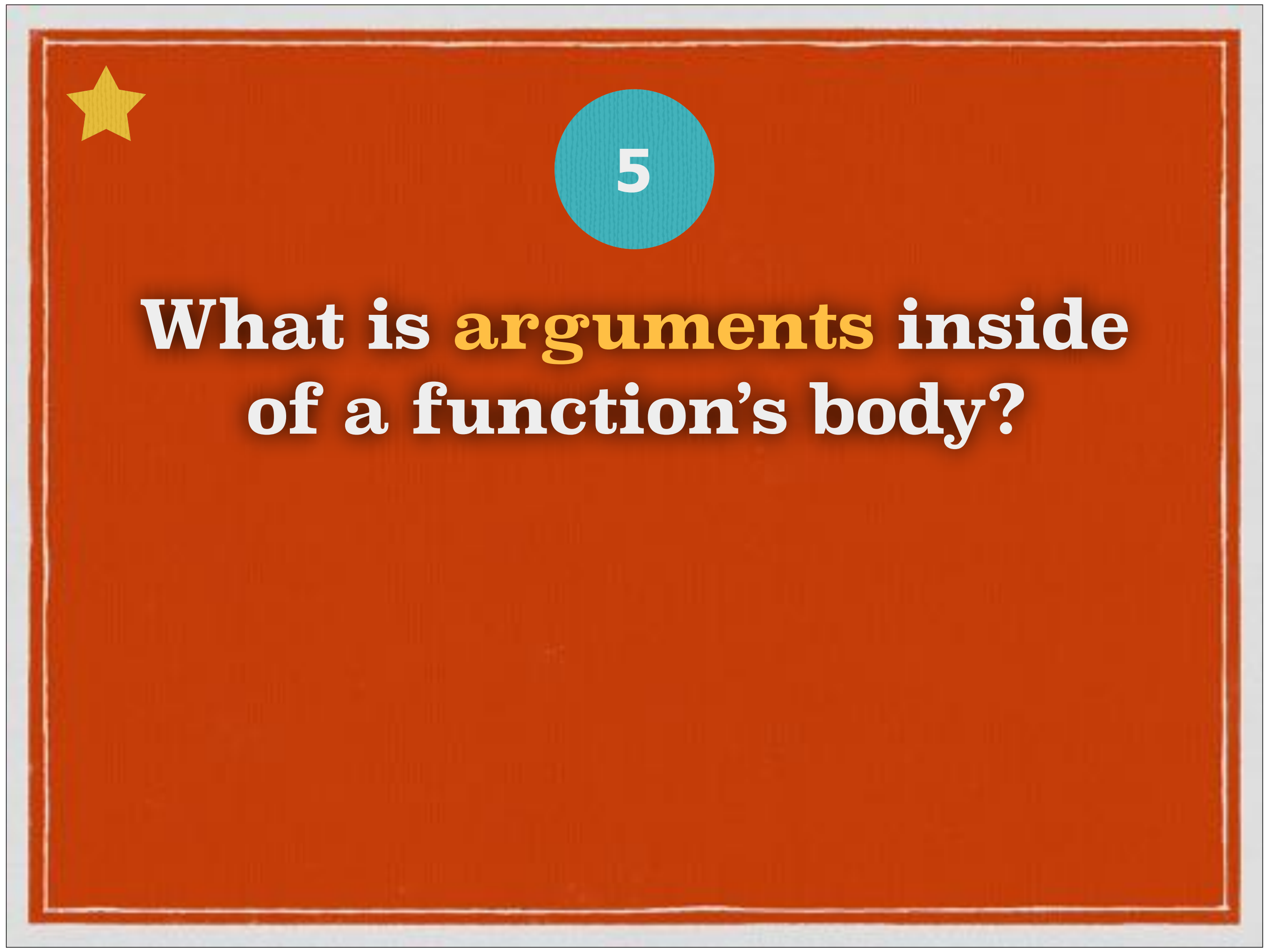
eval takes a string and runs it as JavaScript. If the string represents an expression, it evaluates the expression. If the string represents one or more statements, it executes the statements.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval
http://repl.it/nVq



5

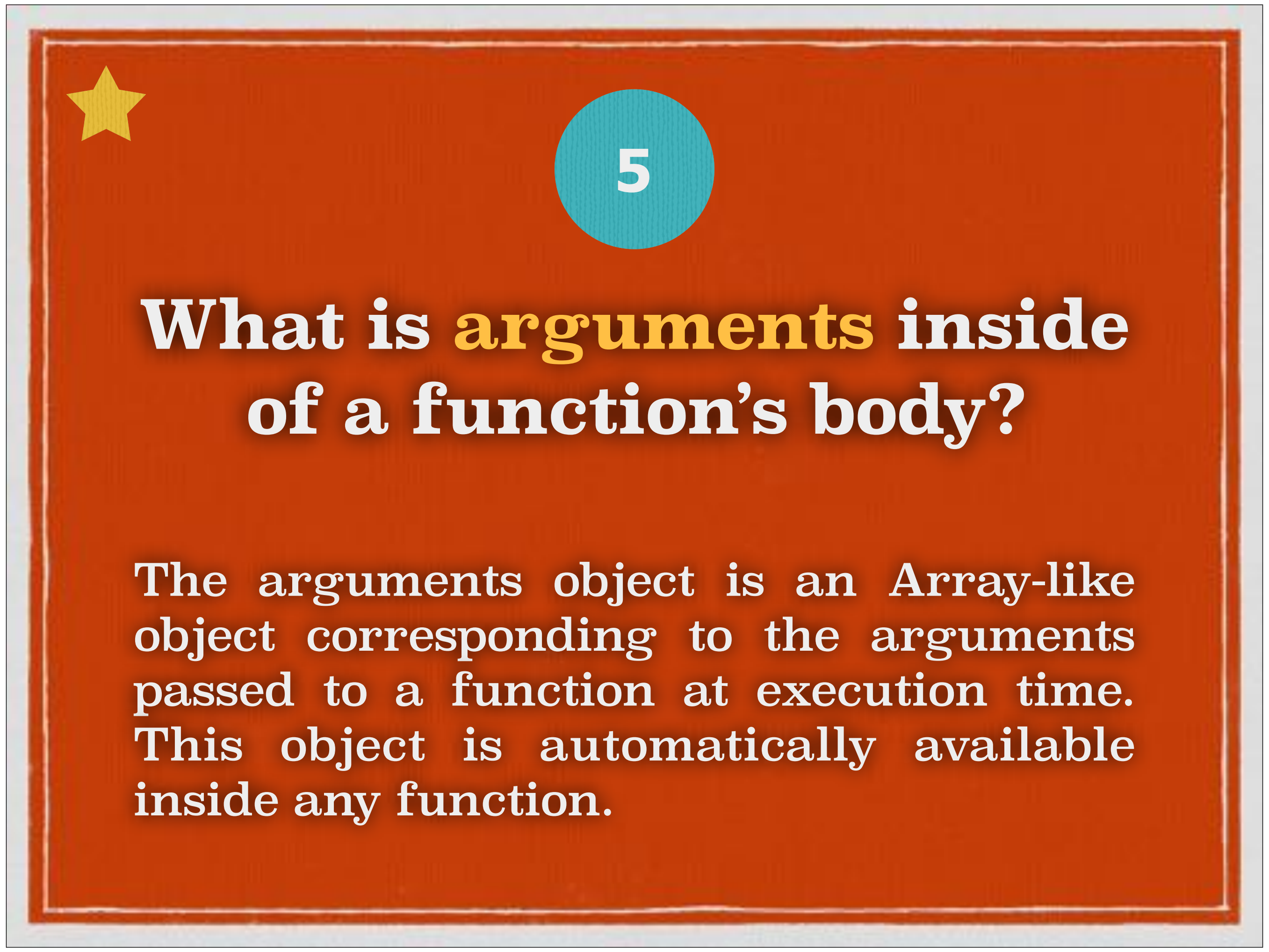
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/arguments
http://repl.it/nVs/1



5

What is arguments inside of a function's body?

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/arguments
http://repl.it/nVs/1



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.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/arguments
http://repl.it/nVs/1

6

6

What is **this** by default?

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.

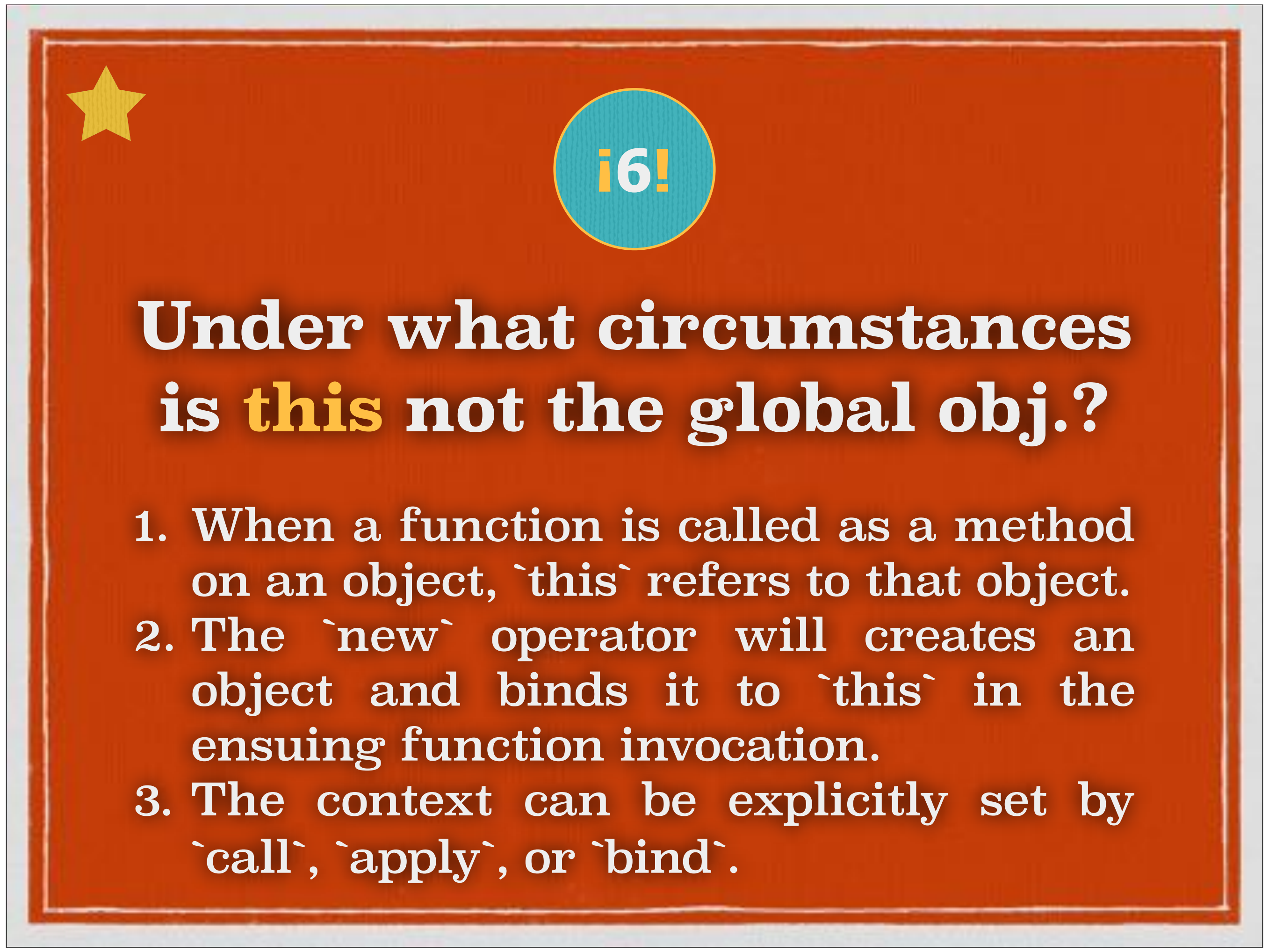
¡6!

http://tomhicks.github.io/code/2014/08/11/some-of-this.html
http://repl.it/nVs/4

¡6!

Under what circumstances is this not the global obj.?

http://tomhicks.github.io/code/2014/08/11/some-of-this.html
http://repl.it/nVs/4



¡6!

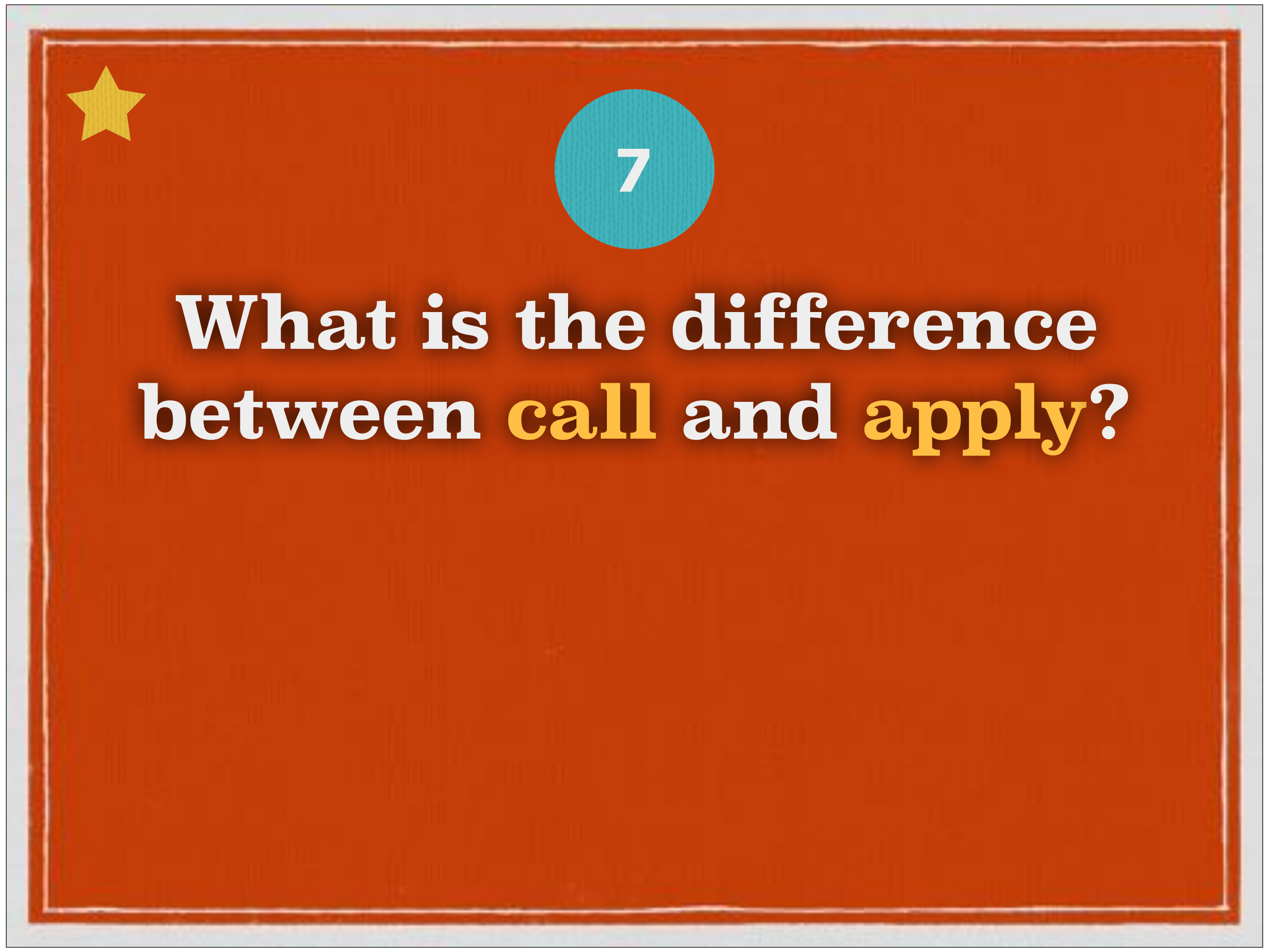
Under what circumstances is this not the global obj.?

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\`.

http://tomhicks.github.io/code/2014/08/11/some-of-this.html
http://repl.it/nVs/4

7

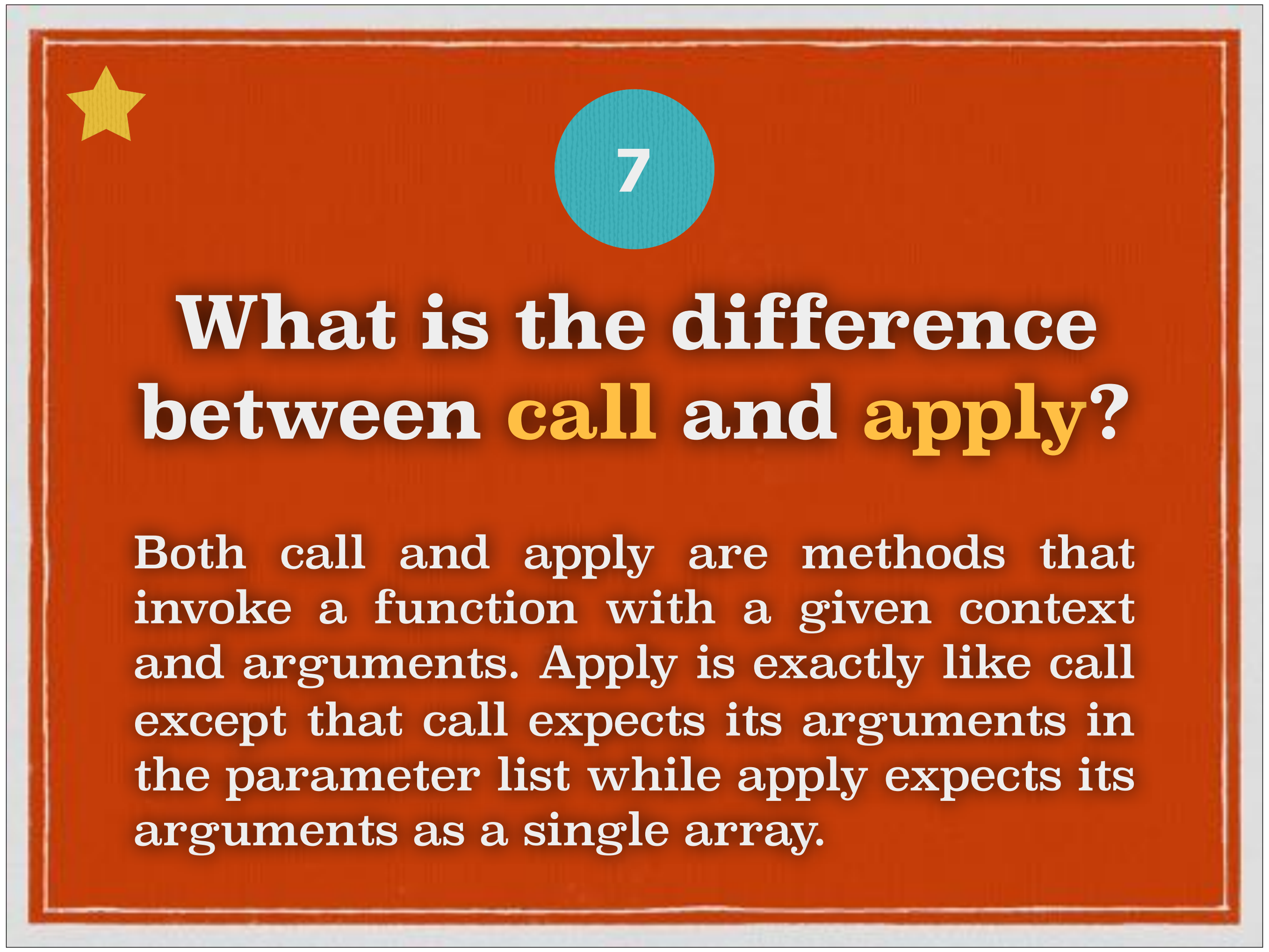
http://repl.it/nWF/1



7

What is the difference between call and apply?

http://repl.it/nWF/1



7

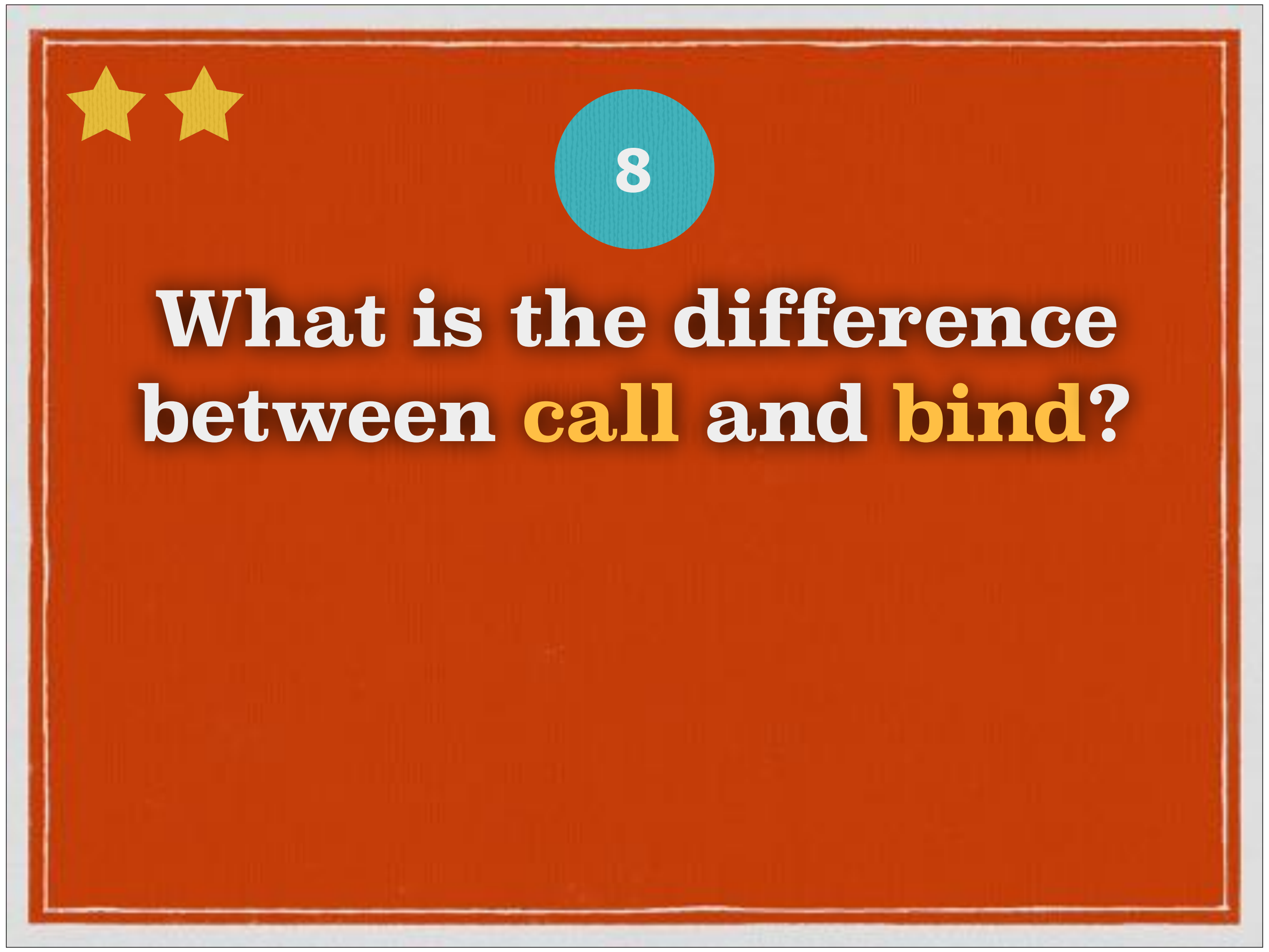
What is the difference between call and apply?

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.

http://repl.it/nWF/1

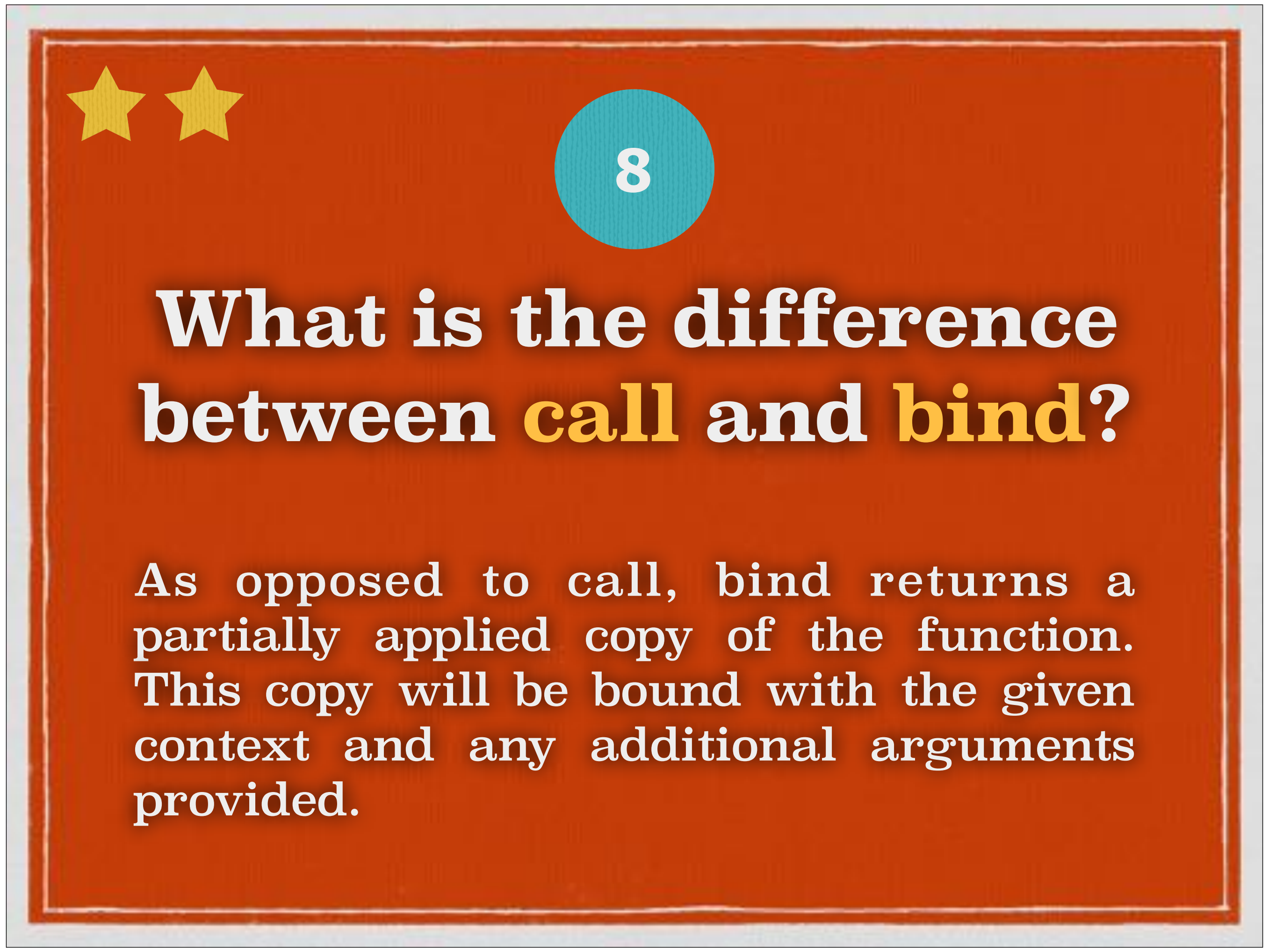
8

http://repl.it/nWI


8

What is the difference between call and bind?

http://repl.it/nWI



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.

http://repl.it/nWI

9

9

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

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

10

What is a getter?

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!

¡10!

How can you define a getter?

¡10!

How can you define a getter?

```
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

★★

11

Describe hoisting.

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

12

Define **IIFE**.

12

Define **IIFE**.

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

¡12!

¡12!

Valid IIFE: `(function (){})()`

Invalid IIFE: `function (){}()`

Why?

¡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

13

What does this log?

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

13

What does this log?

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

```
f
```

14

14

What is arity?

What is **arity**?

Arity is the number of explicitly declared arguments of a function.

¡14!

¡14!

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

¡14!

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

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

15

15

What are higher order functions?

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

16

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

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…
negative number: “a” should come first
positive number: “b” should come first
zero: neither takes precedence

17

17

Name three things a spy typically tracks.

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?

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!

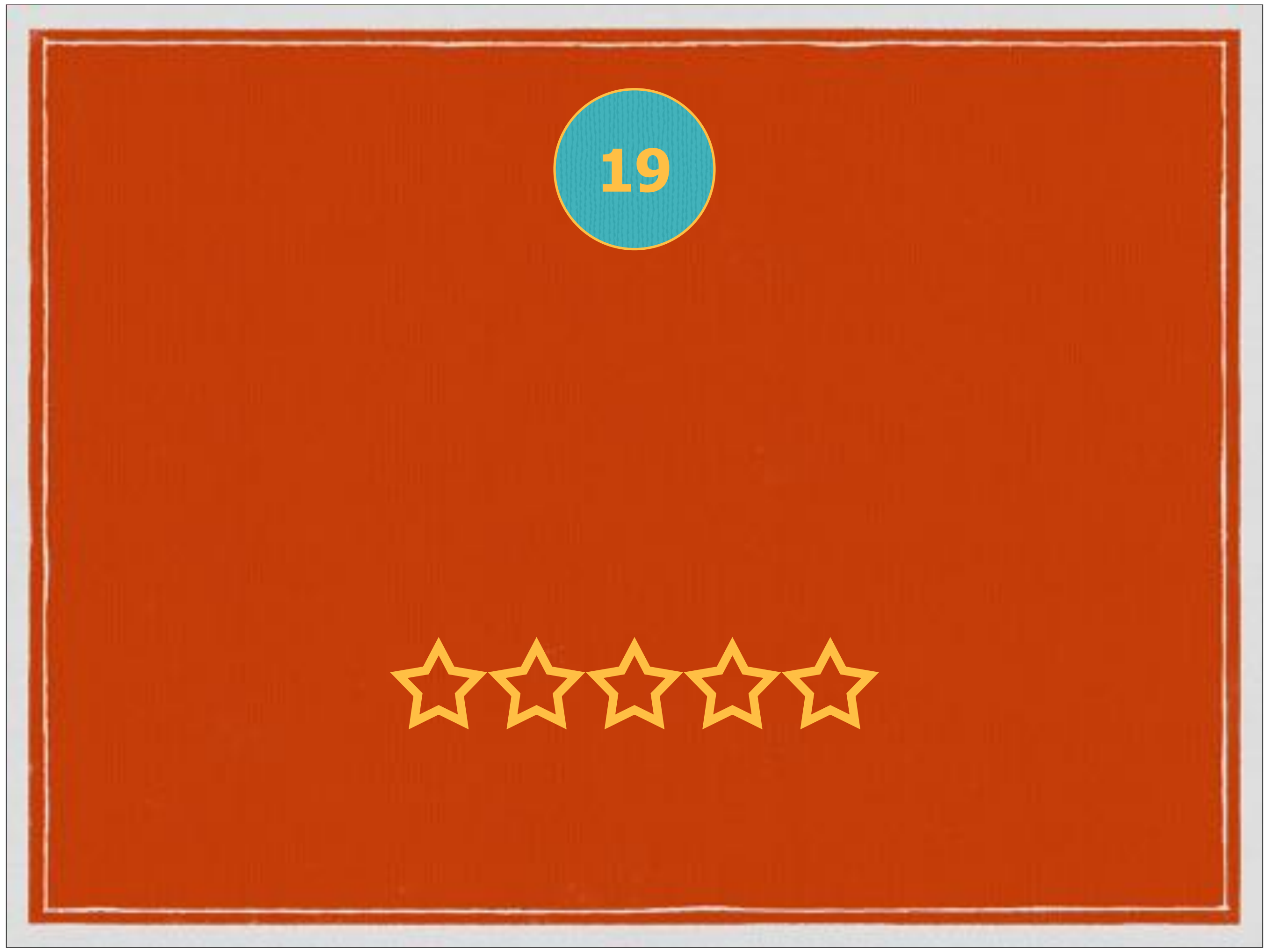
¡18!

What is a mock?

¡18!

What is a **mock**?

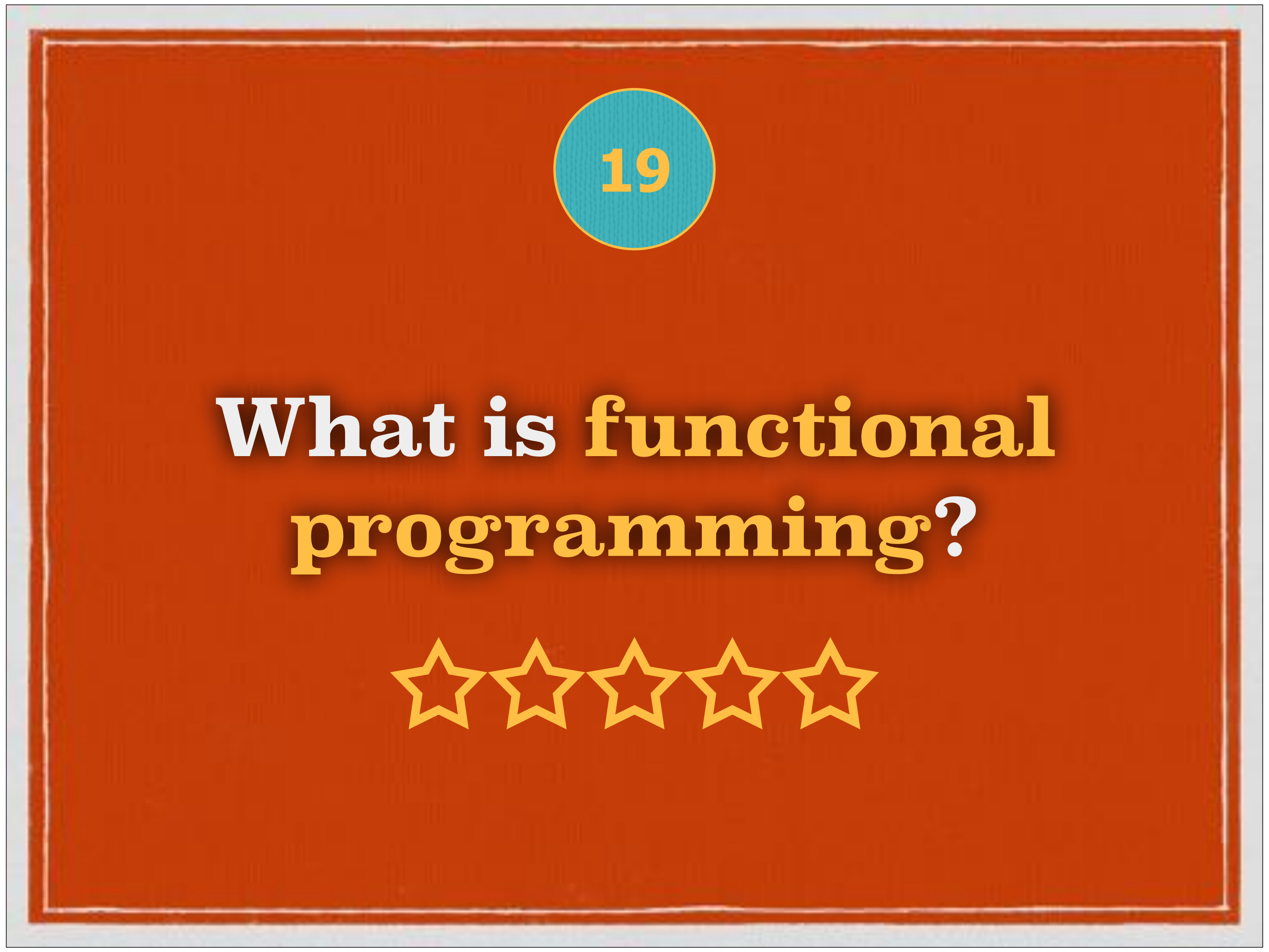
A mock is a stub with expectations built in. The expectations specify the calls it is expected to receive. A mock will cause a test spec to fail if these expectations are not met.



19

☆☆☆☆☆

* Statelessness
* No side effects
* Higher order
* First class
* Modularization
* Chaining
* Something about return vals?



19

What is functional programming?

☆☆☆☆☆

* Statelessness
* No side effects
* Higher order
* First class
* Modularization
* Chaining
* Something about return vals?