**Ext JS quiz**

In all questions below, if not specially noted, the answer is supposed to use the ExtJS library, however if mentioned, pure JavaScript should be used. You can skip the question #14, though solving it will be a big plus.

1. Given 2 arrays “a” and “b”, how would you:
   1. append all elements of “b” to “a”
   2. prepend all elements of ”b” to ”a”
   3. put all elements of ”b” in the middle of ”a” (assuming ”a” contains even number of elements)

Your code should modify *existing* array “a”, not create a new one. Try to find the shortest code in pure JavaScript to solve this task.

**Answer:**

1. Array.prototype.push.apply(a,b)
2. Array.prototype.unshift.apply(a,b)
3. Array.prototype.splice.apply(a,[a.length/2,0].concat(b))
4. Given an arbitrary array which potentially contains other *flat* arrays:

[ [1, 2, “a”], 3, “foo”, [ “bar”, {} ] ]

How do you transform this into a completely flat array?

[ 1, 2, “a”, 3, “foo”, “bar”, {} ]

Try to find the shortest code in pure JavaScript to solve this task.

**Answer:**

Array.prototype.concat.apply([],a)

1. Write a single RegExp to extract the Ext JS version identifier (e.g. “3.4.0”, or “4.1.1-rc-1”) out of the URLs below. Note, that URLs potentially can point to some other resource with similar URL (for example ext-all-this-is-not-ext.js), in which case the RegExp should not match.

http://bryntum.com/examples-1.2.3/advanced/advanced.html  
http://bryntum.com/doc/1.2.3/foo/bar.html  
http://cdn.sencha.io/ext-3.4.0/ext-all-debug.js  
http://cdn.sencha.io/ext-3.4.0-beta-1/ext-all.js  
http://cdn.sencha.io/extjs-4.1.0-rc-1/ext-all-debug.js  
http://bryntum.com/examplesfor/extjs-4.1.1/foo/bar.html  
http://bryntum.com/library/extjs-4.1.2/ext-all-debug.js  
http://bryntum.com/library/extjs-10.11.21/ext-all-debug.js  
http://bryntum.com/library/extjs-4.1.2/ext-all-this-is-not-ext.js

**Answer:**

/\/ext(?:js)?-(\d+\.\d+\.\d+[\w-]\*)\/ext-all(?:-debug)?\.js$/

1. Write a RegExp to extract the x, y, z offset values from these CSS style declarations:

translate3d(0, 0, 0)  
translate3d(0, 0, 0)  
translate3d(1px, 10px, 100px)  
translate3d(-1px, -10px, -100px)

The x, y, z offsets should be numbered as 1, 2, 3 in the RegExp match object. Strings of other formats should not match.

**Answer:**

/^translate3d\(\s\*(-?\d+)(?:px)?\s\*,\s\*(-?\d+)(?:px)?\s\*,\s\*(-?\d+)(?:px)?\s\*\)$/

1. You are given a following JSON tree structure:

var tree = {

id : '1',

children : [

{

id : '2',

children : [

{

id : '5',

children : [

{

id : '9'

}

]

},

{

id : '6'

}

]

},

{

id : '3',

children : [

{

id : '7'

},

{

id : '8'

}

]

},

{

id : '4'

}

]

}

Write the iterator function using pure JavaScript, which will call the provided “func” for each node in the tree, so the following console output will be correct:

var iterator = function (root, func) {

...

}

iterator(tree, function (node) {

console.log(node.id)

})

>>Output:

1

2

3

4

5

6

7

8

9

**Answer:**

var iterator = function(node, fun) {

var q = [node];

while((node = q.shift())){

fun(node);

Array.prototype.push.apply(q, node.children)

}

};

1. You need to add a visual effect when the user is hovering the mouse over the <li> items in the list below, how do you solve it? The <ul> list contains 10 items.

<ul class=”somelist”>

<li>one</li>

<li>two</li>

<li>three</li>

...

</ul>

**Answer:**

Ext.select('.somelist').on({ mouseover:over, mouseout:out, delegate:'li'});

1. What is wrong with the following Ext JS 4 class definition:

Ext.define('MyPanel', {

extend : 'Ext.Panel',

sayHello : function() { console.log('Hello world'); },

tbar : [

{

text : 'Click to say hello',

handler : this.sayHello,

scope : this

}

],

... // stores, columns defined below

});

**Answer:**

this points to window

**Corrected class definition:**

Ext.define('MyPanel', {

extend : 'Ext.panel.Panel',

sayHello : function() { console.log('Hello world'); },

initComponent:function(){

Ext.apply(this,{

tbar : [

{

text : 'Click to say hello',

handler : this.sayHello,

scope : this

}

]

});

this.callParent();

}

});

1. The *hypothetical* grid class below has special attribute “rowColors” and can update it dynamically using “setRowColor”. What is problematic with this implementation?

Ext.define('MyGrid', {

extend : 'Ext.grid.Panel',

// Row coloring for normal and alternate rows

rowColors : {

normal : ‘blue’,

alternate : ‘white’

},

// Updates the grid row colors

setRowColors : function(normal, alternate) {

this.rowColors.normal = normal;

this.rowColors.alternate = alternate;

this.getView().refresh();

},

...

});

**Answer:**

**Corrected class definition:**

1. What is wrong with the following Ext JS 4 class definition:

Ext.define('ReusableButton', {

extend : 'Ext.Button',

initComponent : function() {

this.id = ‘reusable-button’; // So we can use Ext.getCmp

this.callParent();

}

});

Ext.define('ReusablePanel', {

extend : 'Ext.Panel',

initComponent : function() {

this.buttons = [

new ReusableButton({

text : ‘Click me’

})

];

this.callParent();

},

...

})

**Answer:**

Explicit Component creating does not allow using asynchronous loading. Also it would be better if you set component id in config to use Ext.getCmp(Although in this example using Ext.getCmp is not recommended – use componentQuery by itemId or action).

**Corrected class definition:**

Ext.define('ReusableButton', {

extend : 'Ext.Button',

xtype:'reusablebutton',

initComponent : function() {

this.callParent();

}

});

Ext.define('ReusablePanel', {

extend : 'Ext.Panel',

initComponent : function() {

this.buttons = [

{

xtype:'reusablebutton',

id : 'reusable-button', // So we can use Ext.getCmp

text : 'Click me'

}

];

this.callParent();

}

});

1. What is wrong with the following Ext JS snippet:

var myEl = Ext.fly(‘myDiv’);

myEl.highlight(); // Show user where the element is

myEl.setHeight(200);

// Some other code

// …

var *panel = new Ext.Panel();*

*panel.render(document.body);*

*myEl.setWidth(200);*

**Answer:**

Ext.fly holds the last accessed element(flyweight). So myEl in the last line points to the new created panel.

It is not recommend to save Ext.fly result to use it later.

1. One of our clients reports an exception using our Scheduler component (which supports both TreeStore and the normal flat Store) in his application. He points to this line in our source code, what could be the issue? Assume that “this” reference contains a correct reference to a Scheduler with the flat store.

var isTree = this.store instanceof Ext.data.TreeStore;

**Answer and improved solution:**

1. List all the potential issues with the following Ext JS 4 plugin definition:

Ext.define('MyGridPlugin', {

init : function(grid) {

grid.getStore().on({

load : function() {

grid.el.highlight(); // Some cool fx to bring attention to the grid

}

});

}

});

// Let’s try it out

var myGrid = new Ext.grid.Panel({

renderTo : document.body,

store : someStore,

plugins : new MyGridPlugin()

});

someStore.load();

**Answer:**

Plugins should extend Ext.AbstractPlugin and

MyGridPlugin may not be accessible because it is not loaded. (Explicit new)

**Corrected plugin definition:**

Ext.define('MyGridPlugin', {

extend:'Ext.AbstractPlugin',

alias:'plugin.MyGridPlugin',

init : function(grid) {

grid.getStore().on({

load : function() {

// grid.el.highlight(); // Some cool fx to bring attention to the grid

}

});

}

});

myGrid = new Ext.grid.Panel({

renderTo : Ext.getBody(),

store: someStore,

plugins : {ptype:'MyGridPlugin'}

});

1. Designing complex and re-usable ExtJS UI widgets (potentially consisting from other re-usable widgets), what will be your design principles / best practices / etc?

**Answer:**

1. File structure repeats namespaces(Widget is in file with the same name)

2. Use aliases(xtype) for widgets(widget, plugin, store, data, features and etc)

3. Do not use Ext.getCmp or similar methods to access Component by id(Use itemId and refs to access children)

4. Do not overnest Components

5. Move general functionality to the Base classes

6. Use mixins for cross-functionality

7. Use singleton in OneInstanceWidgets. Also singleton can be used for create objects to hold constants and some util functions

8. Use event listeners from children items instead of override chlidren methods in config.

9. Sometimes overriding component properties in its xtype-config is better than create new subclass.

10.Avoid using ownerCt

1. You are given an input array of the intervals on the number axis. For each interval a start point, end point and some “value” is known. Write a function in pure JavaScript, that will transform the input array to another array of the objects with the same structure. Function should sum the “values” of all intersecting intervals and include in the result all intervals with *different* sum of “values”. See the examples of input and expected output. Note, that start and end point coordinates may have fractional part.

**var** input = [{

start : 1,

end : 2,

value : 1

}, {

start : 2,

end : 3,

value : 1

}]

**var** output = [{

start : 1,

end : 3,

value : 1

}]

**var** input = [{

start : 1,

end : 3,

value : 1

}, {

start : 2,

end : 4,

value : 1

}]

**var** output = [{

start : 1,

end : 2,

value : 1

}, {

start : 2,

end : 3,

value : 2

}, {

start : 3,

end : 4,

value : 1

}]

**var** input = [{

start : 1,

end : 3,

value : 1

}, {

start : 2,

end : 4,

value : 1

}, {

start : 2,

end : 5,

value : 2

}, {

start : 4,

end : 5,

value : 1

}]

**var** output = [{

start : 1,

end : 2,

value : 1

}, {

start : 2,

end : 3,

value : 4

}, {

start : 3,

end : 5,

value : 3

}]

**Answer:**

function integrate(/\*Array\*/ input){

var points = [];

// sort all points and distinct

for(var i = 0; i < input.length; i++) {

points.push(input[i].start);

points.push(input[i].end);

}

points.sort();

for(var i = 1, a = points[0], unique = [a]; i < points.length; i++) {

if(a != points[i]){

unique.push(points[i]);

a = points[i];

}

}

// interpolate

var out = [], prevSum = 0;

for(var i = 0; i < unique.length -1; i++) {

var start = unique[i], end = unique[i+1] , sum = 0;

for(var j = 0; j < input.length; j++)

if(input[j].start <= start && input[j].end >=end)

sum += input[j].value;

if(sum == prevSum){

out[out.length - 1].end = end;

continue;

}

out.push({start:start, end:end, value:sum});

prevSum = sum;

}

return out;

}