JavaScript

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

- [ ] <>
- [ ] ~
- [ ] ==!
- [x] !==

Reference Javascript Comparison Operators

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

- [ ] Only a for statement uses a callback function.
- [x] A for statement is generic, but a forEach statement can be used only with an array.
- [ ] 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.

Reference Differences between forEach and for loop

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

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


- [x] `addTax(50);`
- [ ] `addTax 50;`

Reference functions in javascript

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

- [x] `let rate = 100;`
- [ ] `let 100 = rate;`
- [ ] `100 = let rate;`
- [ ] `rate = 100;`

Reference Javascript Assignment operators

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

- [x] `var student = new Person();`
- [ ] `var student = construct Person;`
- [ ] `var student = Person();`
- [ ] `var student = construct Person();`

Reference

Q6. When would the final statement in the code shown be logged to the console? When would 'results 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
- [x] immediately

Reference Javascript is synchronous and single threaded

Q7. Which snippet could you add to this code to print "food" to the console?

```
class Animal {
  static belly = [];
  eat() {
    Animal.belly.push('food');
  }
}
let a = new Animal();
a.eat();
console.log(/* Snippet Here */); //Prints food
```

- [ ] a.prototype.belly[0]
- [ ] Object.getPrototype0f (a).belly[0]
- [x] Animal.belly[0]
- [ ] a.belly[0]

Reference Javascript Class static Keyword

Q8. 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?

- [ ] A

```
for (var i = 1; i <= 4; i++) {
  setTimeout(function () {
    console.log(i);
  }, i * 10000);
}
```

- [ ] B

```
for (var i = 1; i <= 4; i++) {
  (function (i) {
    setTimeout(function () {
      console.log(j);
```

```
    }, j * 1000);
  })(j);
}
```

- [ ] C

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

- [x] D

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

- [ ] E

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

1. Reference setTimeout
2. Reference immediately invoked anonymous functions

Q9. How does a function create a closure?

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

Reference

Q10. Which statement creates a new function called discountPrice?

- [x] A

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

- [ ] B

```
let discountPrice(price) {
  return price * 0.85;
};
```

- [ ] C

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

- [x] D

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

Reference defining javascript functions

Q11. 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'

- [x] 'snow'

Reference prototype chain

Q12. 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,}/`
- [x] `/\d\d:\d\d:\d\d/`
- [ ] `/[0-9]+:[0-9]+:[0-9]+/`
- [ ] `/ : : /`

NOTE: The first three are all partially correct and will match digits, but the **second option is the most correct** because it will **only** match **2 digit** time values (12:00:32). The first option would have worked if the repetitions range looked like `[0-9]{2}`, however because of the **comma** `[0-9]{2,}` it will select 2 **or more** digits (120:000:321). The third option will any range of time digits, single *and* multiple (meaning `1:2:3` will also match).

More resources:

1. Repeating characters
2. Kleene operators

Q13. 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`
- [x] `{desc: "logger"}`
- [ ] `function`

Reference javascript classes

Q14. How would you reference the text 'avenue' in the code shown?

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

- [ ] roadTypes.2
- [ ] roadTypes[3]
- [ ] roadTypes.3
- [x] roadTypes[2]

Reference accessing javascript arrays

Q15. What is the result of running this statement?

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

- [ ] 'float'
- [ ] 'value'
- [x] 'number'
- [ ] 'integer'

Reference javascript data types

Q16. Which property references the DOM object that dispatched an event?

- [ ] self
- [ ] object
- [x] target
- [ ] source

Reference DOM events

Q17. 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')
- [x] throw('One or both parameters are not numbers')

Reference javascript throw

Q18. Which method converts JSON data to a JavaScript object?

- [ ] `JSON.fromString();`
- [x] `JSON.parse()`
- [ ] `JSON.toObject()`
- [ ] `JSON.stringify()`

Reference convert json to javascript object

Q19. When would you use a conditional statement?

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

Reference javascript conditionals

Q20. What would be the result in the console of running this code?

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

- [ ] 1 2 3 4 5
- [ ] 1 2 3 4
- [x] 0 1 2 3 4
- [ ] 0 1 2 3 4 5

Reference javascript for loops

Q21. Which Object method returns an iterable that can be used to iterate over the properties of an object?

- [ ] `Object.get()`
- [ ] `Object.loop()`
- [ ] `Object.each()`
- [x] `Object.keys()`

Reference javascript object static methods

Q22. What will be logged to the console?

```
var a = ['dog', 'cat', 'hen'];
a[100] = 'fox';
console.log(a.length);
```

- [x] 101
- [ ] 3
- [ ] 4
- [ ] 100

Q23. What is one difference between collections created with Map and collections created with Object?

- [ ] You can iterate over values in a Map in their insertion order.
- [x] You can count the records in a Map with a single method call.
- [ ] Keys in Maps can be strings.
- [ ] You can access values in a Map without iterating over the whole collection.

Explanation: `Map.prototype.size returns the number of elements in a Map, whereas Object does not have a built-in method to return its size.` Reference map methods javascript

Q24. What is the value of dessert.type after executing this code?

```
const dessert = { type: 'pie' };
dessert.type = 'pudding';
```

- [ ] pie
- [ ] The code will throw an error.
- [x] pudding
- [ ] undefined

Reference working with js objects

Q25. 0 && hi

- [ ] ReferenceError

- [ ] true
- [x] 0
- [ ] false

Reference boolean logic

Q26. Which of the following operators can be used to do a short-circuit evaluation?

- [ ] `++`
- [ ] `--`
- [ ] `==`
- [x] `||`

Reference short circuit javascript

Q27. Which statement sets the Person constructor as the parent of the Student constructor in the prototype chain?

- [ ] `Student.parent = Person;`
- [x] `Student.prototype = new Person();`
- [ ] `Student.prototype = Person;`
- [ ] `Student.prototype = Person();`

Reference prototype object js

Q28. Why would you include a "use strict" statement in a JavaScript file?

- [ ] to tell parsers to interpret your JavaScript syntax loosely
- [x] to tell parsers to enforce all JavaScript syntax rules when processing your code
- [ ] to instruct the browser to automatically fix any errors it finds in the code
- [ ] to enable ES6 features in your code

Reference what is use strict in js

Q29. Which Variable-defining keyword allows its variable to be accessed (as undefined) before the line that defines it?

- [ ] all of them
- [ ] `const`
- [x] `var`
- [ ] `let`

Reference var vs let vs const in js

Q30. Which of the following values is not a Boolean false?

- [ ] `Boolean(0)`
- [ ] `Boolean("")`
- [ ] `Boolean(NaN)`
- [x] `Boolean("false")`

Reference boolean of a string

Q31. Which of the following is not a keyword in JavaScript?

- [ ] `this`
- [ ] `catch`
- [ ] `function`
- [x] `array`

Reference js reserved words

Q32. Which variable is an implicit parameter for every function in JavaScript?

- [x] Arguments
- [ ] args
- [ ] argsArray
- [ ] argumentsList

Reference implicit js parameters for functions

Q33. For the following class, how do you get the value of 42 from an instance of X?

```
class X {
  get Y() {
    return 42;
  }
}
var x = new X();
```

- [ ] `x.get('Y')`
- [x] `x.Y`
- [ ] `x.Y()`

- [ ] `x.get().Y`

Reference getters

Q34. What is the result of running this code?

```
sum(10, 20);
diff(10, 20);
function sum(x, y) {
  return x + y;
}

let diff = function (x, y) {
  return x - y;
};
```

- [ ] 30, ReferenceError, 30, -10
- [x] 30, ReferenceError
- [ ] 30, -10
- [ ] ReferenceError, -10

Reference accessing before initialization

Q35. Why is it usually better to work with Objects instead of Arrays to store a collection of records?

- [ ] Objects are more efficient in terms of storage.
- [ ] Adding a record to an object is significantly faster than pushing a record into an array.
- [x] Most operations involve looking up a record, and objects can do that better than arrays.
- [ ] Working with objects makes the code more readable.

Reference efficiency of lookups

Explanation: Records in an object can be retrieved using their key which can be any given value (e.g. an employee ID, a city name, etc), whereas to retrieve a record from an array we need to know its index.

Q36. Which statement is true about the "async" attribute for the HTML script tag?

- [ ] It can be used for both internal and external JavaScript code.
- [ ] It can be used only for internal JavaScript code.
- [ ] It can be used only for internal or external JavaScript code that exports a promise.
- [x] It can be used only for external JavaScript code.

Reference async attribute for html

Q37. How do you import the lodash library making it top-level Api available as the "_" variable?

- [x] `import _ from 'lodash';`
- [ ] `import 'lodash' as _;`
- [ ] `import '_' from 'lodash;`
- [ ] `import lodash as _ from 'lodash';`

Reference how to import library in js

Q38. What does the following expression evaluate to?

```
[] == [];
```

- [ ] true
- [ ] undefined
- [ ] []
- [x] false

Reference arrays in js are objects

Q39. What type of function can have its execution suspended and then resumed at a later point?

- [x] Generator function
- [ ] Arrow function
- [ ] Async/ Await function
- [ ] Promise function

Reference what are generators in nodejs

Q40. What will this code print?

```
var v = 1;
var f1 = function () {
  console.log(v);
};

var f2 = function () {
  var v = 2;
```

```
  f1();
};

f2();
```

- [ ] 2
- [x] 1
- [ ] Nothing - this code will throw an error.
- [ ] undefined

Reference closures in js / nested functions

Q41. Which statement is true about Functional Programming?

- [ ] Every object in the program has to be a function.
- [ ] Code is grouped with the state it modifies.
- [ ] Date fields and methods are kept in units.
- [x] Side effects are not allowed.

Reference functional programming

Q42. Your code is producing the error: TypeError: Cannot read property 'reduce' of undefined. What does that mean?

- [x] You are calling a method named reduce on an object that's declared but has no value.
- [ ] You are calling a method named reduce on an object that does not exist.
- [ ] You are calling a method named reduce on an empty array.
- [ ] You are calling a method named reduce on an object that's has a null value.

Explanation: `You cannot invoke reduce on undefined object... It will throw (yourObject is not Defined...)`

Q43. How many prototype objects are in the chain for the following array?

```
let arr = [];
```

- [ ] 3
- [x] 2
- [ ] 0
- [ ] 1

Reference array prototype

Q44. Which choice is *not* a unary operator?

- [ ] `typeof`
- [ ] `delete`
- [x] `instanceof`
- [ ] `void`

Reference js unary operators

Q45. What type of scope does the end variable have in the code shown?

```
var start = 1;
if (start === 1) {
  let end = 2;
}
```

- [ ] conditional
- [x] block
- [ ] global
- [ ] function

Reference block vs function scope

Q46. What will the value of y be in this code:

```
const x = 6 % 2;
const y = x ? 'One' : 'Two';
```

- [ ] One
- [ ] undefined
- [ ] TRUE
- [x] Two

Reference ternary operator js

Q47. Which keyword is used to create an error?

- [x] `throw`
- [ ] `exception`
- [ ] `catch`

- [ ] error

Reference throwing errors in js

Q48. What's one difference between the async and defer attributes of the HTML script tag?

- [ ] The defer attribute can work synchronously.
- [ ] The defer attribute works only with generators.
- [ ] The defer attribute works only with promises.
- [x] The defer attribute will asynchronously load the scripts in order.

Reference async vs defer

Q49. The following program has a problem. What is it?

```
var a;
var b = (a = 3) ? true : false;
```

- [x] The condition in the ternary is using the assignment operator.
- [ ] You can't define a variable without initializing it.
- [ ] You can't use a ternary in the right-hand side of an assignment operator.
- [ ] The code is using the deprecated var keyword.

Reference ternary operator js

Q50. Which statement references the DOM node created by the code shown?

```
<p class="pull">lorem ipsum</p>
```

- [ ] Document.querySelector('class.pull')
- [x] document.querySelector('.pull');
- [ ] Document.querySelector('pull')
- [ ] Document.querySelector('#pull')

Reference query selector

Q51. What value does this code return?

```
let answer = true;
if (answer === false) {
```

```
  return 0;
} else {
  return 10;
}
```

- [x] 10
- [ ] true
- [ ] false
- [ ] 0

Reference javascript conditionals

Q52. What is the result in the console of running the code shown?

```
var start = 1;
function setEnd() {
  var end = 10;
}
setEnd();
console.log(end);
```

- [ ] 10
- [ ] 0
- [x] ReferenceError
- [ ] undefined

Reference

Q53. What will this code log in the console?

```
function sayHello() {
  console.log('hello');
}

console.log(sayHello.prototype);
```

- [ ] undefined
- [ ] "hello"
- [x] an object with a constructor property
- [ ] an error message

Reference prototypes

Q54. Which collection object allows unique value to be inserted only once?

- [ ] Object
- [x] Set
- [ ] Array
- [ ] Map

Reference javascript sets

Q55. What two values will this code print?

```
function printA() {
  console.log(answer);
  var answer = 1;
}
printA();
printA();
```

- [ ] `1` then `1`
- [ ] `1` then `undefined`
- [x] `undefined` then `undefined`
- [ ] `undefined` then `1`

Reference

Q56. How does the `forEach()` method differ from a `for` statement?

- [ ] forEach allows you to specify your own iterator, whereas for does not.
- [ ] forEach can be used only with strings, whereas for can be used with additional data types.
- [x] forEach can be used only with an array, whereas for can be used with additional data types.
- [ ] for loops can be nested; whereas forEach loops cannot.

Reference Differences between forEach and for loop

Q57. Which choice is an incorrect way to define an arrow function that returns an empty object?

- [ ] `=> ({})`

- [x] `=> {}`
- [ ] `=> { return {};}`
- [ ] `=> (({}))`

Reference arrow functions

Q58. Why might you choose to make your code asynchronous?

- [x] to start tasks that might take some time without blocking subsequent tasks from executing immediately
- [ ] to ensure that tasks further down in your code are not initiated until earlier tasks have completed
- [ ] to make your code faster
- [ ] to ensure that the call stack maintains a LIFO (Last in, First Out) structure

EXPLANATION: `"to ensure that tasks further down in your code are not initiated until earlier tasks have completed" you use the normal (synchronous) flow where each command is executed sequentially. Asynchronous code allows you to break this sequence: start a long running function (AJAX call to an external service) and continue running the rest of the code in parallel.`

Q59. Which expression evaluates to true?

- [ ] `[3] == [3]`
- [x] `3 == '3'`
- [ ] `3 != '3'`
- [ ] `3 === '3'`

1. Reference booleans
2. Reference 2 - booleans

Q60. Which of these is a valid variable name?

- [ ] 5thItem
- [x] firstName
- [ ] grand total
- [ ] function

Reference coding conventions

Q61. Which method cancels event default behavior?

- [ ] `cancel()`
- [ ] `stop()`
- [x] `preventDefault()`
- [ ] `prevent()`

Reference javascript events

Q62. Which method do you use to attach one DOM node to another?

- [ ] `attachNode()`
- [ ] `getNode()`
- [ ] `querySelector()`
- [x] `appendChild()`

Reference Node interface

Q63. What statement can be used to skip an iteration in a loop?

- [ ] `break`
- [ ] `pass`
- [ ] `skip`
- [x] `continue`

Reference break vs continue

Q64. Which choice is a valid example for an arrow function?

- [x] `(a,b) => c`
- [ ] `a, b => {return c;}`
- [ ] `a, b => c`
- [ ] `{ a, b } => c`

Reference arrow functions

Q65. Which concept is defined as a template that can be used to generate different objects that share some shape and/or behavior?

- [x] class
- [ ] generator function
- [ ] map
- [ ] proxy

Reference javascript classes

Q66. How do you add a comment to JavaScript code?

- [ ] `! This is a comment`
- [ ] `# This is a comment`
- [ ] `\\ This is a comment`
- [x] `// This is a comment`

Reference comments in javascript

Q67. If you attempt to call a value as a function but the value is not a function, what kind of error would you get?

- [x] TypeError
- [ ] SystemError
- [ ] SyntaxError
- [ ] LogicError

Reference javascript errors

Q68. Which method is called automatically when an object is initialized?

- [ ] create()
- [ ] new()
- [x] constructor()
- [ ] init()

Reference javascript constructors

Q69. What is the result of running the statement shown?

```
let a = 5;
console.log(++a);
```

- [ ] 4
- [ ] 10
- [x] 6
- [ ] 5

Reference ++x vs x++

Q70. You've written the event listener shown below for a form button, but each time you click the button, the page reloads. Which statement would stop this from happening?

```
button.addEventListener(
  'click',
  function (e) {
    button.className = 'clicked';
  },
  false,
);
```

- [ ] `e.blockReload();`
- [ ] `button.preventDefault();`
- [ ] `button.blockReload();`
- [x] `e.preventDefault();`

Reference events in javascript

Q71. Which statement represents the starting code converted to an IIFE?

- [ ] `function() { console.log('lorem ipsum'); }()();`
- [ ] `function() { console.log('lorem ipsum'); }();`
- [x] `(function() { console.log('lorem ipsum'); })();`

Reference what is an Immediately Invoked Function Expression

Q72. Which statement selects all img elements in the DOM tree?

- [ ] `Document.querySelector('img')`
- [ ] `Document.querySelectorAll('<img>')`
- [x] `Document.querySelectorAll('img')`
- [ ] `Document.querySelector('<img>')`

Reference query selector

Q73. Why would you choose an asynchronous structure for your code?

- [ ] To use ES6 syntax
- [x] To start tasks that might take some time without blocking subsequent tasks from executing immediately
- [ ] To ensure that parsers enforce all JavaScript syntax rules when processing your code

- [ ] To ensure that tasks further down in your code aren't initiated until earlier tasks have completed

Reference async function

Q74. What is the HTTP verb to request the contents of an existing resource?

- [ ] DELETE
- [x] GET
- [ ] PATCH
- [ ] POST

Reference http methods

Q75. Which event is fired on a text field within a form when a user tabs to it, or clicks or touches it?

- [x] focus
- [ ] blur
- [ ] hover
- [ ] enter

Reference javascript events

Q76. What is the result in the console of running this code?

```
function logThis() {
  console.log(this);
}
logThis();
```

- [ ] function
- [ ] undefined
- [ ] Function.prototype
- [x] window

Reference what is the javascript window

Q77. Which class-based component is equivalent to this function component?

```
const Greeting = ({ name }) => <h1>Hello {name}!</h1>;
```

- [x] `class Greeting extends React.Component { render() { return <h1>Hello {this.props.name}!</h1>; } }`
- [ ] `class Greeting extends React.Component { constructor() { return <h1>Hello {this.props.name}!</h1>; } }`
- [ ] `class Greeting extends React.Component { <h>Hello {this.props.name}!</h>; } }`
- [ ] `class Greeting extends React.Component { render({ name }) { return <h1>Hello {name}!</h1>; } }`

Q78. Which class-based lifecycle method would be called at the same time as this effect Hook?

```
useEffect(() => {
  // do things
}, []);
```

- [ ] componentWillUnmount
- [ ] componentDidUpdate
- [ ] render
- [x] componentDidMount

Reference

Q79. What is the output of this code?

```
var obj;
console.log(obj);
```

- [ ] `ReferenceError: obj is not defined`
- [ ] `{}`
- [x] `undefined`
- [ ] `null`

Reference working with objects

Q80. How would you use the TaxCalculator to determine the amount of tax on $50?

```
class TaxCalculator {
  static calculate(total) {
    return total * 0.05;
```

```
  }
}
```

- [ ] calculate(50);
- [ ] new TaxCalculator().calculate($50);
- [x] TaxCalculator.calculate(50);
- [ ] new TaxCalculator().calculate(50);

Reference functions in javascript

Q81. What is wrong with this code?

```
const foo = {
  bar() {
    console.log('Hello, world!');
  },
  name: 'Albert',
  age: 26,
};
```

- [ ] The function bar needs to be defined as a key/value pair.
- [ ] Trailing commas are not allowed in JavaScript.
- [ ] Functions cannot be declared as properties of objects.
- [x] Nothing, there are no errors.

1. Reference functions in javascript
2. Reference working with objects

Q82. What will be logged to the console?

```
console.log('I');
setTimeout(() => {
  console.log('love');
}, 0);
console.log('Javascript!');
```

- [x] .

```
I
Javascript!
love
```

- [ ] .

```
love
I
Javascript!
```

- [ ] The output may change with each execution of code and cannot be determined.
- [ ] .

```
I
love
Javascript!
```

Reference https://developer.mozilla.org/en-US/docs/Web/API/setTimeout#reasons_for_delays_longer_than_specified especially see the 'late timeouts' section.

Q83. What will this code log to the console?

```
const foo = [1, 2, 3];
const [n] = foo;
console.log(n);
```

- [x] 1
- [ ] undefined
- [ ] NaN
- [ ] Nothing--this is not proper JavaScript syntax and will throw an error.

Reference array deconstruction

Q84. How do you remove the property name from this object?

```
const foo = {
  name: 'Albert',
};
```

- [ ] delete name from foo;
- [x] delete foo.name;
- [ ] del foo.name;

- [ ] remove foo.name;

Reference working with objects

Q85. What is the difference between the `map()` and the `forEach()` methods on the Array prototype?

- [ ] There is no difference.
- [ ] The `forEach()` method returns a single output value, whereas the `map()` method performs operation on each value in the array.
- [x] The map() methods returns a new array with a transformation applied on each item in the original array, whereas the `forEach()` method iterates through an array with no return value.
- [ ] The `forEach()` methods returns a new array with a transformation applied on each item in the original array, whereas the `map()` method iterates through an array with no return value.

1. Reference map
2. Reference Differences between forEach and for loop

Q86. Which concept does this code illustrate?

```
function makeAdder(x) {
  return function (y) {
    return x + y;
  };
}

var addFive = makeAdder(5);
console.log(addFive(3));
```

- [ ] overloading
- [ ] closure
- [x] currying
- [ ] overriding

Reference currying

Q87. Which tag pair is used in HTML to embed JavaScript?

- [x] `<script></script>`
- [ ] `<js></js>`

- [ ] `<javascript></javascript>`
- [ ] `<code></code>`

Reference add js to html file

Q88. If your app receives data from a third-party API, which HTTP response header must the server specify to allow exceptions to the same-origin policy?

- [ ] Security-Mode
- [x] Access-Control-Allow-Origin
- [ ] Different-Origin
- [ ] Same-Origin

Reference Cross-Origin Resource Sharing

Q90. What is the output of this code?

```
let rainForests = ['Amazon', 'Borneo', 'Cerrado', 'Congo'];
rainForests.splice(0, 2);
console.log(rainForests);
```

- [ ] `["Amazon","Borneo","Cerrado","Congo"]`
- [x] `["Cerrado", "Congo"]`
- [ ] `["Congo"]`
- [ ] `["Amazon","Borneo"]`

Reference array methods

Q91. Which missing line would allow you to create five variables(one,two,three,four,five) that correspond to their numerical values (1,2,3,4,5)?

```
const numbers = [1, 2, 3, 4, 5];
//MISSING LINE
```

- [x] `const [one,two,three,four,five]=numbers`
- [ ] `const {one,two,three,four,five}=numbers`
- [ ] `const [one,two,three,four,five]=[numbers]`
- [ ] `const {one,two,three,four,five}={numbers}`

Reference array destructuring

Q92. What will this code print?

```
const obj = {
  a: 1,
  b: 2,
  c: 3,
};

const obj2 = {
  ...obj,
  a: 0,
};

console.log(obj2.a, obj2.b);
```

- [ ] Nothing, it will throw an error
- [x] 0 2
- [ ] undefined 2
- [ ] undefined 2

Reference spread syntax es6

Q93. Which line could you add to this code to print "jaguar" to the console?

```
let animals = ['jaguar', 'eagle'];
//Missing Line
console.log(animals.pop()); //Prints jaguar
```

- [ ] `animals.filter(e => e === "jaguar");`
- [ ] `animals.reverse();`
- [ ] `animals.shift();`
- [x] `animals.pop();`

Reference Javascript Array pop()

shift() - removes the FIRST element of an array and returns the removed item.

pop() - removes the LAST element of an array and returns the removed item.

reverse() - reverses the order of the elements in an array.

filter() - get every element in the array that meets the condition.

Q94. What line is missing from this code?

```
//Missing Line
for (var i = 0; i < vowels.length; i++) {
  console.log(vowels[i]);
  //Each letter printed on a separate line as follows;
  //a
  //e
  //i
  //o
  //u
}
```

- [ ] `let vowels = "aeiou".toArray();`
- [ ] `let vowels = Array.of("aeiou");`
- [ ] `let vowels = {"a", "e", "i", "o", "u"};`
- [x] `let vowels = "aeiou";`

Reference working with arrays

Q95. What will be logged to the console?

```
const x = 6 % 2;
const y = x ? 'One' : 'Two';
console.log(y);
```

- [ ] undefined
- [ ] One
- [ ] true
- [x] Two

Note: `this question is same with Q46.` Reference ternary operator js

Q96. How would you access the word It from this multidimensional array?

```
let matrix = [["You","Can"],["Do","It"],["!","!","!"]];
```

- [ ] `matrix[1[2]]`
- [x] `matrix[1][1]`
- [ ] `matrix[1,2]`
- [ ] `matrix[1][2]`

Q97. What does this code do?

```
const animals = ['Rabbit', 'Dog', 'Cat'];
animals.unshift('Lizard');
```

- [x] It adds "Lizard" to the start of the animals array.
- [ ] It adds "Lizard" to the end of the animals array.
- [ ] It replaces "Rabbit" with "Lizard" in the animals array.
- [ ] It replaces "Cat" with "Lizard" in the animals array.

Reference working with arrays

Q98. What is the output of this code?

```
let x = 6 + 3 + '3';
console.log(x);
```

- [x] 93
- [ ] 12
- [ ] 66
- [ ] 633

Reference type coercion

Q99. Which statement can take a single expression as input and then look through a number of choices until one that matches that value is found?

- [ ] else
- [ ] when
- [ ] if
- [x] switch

Reference switch

Q100. Which statement prints "roar" to the console?

```
var sound = 'grunt';
var bear = { sound: 'roar' };
function roar() {
```

```
  console.log(this.sound);
}
```

- [ ] `bear.bind(roar);`
- [ ] `roar.bind(bear);`
- [x] `roar.apply(bear);`
- [ ] `bear[roar]();`

1. Reference Apply
2. Reference this
3. Reference bind

Q101. Which choice is a valid example of an arrow function, assuming c is defined in the outer scope?

- [ ] `a, b => { return c; }`
- [ ] `a, b => c`
- [ ] `{ a, b } => c`
- [x] `(a,b) => c`

Reference arrow functions

Q102. Which statement correctly imports this code from some-file.js?

```
//some-file.js
export const printMe = (str) => console.log(str);
```

- [ ] `import printMe from './some-file';`
- [x] `import { printMe } from './some-file';`
- [ ] `import default as printMe from './some-file';`
- [ ] `const printMe = import './some-file';`

Reference importing libraries in javascript

Q103. What will be the output of this code?

```
const arr1 = [2, 4, 6];
const arr2 = [3, 5, 7];

console.log([...arr1, ...arr2]);
```

- [ ] `[2, 3, 4, 5, 6, 7]`
- [ ] `[3,5,7,2,4,6]`
- [ ] `[3, 5, 7, 2, 4, 6]`
- [ ] `[[2, 4, 6], [3, 5, 7]]`
- [x] `[2, 4, 6, 3, 5, 7]`

Reference spread syntax

Q104. Which method call is chained to handle a successful response returned by `fetch()`?

- [ ] `done()`
- [x] `then()`
- [ ] `finally()`
- [ ] `catch()`

Reference fetch

Q105. Which choice is not an array method?

- [ ] `array.slice()`
- [ ] `array.shift()`
- [ ] `array.push()`
- [x] `array.replace()`

Reference working with arrays

Q106. Which JavaScript loop ensures that at least a singular iteration will happen?

- [x] do...while
- [ ] forEach
- [ ] while
- [ ] for

Reference loops in js

Q107. What will be logged to the console?

```
console.log(typeof 'blueberry');
```

- [x] `string`

- [ ] array
- [ ] Boolean
- [ ] object

Reference what is typeof

Q108. What is the output that is printed when the div containing the text "Click Here" is clicked?

```
//HTML Markup
<div id="A">
  <div id="B">
    <div id="C">Click Here</div>
  </div>
</div>
```

```
//JavaScript
document.querySelectorAll('div').forEach((e) => {
  e.onclick = (e) => console.log(e.currentTarget.id);
});
```

- [x] C B A
- [ ] A
- [ ] C
- [ ] A B C

1. Reference query selector
2. Reference events

Q109. What will this code log to the console?

```
const myNumbers = [1, 2, 3, 4, 5, 6, 7];
const myFunction = (arr) => {
  return arr.map((x) => x + 3).filter((x) => x < 7);
};
console.log(myFunction(myNumbers));
```

- [ ] [4,5,6,7,8,9,10]
- [ ] [4,5,6,7]
- [ ] [1,2,3,4,5,6]

- [x] [4,5,6]

Reference functions in javascript

Q110. What does this code print to the console?

```
let rainForestAcres = 10;
let animals = 0;

while (rainForestAcres < 13 || animals <= 2) {
  rainForestAcres++;
  animals += 2;
}

console.log(animals);
```

- [ ] 2
- [ ] 4
- [x] 6
- [ ] 8

Reference MDN JavaScript Looping code

Q111. Which snippet could you add to this code to print "YOU GOT THIS" to the console?

```
let cipherText = [...'YZOGUT QGMORTZ MTRHTILS'];
let plainText = '';

/* Missing Snippet */

console.log(plainText); //Prints YOU GOT THIS
```

- [ ] A

```
for (let key of cipherText.keys()) {
  plainText += key % 2 === 0 ? key : ' ';
}
```

- [ ] B

```
for (let [index, value] of cipherText.entries()) {
  plainText += index % 2 !== 0 ? value : '';
```

```
}
```

- [x] C

```
for (let [index, value] of cipherText.entries()) {
  plainText += index % 2 === 0 ? value : '';
}
```

- [ ] D

```
for (let value of cipherText) {
  plainText += value;
}
```

1. Reference MDN JavaScript Destructuring
2. Reference MDN JavaScript Array entries
3. Reference MDN JavaScript Remainder/Modulo

Q112. Which Pokemon will be logged to the console?

```
var pokedex = ['Snorlax', 'Jigglypuff', 'Charmander', 'Squirtle'];
pokedex.pop();
console.log(pokedex.pop());
```

- [x] Charmander
- [ ] Jigglypuff
- [ ] Snorlax
- [ ] Squirtle

Explanation: `The pop() method removes the last element from an array and returns that element. This method changes the length of the array.`

Reference Array.pop

Q113. Which statement can be used to select the element from the DOM containing the text "The LinkedIn Learning library has great JavaScript courses" from this markup?

```
  <span class="content">The LinkedIn Learning library has great JavaScript cou
</div>
```

- [ ] document.querySelector("div.content")
- [x] document.querySelector("span.content")
- [ ] document.querySelector(".content")
- [ ] document.querySelector("div.span")

Q114. Which value is not falsey?

- [x] `[]`
- [ ] `undefined`
- [ ] `0`
- [ ] `null`

Reference Falsy

Q115. What line of code causes this code segment to throw an error?

```
const lion = 1;
let tiger = 2;
var bear;

++lion;
bear += lion + tiger;
tiger++;
```

- [x] `line 5, because lion cannot be reassigned a value`
- [ ] `line 6, because the += operator cannot be used with the undefined variable bear`
- [ ] `line 5, because the prefix (++) operator does not exist in JavaScript`
- [ ] `line 3, because the variable bear is left undefined`

1. Reference const in js
2. Reference TypeError: invalid assignment to const "x"

Q116. What will be the value of `result` after running this code?

```
const person = { name: 'Dave', age: 40, hairColor: 'blue' };
const result = Object.keys(person).map((x) => x.toUpperCase());
```

- [ ] It will throw a TypeError.
- [ ] `["Name", "Age", "HairColor"]`
- [ ] `["DAVE", 40, "BLUE"]`
- [x] `["NAME", "AGE", "HAIRCOLOR"]`

1. Reference Object.keys()
2. Reference Array.prototype.map()
3. Reference String.prototype.toUpperCase()

Q117. Which snippet could you insert to this code to print "swim" to the console?

```
let animals = ["eagle", "osprey", "salmon"];
let key = animal => animal === "salmon";

if(/* Insert Snippet Here */){
  console.log("swim");
}
```

- [ ] `animals.every(key)`
- [ ] `animals.some(key).length === 1`
- [ ] `animals.filter(key) === true`
- [x] `animals.some(key)`

Reference Array.prototype.some

Q118. What is the output of this code?

```
class RainForest {
  static minimumRainFall = 60;
}

let congo = new RainForest();
RainForest.minimumRainFall = 80;
console.log(congo.minimumRainFall);
```

- [x] `undefined`
- [ ] `None of these answers, as static is not a feature in Javascript.`
- [ ] `60`
- [ ] `80`

Reference Classes static

Q119. How can you attempt to access the property `a.b` on `obj` without throwing an error if a is undefined?

```
let obj = {};
```

- [ ] `obj?.a.b`
- [x] `obj.a?.b`
- [ ] `obj[a][b]`
- [ ] `obj.?a.?b`

Reference Optional chaining (?.)

Q120. What happens when you run this code?

```
if (true) {
  var x = 5;
  const y = 6;
  let z = 7;
}
console.log(x + y + z);
```

- [ ] It will throw a `ReferenceError` about `x`.
- [ ] It will print `18`.
- [ ] It will print `undefined`.
- [x] It will throw a `ReferenceError` about `y`.

Reference let statement

Q121. What does this code print to the console?

```
const x = [1, 2];
const y = [5, 7];
const z = [...x, ...y];
console.log(z);
```

- [x] `[1,2,5,7]`
- [ ] `[[1, 2], [5, 7]]`
- [ ] `[2,7]`
- [ ] `[2,1,7,5]`

Reference spread syntax (...)

Q122. Given this code, which statement will evaluate to false?

```
const a = { x: 1 };
const b = { x: 1 };
```

- [ ] a['x'] === b['x']
- [ ] a != b
- [x] a === b
- [ ] a.x === b.x

Reference

Q123. What will this code log to the console?

```
console.log(typeof 41.1);
```

- [ ] Nothing. It resuults in a ReferenceError.
- [ ] decimal
- [ ] float
- [x] number

Reference

Q124. What is the output of this code?

```
let scores = [];
scores.push(1, 2);
scores.pop();
scores.push(3, 4);
scores.pop();
score = scores.reduce((a, b) => a + b);
console.log(score);
```

- [ ] 3
- [x] 4
- [ ] 6
- [ ] 7

1. Reference Array.prototype.push()
2. Reference Array.prototype.pop()

3. Reference Array.prototype.reduce()

Q125. What does this code print to the console?

```
let bear = {
  sound: 'roar',
  roar() {
    console.log(this.sound);
  },
};

bear.sound = 'grunt';
let bearSound = bear.roar;
bearSound();
```

- [ ] Nothing is printed to the console.
- [ ] grunt
- [x] undefined
- [ ] roar

Reference

Q126. What is the output of this code?

```
var cat = { name: 'Athena' };

function swap(feline) {
  feline.name = 'Wild';
  feline = { name: 'Tabby' };
}

swap(cat);
console.log(cat.name);
```

- [ ] undefined
- [x] Wild
- [ ] Tabby
- [ ] Athena

Q127. What will this code output to the log?

```
var thing;
let func = (str = 'no arg') => {
  console.log(str);
};
func(thing);
func(null);
```

- [ ] null no arg
- [ ] no arg no arg
- [ ] null null
- [x] no arg null

Q128. What will this code print to the console?

```
const myFunc = () => {
  const a = 2;
  return () => console.log('a is ' + a);
};
const a = 1;
const test = myFunc();
test();
```

- [ ] a is 1
- [ ] a is undefined
- [ ] It won't print anything.
- [x] a is 2

Q129. What will this code print to the console?

```
const myFunc = (num1, num2 = 2, num3 = 2) => {
  return num1 + num2 + num3;
};
let values = [1, 5];
const test = myFunc(2, ...values);
console.log(test);
```

- [x] 8
- [ ] 6
- [ ] 2
- [ ] 12

Q130. Which code would you use to access the Irish flag?

```
var flagsJSON =
  '{ "countries" : [' +
  '{ "country":"Ireland" , "flag":"IE" },' +
  '{ "country":"Serbia" , "flag":"RS" },' +
  '{ "country":"Peru" , "flag":"PE" } ]}';

var flagDatabase = JSON.parse(flagsJSON);
```

- [ ] flagDatabase.countries[1].flag
- [x] flagDatabase.countries[0].flag
- [ ] flagDatabase[1].flag
- [ ] flagsJSON.countries[0].flag

Q131. Which snippet allows the acresOfRainForest variable to increase?

```
let conservation = true;
let deforestation = false;
let acresOfRainForest = 100;
if (/* Snipped goes here */){
    ++acresOfRainForest;
}
```

- [x] conservation && !deforestation
- [ ] !deforestation && !conservation
- [ ] !conservation || deforestation
- [ ] deforestation && conservation || deforestation

Q132. Which of these evaluate to true?

- [x] Boolean("false")
- [ ] Boolean("")
- [ ] Boolean(0)
- [ ] Boolean(NaN)

Q133. How would you add a data item named animal with a value of sloth to local storage for the current domain?

- [ ] LocalStorage.setItem("animal","sloth");
- [x] document.localStorage.setItem("animal","sloth");

- [ ] localStorage.setItem({animal:"sloth"});
- [ ] localStorage.setItem("animal","sloth");

Reference

Q134. What value is printed to the console after this code execute?

```
let cat = Object.create({ type: 'lion' });
cat.size = 'large';

let copyCat = { ...cat };
cat.type = 'tiger';

console.log(copyCat.type, copyCat.size);
```

- [ ] tiger large
- [ ] lion undefined
- [x] undefined large
- [ ] lion large

Reference

Q135. What does this code print to the console?

```
let animals = [{ type: 'lion' }, 'tiger'];
let clones = animals.slice();

clones[0].type = 'bear';
clones[1] = 'sheep';

console.log(animals[0].type, clones[0].type);
console.log(animals[1], clones[1]);
```

- [x] bear bear tiger sheep
- [ ] lion bear sheep sheep
- [ ] bear bear tiger tiger
- [ ] lion bear tiger sheep

Reference

Q136. What will be the output of the following code.

```
a=5;
b=4;
alert(a++(+(+(+b))));
```

- [ ] 18
- [ ] 10
- [x] 9
- [ ] 20

Q137. Which snippet could you add to this code to print "{"type": "tiger"}" to the console?

```
let cat = { type: "tiger", size: "large" };

let json = /* Snippet here */;

console.log(json); // print {"type":"tiger"}
```

- [ ] `cat.toJSON("type");`
- [x] `JSON.stringify(cat, ["type"]);`
- [ ] `JSON.stringify(cat);`
- [ ] `JSON.stringify(cat, /type/);`

Reference

Q138. Which document method is not used to get a reference to a DOM node?

- [x] document.getNode();
- [ ] document.getElementsByClassName();
- [ ] document.querySelectorAll();
- [ ] document.querySelector();

Reference

Q139. In JavaScript, all objects inherit a built-in property from a ****__****.

- [ ] node
- [ ] instance variable
- [x] prototype
- [ ] accessor

Reference

Q140. Which of the following are not server-side Javascript objects?

- [ ] Date
- [ ] FileUpload
- [ ] Function
- [x] All of the above

Q141. What will be the output of the following code snippet?

```
const obj1 = { first: 20, second: 30, first: 50 };
console.log(obj1);
```

- [ ] first: 30 , second: 50
- [x] first: 50 , second: 30
- [ ] first: 30 , second: 20
- [ ] None of the above

Q142. Which object in Javascript doesn't have a prototype?

- [x] Base Object
- [ ] All objects have prototype
- [ ] None of the objects have prototype
- [ ] None of the above

Q143. What does ... operator do in JS?

- [x] Used to spread iterables to individual elements
- [ ] Describe datatype of undefined
- [ ] No such operator exists
- [ ] None of the above

Q144. How to stop an interval timer in Javascript?

- [x] clearInterval
- [ ] clearTimer
- [ ] intervalOver
- [ ] None of the above

Reference

Q145. What will be the output of the following code snippet?

```
print(typeof NaN);
```

- [ ] Object
- [x] Number
- [ ] String
- [ ] None of the above

Q146. What will be the output of the following code snippet?

```
<script type="text/javascript">a = 5 + "9"; document.write(a);</script>
```

- [ ] Compilation Error
- [ ] 14
- [ ] Runtime Error
- [x] 59

Q147. Which of the following methods can be used to display data in some form using Javascript?

- [ ] document.write()
- [ ] console.log()
- [ ] window.alert()
- [x] all of the above

Q148. What value is assigned to total after this code executes?

```
function sum(num1, num2 = 2, num3 = 3) {
  return num1 + num2 + num3;
}
let values = [1, 5];
let total = sum(4, ...values);
```

- [x] 10
- [ ] 6
- [ ] 7

- [ ] 8

Reference: Rest parameters

Q149. Which statement is applicable to the defer attribute of the HTML <script> tag?

- [x] defer allows the browser to continue processing the page while the script loads in the background.
- [ ] defer causes the script to be loaded from the backup content delivery network (CDN).
- [ ] defer blocks the browser from processing HTML below the tag until the script is completely loaded.
- [ ] defer lazy loads the script, causing it to download only when it is called by another script on the page.

Reference: defer html script attribute

Q150. Which method of a class is called to initialize an object of that class?

- [ ] init()
- [ ] create()
- [ ] new()
- [x] constructor()

Reference: constructor method

Q151. Which expression evaluates to true?

- [ ] Boolean(NaN)
- [ ] Boolean(0)
- [x] Boolean("false")
- [ ] Boolean("")

Reference: Boolean object

Q152. How would you check if the word "pot" is in the word "potato"?

- [ ] "pot".indexOf("potato") !== -1
- [ ] "potato".includes("Pot")
- [x] "potato".includes("pot")
- [ ] "potato".contains("pot");

Reference: String.prototype.includes()

Q153. Which collection object allows a unique value to be inserted only once?

- [ ] Map
- [ ] Array
- [x] Set
- [ ] Object

Reference: developer.mozilla Set

Q154. How would you change the color of this header to pink?

```
<h2 id="cleverest">girls</h2>
```

- [ ] document.getElementByName("cleverest").style.color = "pink";
- [ ] document.getElementsByTagName("h2").style.color = "pink";
- [ ] document.getElementByName("h2").style.color = "pink";
- [x] document.getElementById("cleverest").style.color = "pink";

Reference: W3Schools HTML DOM Style color Property

Q155. Which line is missing from this code if you expect the code to evaluate to true?

```
var compare = function (test1, test2) {
  // Missing line
};

compare(1078, '1078'); // yields true
```

- [ ] test1==test2;
- [ ] return test1===test2;
- [x] return test1==test2;
- [ ] return test1!=test2;

Reference: MDN Equality Docs