JAVASCRIPT

1)

//object literal

var Person = {

firstName: "Srihitha",

lastName: "T",

age: 20

}

console.log(Person.firstName)

//object constructor

var p1 = new Object()

p1.firstname = "srihitha"

console.log(p1)

2)

function getkeyFields(input, field) {

var output = [];

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

output.push(input[i][field]);

return output;

}

objArray = [{channel: 'A'}, {channel: 'B'}, {channel: 'C'}];

console.log(getkeyFields(objArray, "channel"));

3) 4)

var channels = [

{

"channel": "A",

"name": "shoe"

},

{

"channel": "A",

"name": "electronics"

},

{

"channel": "B",

"name": "apparel"

},

{

"channel": "C",

"name": "electronics"

}

]

result = channels.reduce(function (r, a) {

r[a.channel] = r[a.channel] || [];

r[a.channel].push(a);

return r;

}, Object.create(null));

console.log(result);

5)

class Sort{

constructor(originalarray){

this.originalarray = originalarray;

}

sortfun(originalarray){

const oarray = this.originalarray

const a = oarray.sort();

this.getarr(a);

}

getarr(sortedarray){

console.log(sortedarray)

}

}

class Sortobjarray extends Sort{

constructor(originalarray){

super(originalarray)

}

sortarr(array){

return this.sortfun(array)

}

}

const arr = new Sortobjarray([1,4,2,3,5])

arr.sortarr()

NODEJS

1)

.js

var fs = require('fs');

var parse = require('xml-parser');

//var file1 = new File('examples/try.xml')

var fs = require('fs')

fs.exists('examples/try.xml', (exists)=>{

if(exists){

var xml =fs.readFileSync('examples/try.xml', 'utf8')

var obj = parse(xml);

console.log(inspect(obj, { colors: true, depth: Infinity }))}

else {

console.log("file doesn't exists");

}

}

)

var inspect = require('util').inspect;

.xml

<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns="urn:enterprise.soap.sforce.com">

<soapenv:Body>

<createResponse>

<result>

<id>1</id>

<success>true</success>

</result>

<result>

<id>2</id>

<success>true</success>

</result>

</createResponse>

</soapenv:Body>

</soapenv:Envelope>

3)

const http = require('http');

const fs = require('fs');

const { EventEmitter } = require('events');

const eventEmitter = new EventEmitter()

const file = fs.createWriteStream("file.html");

eventEmitter.on('start', ()=>{

const request = http.get("http://www.google.com", function(response) {

response.pipe(file);

});})

eventEmitter.emit('start')

4)

const fs = require('fs');

const request = require('request');

var download = function(uri, filename, callback){

request.head(uri, function(err, res, body){

request(uri).pipe(fs.createWriteStream(filename)).on('close', callback);

console.log(filename)

});

};

download('https://www.google.com/images/srpr/logo3w.png', './images/google.png', function(){

console.log('done');

});

5)

var shortUrl = require("node-url-shortener");

const fs = require('fs')

urls = ['https://doodleart.redbull.com/assets/managed/entries/processed/sm/367010617181759\_36211000.jpg', 'https://i.pinimg.com/originals/e5/55/a3/e555a39ca5457a079a9bcce59f61f8d5.jpg']

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

shortUrl.short(urls[i], function (err, url)

{

fs.appendFileSync('dum.csv', `${url}\n`)

console.log(url);

})

}