

# Sample Code Test Questions

#### RUBY

1.

**How does a symbol differ from a string in Ruby? In what situations is it better to use a symbol rather than a string?**

Symbols are immutable and reusable, retaining the same object\_id. They are better memory-wise because they are only made once.

2.

**How do you declare a global variable in Ruby? When do you use global variables?**

Global variables are declared with the '\$' symbol and can be declared and used anywhere within your program. You should use them sparingly to never.

3.

**What is a class in Ruby? How do you create a class? What's the difference between a class and an instance?**

A class is the blueprint from which individual objects are created. An individual object is an instance.

```
class Student
  some stuff happens in here...
end
```

4.

**What are at least two ways to create getter and setter methods in Ruby?**

Setter and getter methods in Ruby are generated with the attr\_accessor method. You can also take the long route and create them manually.

5.

**Describe the difference between class and instance variables in Ruby?**

Class variables are created with the prefix @@ and are shared by all objects in a class.

Instance variables are created with the prefix @ and belong to a single object within a class.

6.

**What is a gem in Ruby and which are some of your favorites?**

Gems are packaged bits of Ruby code that you can install to extend or add functionality to your app.

7.

**What is the \$LOAD\_PATH magic constant in Ruby?**

It's all the places where 'require' will look for libraries. For more info, see this article and the one it references in the first paragraph:

<http://telladifferentstory.tumblr.com/post/41703448830/solved-mysteries-of-and-load-path>

8.

**What is the ENV magic constant in Ruby? Provide an example of when we would want to use this.**

It's a hash of environment variables from bash. Examples would be: protecting a user password or an API code.

9.

**What's the difference between a block and a proc in Ruby?**

A block is a syntactic chunk of code (inside either curly braces — `{ }` — or `do ... end`). A block is not an object. Because it is not an object, it cannot be passed to a variable.

A proc is an **object** containing a chunk of code. Since it's an object, you can pass a proc to a variable and then explicitly pass it to a method.

For more information, see:

<http://code.tutsplus.com/tutorials/ruby-on-rails-study-guide-blocks-procs-and-lambdas--net-29811>

10.

**Does Ruby have first-class functions? How about Javascript? Explain.**

Ruby does **not** have first-class functions: you can't pass a function to a variable in order to pass it around. You can pass the proc **object** around, but not a function.

Javascript does have first-class functions. In Javascript, a function is a first-class object. It can be: stored in a variable, passed as an argument to a function, created within a function, and returned from a function.

11.

**Explain the difference between a has\_one and belongs\_to association in Rails.**

`has_one`: Indicates a direct 1:1 relationship between objects where each instance of a model contains one instance of another model.

`belongs_to`: Represents the inverse of a `has_one` (or `has_many`) association.

## #### JAVASCRIPT

1.

### **What is the HTML DOM?**

When a web page loads, the browser generates the DOM, or Document Object Model, of the page. The DOM acts as the programming interface for HTML: it defines HTML properties, events, and methods. It also refers to HTML elements as objects. JavaScript relies on this DOM to alter the elements and attributes of a page

2.

### **How do you add JavaScript to a web page (list as many ways as possible)?**

CDN

**OR**

```
<script type="text/javascript" src="YourJavaScriptFile.js"></script>
```

**OR**

```
<script type="text/javascript">
    [javascript code here]
</script>
```

3.

### **What problem does `$(document).ready()` solve? What's the difference between `$(document).ready(function(){ ... });` and `$(function(){ ... });`?**

Code included inside `$( document ).ready()` will run as soon as the page's Document Object Model (DOM) is ready for JavaScript code to execute. All contents of the page need not be loaded.

There is no difference between the two expressions.

4.

### **How do you add comments in JavaScript?**

```
//
```

**OR**

```
/* blah blah blah */
```

### **What are the different JavaScript data types? Give an example of each.**

There are seven data types in JavaScript:

Undefined

Number

String

Boolean  
Object  
Function  
Null

5.

**What's the difference between `==` and `===` in JavaScript? How does this differ from `==` and `===` in Ruby?**

The equality (`==`) operator will compare for equality *after* doing necessary type casting, whereas the identity operator, or strict equality (`===`), doesn't do any conversions and will check if the things being compared have the same datatype. A good practice suggested by Douglas Crockford is to always use strict equality.

In Ruby `===` means "are they the EXACT SAME OBJECT" with the same object ID.,

6.

**When and where should you use `var` in a declaration and when do you not need it?**

Always use `var`. Not using `var` for variable declaration will traverse the scopes all the way up to the global scope. Then, if a variable with that name is not found, js will declare it in the global scope. So, not using `var` implicitly declares a variable in the global scope (which is bad practice).

7.

**Describe how scoping works in Javascript.**

JavaScript has lexical scoping based on functions but not blocks.

8.

**What's a closure? How would you make a closure in Javascript?**

In JavaScript, it's possible to have functions within functions. Closure refers to the system by which inner functions have access to the variables used in the outer function (the value is passed by reference), even after the outer function has executed completely.

9.

**What is hoisting in Javascript?**

Function declarations (`function foo(){}) and variable declarations (var name) are silently moved to the very top of the scope. So, always declare your variables at the top of the scope.`

10.

**Explain the `this` keyword in JavaScript.**

The `this` keyword in JavaScript is used to reference the object in which the function is operating.

11.

**What's the difference between a NAMED FUNCTION DECLARATION and an ANONYMOUS FUNCTION DEFINITION in Javascript?**

Named functions will show in a stack trace.

Named functions can be run as: `name()`;

Anonymous functions can go into a variable, which you can then pass around.

</END>