

- the body of the event handler
- the object containing the instruction
- the HTML element originating the event

Questions on Javascript

12. Consider the following definition of an object *Account* with three private variables, two private methods and two public methods:

```
var Account = (function( ) {  
    var Balance = 0; // private  
    var AcctNo = 0; // private  
    var Owner = ""; // private  
    var getBalance = function( ) { // private method  
        return Balance;  
    }  
    var setBalance = function(newBalance) { // private method  
        Balance = newBalance;  
    }  
    var Open = function(balance, acctno, owner) { // public method  
        setBalance(balance);  
        AcctNo = acctno;  
        Owner = owner;  
    }  
    var Deposit = function(amount) { // public method  
        setBalance(getBalance( ) + amount);  
    }  
    return { ... }; // Your return object here  
})( );
```

Your job is complete the code of this *Account* definition by supplying the return object. Write the answer sheet.

```
return {  
    setBalance : setBalance,  
    open : open,  
    Deposit : Deposit  
};
```


18. The purpose of the following HTML web page and Javascript code is to display list items, each with a Delete button. When the User clicks on a *Delete* Button, the corresponding list item (with its Button) disappears from the list. Your job is to write the code of the Javascript *DeleteItem()* function. Write your answer on the answer sheet.

HTML:

```
<h1>Shopping Cart</h1>
<ul>
  <li> Shirt  <button>Delete</button> </li>
  <li> Tie    <button>Delete</button> </li>
  <li> Jacket <button>Delete</button> </li>
  <li> Hat    <button>Delete</button> </li>
</ul>
```

Javascript:

```
(function( ) {
  function DeleteItem( ) {
    ... Insert your code here
  }
  $(document).ready( function( ) {
    $("button").click(DeleteItem);
  });
})();
```


4. Which jQuery function is used to prevent code from running, before it

- (a) `$(body).onload()`
- ☒ (b) `$(document).ready()`
- (c) `$(document).load()`
- (d) `window.onload()`
- (e) all of the above

5. Look at the following selector: `$("div p")`. What does it select?

- (a) All p elements inside a div element
- (b) The first p element inside a div element
- ☒ (c) All div elements with a p element
- (d) Not a valid selector
- (e) All div elements

6. Which jQuery method is used to remove selected elements?

- (a) `remove()`
- (b) `detach()`
- ☒ (c) Both methods can be used

7. Which jQuery method returns the direct parent element of the selected

- ☒ (a) `parent()`
- (b) `parents()`
- (c) `ancestor()`
- (d) `ancestors()`
- (e) `above()`

8. Which of the following jQuery method sets the html contents of an e

- ☒ (a) `html(val)`
- (b) `setHtml(val)`
- (c) `setInnerHtml(val)`
- (d) `innerHTML(val)`
- (e) None of the above

18. The purpose of the following HTML web page and Javascript code is to display list items, each with a Delete button. When the User clicks on a *Delete* Button, the corresponding list item (with its Button) disappears from the list. Your job is to write the code of the Javascript *DeleteItem*() function. Write your answer on the answer sheet.

HTML:

```
<h1>Shopping Cart</h1>
```

```
<ul>
```

```
  <li> Shirt  <button>Delete</button>  </li>
```

```
  <li> Tie    <button>Delete</button>  </li>
```

```
  <li> Jacket <button>Delete</button>  </li>
```

```
  <li> Hat    <button>Delete</button>  </li>
```

```
</ul>
```

Javascript:

```
(function( ) {
```

```
  function DeleteItem( ) {
```

```
    ... Insert your code here
```

```
  }
```

```
  $(document).ready( function( ) {
```

```
    $("button").click(DeleteItem);
```

```
  });
```

```
})();
```

16. Consider the following *Object Factory Function* called *Employee*:

```
Employee = function( ) {  
    theEmp = {  
        name: "", // public field  
        salary: 0 // public field  
    };  
    getSalary = function( ) { // private method  
        return theEmp.salary;  
    }  
    setSalary = function(aSalary) { // private method  
        theEmp.salary = aSalary;  
    }  
    giveRaise = function(amount) { // public method  
        setSalary(getSalary( ) + amount);  
    }  
    ... // insert a statement here to make giveRaise public  
    return theEmp;  
}
```

A new *Employee* object with *Salary* 15000 is created as follows:

```
aEmp = Employee( );  
aEmp.name = "George Smith";  
aEmp.salary = 15000;  
aEmp.giveRaise(1000); // Give raise of 1000
```

To make this *Object Factory* work correctly, one additional Javascript statement is needed right before make the *giveRaise* method public, as indicated by the comment shown above. Your job is to write the statement. Write your answer on the answer sheet.

```
17. function Data() {  
    var x = [];  
    return {  
        put: function (item) {  
            x.push(item);  
        },  
        take: function() {  
            return x.pop();  
        },  
        size: function() {  
            return x.length;  
        }  
    };  
}
```

```
y = Data(); y.put(10); y.put(-15);  
alert(y.size());  
alert(y.take());  
alert(y.size());
```

What is the result of executing the above Javascript code? Write your answer on the answer sheet.

12. HTML Form data always goes directly to the Server, it cannot be accessed by Javascript running in the Browser.

13. An XMLHttpRequest to a Web Server must always return data in XML format.

14. In a jQuery \$.ajax() call, webpage form data can be sent to the server as a GET request without appending a query string to the ajax URL parameter.

15. When using jQuery \$.ajax(), both Get and Post requests are allowed.

16. By using jQuery \$.ajax(), the Javascript code running in the Web Browser can access SQL Databases on the client computer.

17. Most Web Browsers have jQuery built into the Browser.

18. When the Javascript "new" operator creates an object, the constructor function defines the prototype of the object.

19. A Javascript object created with *Object.create*() can be extended with new fields without changing its prototype object.

20. Javascript variables defined with *let* have block scope.

13. What is the result of executing the following Javascript code:

```
var a = 10;
function f( ) {
  var b = 20;
  function g( ) {
    var c = a + b;
    return c;
  }
  b = 30; return g;
}
myFunc = f( );
x = myFunc( );
alert(x); 40
a = 100;
y = myFunc( );
alert(y); 130
```

14. What is the result of executing the following Javascript code:

```
var b = 1000;
function g( ) {
  var a = 100;
  b = 10;
  var z = a + b;
  alert(z); 110
  function f(z) {
    return 2*z;
  }
}
```


Javascript

```
{function( ) {  
    function DeleteItem( ) {  
        ... Insert your code here  
    }  
    $(document).ready( function( ) {  
        $("button").click(DeleteItem);  
    })  
})();
```

True/False Questions

Instructions: Indicate *True* or *False* for each statement.

1. In Javascript, an *event handler* is always a function.
2. The DOM document object contains the window object.
3. All global variables in any Javascript file are visible to all other Javascript files of the same domain.
4. Each Javascript file has its own global namespace, separate from other files.
5. All declarations of free variables are *hoisted* into the global object.
6. Using the DOM, Javascript code can determine which Browser is being used.
7. *Unobtrusive* Javascript means all Javascript code for a website is put in the same file.
8. Javascript code is executed when the Browser reads the *script* tag.
9. An *anonymous* function can never be called because it has no name.
10. The Javascript instruction `x = $("#p")` stores a jQuery object in variable `x`.
11. If `y` is a DOM object, then `y.tagName` contains the HTML tag name.

True/False Questions

1. **True** All global variables in any javascript file are visible to all other javascript files on the same webpage.
2. **False** Each Javascript file has its own global namespace, separate from other files.
3. **False** All declarations of free variables are hoisted into the global object.
4. **True** Using the DOM, Javascript code can determine which Browser is being used.
5. **False** Unobtrusive Javascript means all Javascript code for a website is put in the same file.
6. **True** JavaScript code is executed when the Browser reads the script tag.
7. **False** An anonymous function can never be called because it has no name.
8. **True** The Javascript instruction `x = $("p")` stores a jQuery object in variable x.
9. **True** If y is a DOM object, then `y.tagName` contains the HTML tag name.
10. **False** HTML Form data always goes directly to the Server, it cannot be accessed by Javascript running on the browser.
11. **False** An XMLHttpRequest to a Web Server must always return data in XML format.
12. **True?/False!** When using a jQuery `$.ajax()` call, webpage form data in XML format.
13. **True** When using jQuery `$.ajax()`, both Get and Post requests are allowed.
14. **False** By using jQuery `$.ajax()`, the Javascript code running in the Web Browser can access SQL Databases stored on the client computer.
15. **False** Most Web Browsers have jQuery built into the Browser.
16. **True** When the Javascript "new" operator creates an object, the constructor function defines the prototype of the object.
17. **True** A Javascript object created with `Object.create()` can be extended with new fields without changing its prototype object.
18. **True** In Javascript, an event handler is always a function.
19. **False** The DOM document object contains the window object.
20. **True** Javascript variables defined with `let` have block scope.
21. **True** In a jQuery `$.ajax()` call, webpage format data can be sent to the server as a GET request without app query string to the Ajax URL parameter?
22. **True** When using a jQuery `$.ajax()` call, webpage form data can be sent to the server.