

JavaScript Assessment

Q1. Which operator returns true if the two compared values are not equal?

- <>
- ~
- ==!
- !== <<<---Correct

Q2. How is a forEach statement different from a for statement?

- Only a for statement uses a callback function
- A for statement is generic, but a forEach statement can be used only with an array <<<---Correct
- Only a forEach statement lets you specify your own iterator.
- A forEach statement is generic, but a for statement can be used only with an array

Q3. Review the code below. Which statement calls the addTax function and passes 50 as an argument?

```
function addTax(total){ return total * 1.05; }
```

- addTax = 50;
- return addTax 50;
- addTax(50); <<<---Correct
- addTax 50;

Q4. Which statement is the correct way to create a variable called rate and assign it the value 100?

- let rate = 100; <<<---Correct
- let 100 = rate;
- 100 = let rate;
- rate = 100;

Q5. Which statement creates a new object using the Person constructor?

- var student = new Person(); <<<---Correct

- `var student = construct Person;`
- `var student = Person();`
- `var student = construct Person();`

Q6. When would the final statement in the code shown be logged to the console?

```
let modal = document.querySelector('#result'); setTimeout(function(){
modal.classList.remove('hidden'); }, 10000); console.log('Results shown');
```

- after 10 second
- after results are received from the HTTP request
- after 10000 seconds
- immediately <<<<----Correct

Q7. You've written the code shown to log a set of consecutive values, but it instead results in the value 5, 5,

5, and 5 being logged to the console. Which revised version of the code would result in the value 1, 2, 3,

and 4 being logged?

- `for (var i=1; i<=4; i++){ setTimeout(function(){ console.log(i); }, i*1000); }`
- `for (var i=1; i<=4; i++){ (function(i){ setTimeout(function(){ console.log(j); }, j*1000); })(j) }`
- `while (var i=1; i<=4; i++) { setTimeout(function() { console.log(i); }, i*1000); }`
- `for (var i=1; i<=4; i++) { {function(j) { setTimeout(function(){ console.log(j); }, j*1000); }}(i) }`
- `for (var j=1; j<=4; j++) { setTimeout(function() { console.log(j); }, j*1000); } // Correct`

Q8. How does a function create a closure?

- It reloads the document whenever the value changes.
- It returns a reference to a variable in its parent scope <<<<---Correct
- It completes execution without returning
- It copies a local variable to the global scope

Q9. Which statement creates a new function called discountPrice?

- `let discountPrice = function(price) { return price * 0.85; };`
- `let discountPrice(price) { return price * 0.85; };`
- `let function = discountPrice(price) { return price * 0.85; };`
- `discountPrice = function(price) { return price * 0.85; };` <<<<----Correct

10. What is the result in the console of running the code shown?

```
var Storm = function() {}; Storm.prototype.precip = 'rain'; var WinterStorm = function() {}; WinterStorm.prototype = new Storm(); WinterStorm.prototype.precip = 'snow'; var bob = new WinterStorm(); console.log(bob.precip);
```

- `Storm()`
- `undefined`
- `'rain'`
- `'snow'` <<<---Correct

Q11. You need to match a time value such as 12:00:32. Which of the following regular expressions would work

for your code?

- `/[0-9]{2,}:[0-9]{2,}:[0-9]{2,}/`
- `^d\d:\d\d:\d\d/`
- `/[0-9]+:[0-9]+:[0-9]+/` <<<---Correct
- `/::/`

Q12. What is the result in the console of running this code?

```
"use strict"; function logThis() { this.desc = "logger"; console.log(this); } new logThis();
```

- `undefined`
- `window`
- `{desc: "logger"}` <<<---Correct
- `function`

Q13. How would you reference the text 'avenue' in the code shown?

```
let roadTypes = ['street', 'road', 'avenue', 'circle'];
```

- `roadTypes.2`

- roadTypes[3]
- roadTypes.3
- roadTypes[2] <<<---Correct

Q14. What is the result of running this statement?

```
console.log(typeof(42));
```

- 'float'
- 'value'
- 'number' <<<---Correct
- 'integer'

Q15. Which property references the DOM object that dispatched an event?

- self
- object
- target <<<---Correct
- source

Q16. You're adding error handling to the code shown. Which code would you include within the if statement to

specify an error message?

```
function addNumbers(x, y){ if (isNaN(x) || isNaN(y)) { } }
```

- exception('One or both parameters are not numbers')
- catch('One or both parameters are not numbers')
- error('One or both parameters are not numbers')
- throw('One or both parameters are not numbers') <<<---Correct

Q17. Which method converts JSON data to a JavaScript object?

- JSON.fromString();
- JSON.parse() <<<---Correct
- JSON.toObject()
- JSON.stringify()

Q18. When would you use a conditional statement?

- When you want to reuse a set of statements multiple times.
- When you want your code to choose between multiple options <<<---Correct
- When you want to group data together
- When you want to loop through a group of statement.

Q19. What would be the result in the console of running this code?

```
for (var i=0; i<5; i++){ console.log(i); }
```

- 12345
- 1234
- 01234 <<<<---Correct
- 012345

```
//
```

```
Q1
```

```
function trueOrNotTrue(first, second) {  
    if (first !== second) {  
        return true;  
    }  
}
```

```
console.log(trueOrNotTrue(5, 10));
```

```
// Q3
```

```
function addTax(total){  
    return total * 1.05;  
}
```

```
console.log(addTax(50));
```

```
// Q9
```

```
discountPrice = function(price) {  
    return price * 0.85;  
};
```

```
console.log(discountPrice(50));
```