**1.**

/abc/.test('abcdefg'); //tests if abc is present in string resulting in true.

/[^0-9]/g.test('777'); // ???^ means not – so false

/\bis\b/.test('This is awesome!'); // ???looking for is the word ‘is’ so will be true.

* [More on Regular Expressions](http://eloquentjavascript.net/09_regexp.html)

**2.**

What will be logged and **why**?

var expected = { message: 'What did you expect?', expect: expect };

function expect()

{ return this.bind({

message: 'Expectations are powerful!' }).call({ message: 'Expecting this?' }); }

expected.expect.call(function () { console.log(this.message); // ??? });

Returns Expectations are powerful as .call binding does not overwrite the .bind binding. .bind returns a function so you can chain it.

**3.**

What will be logged and **why**?

var message = 'Hello';

var word = message.split('').reverse().join('');

console.log(word); // ???

Hello backwards i.e. olleH

**4.**

What will be logged and **why**?

(42).toFixed(3); // ??? 42.000

0.42.toFixed(3); // ???.042

42.toFixed(3) not able to be called as JS doesn’t know if it’s a number or not so will produce error.

**5.**

What will be logged and **why**?

console.log(0.1 + 0.2 === 0.3); //

1+2===3 is true but

this returns false! As floating point numbers are only approximations in JS.

Beware the floating point numbers.

0.1+0.2>0.3 is true

**6.**

Would it be correct to say?

* null is an empty value.
* undefined is a missing value.

Yes,

Null is a value therefore it has a type.

Undefined – not been assigned a value eg. Var a; a is undefined.

Undefined is an object (but this is a bug in JS, never been fixed).

**7.**

* Read about [void operator](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/void).

void 10; // ??? it returns undefined.

It returns any expression into undefined.

**8.**

What will be logged and **why**?

+'-0'; -0 in Chrome console, but 0 in Terminal as Chrome is more adept at JS.

**9.**

What will be logged and **why**?

var message = 'Ok';

var text = message;

text = 'Done';

console.log(message); // ???

Returns ok.

**10.**

What will be logged and **why**?

var messages = ['This', 'is', 'very', 'important'];

var words = messages;

words.pop();

console.log(messages.length);

returns 3(as pop chops one off the end).

Beware arrays are diff. Variables get a reference to an array so it can change the value.

**11.**

Implement keysToUpperCase function.

var data = { 'london': 1,

'san francisco': 10,

'new york': 2 };

console.log(data.keysToUpperCase()); // { 'LONDON': 1, 'SAN FRANCISCO': 10, 'NEW YORK': 2 }