

The first part of the challenge is the web challenge, since we need to deobfuscate the js code. I used <https://lelinhtinh.github.io/de4js/>. First, I chose Packer option (to unpack it), then I copied the unpacked js into the encoded box and clicker auto-decode. The result is as follows:

```
var FLAG = ['YWg', '/gQ', '0b3', 'wHH', '-%$', '1cf', 'wEH', '%M  
R', '4fb', 'wLM', '%\"R', '29b', 'HFK', 'MQ ', '4e1', 'JHM', 'N  
%', 'ac4', 'HMI', '%%', '87b', 'IFv', '$!!', '}'];  
var MAXN = 0x32;  
  
function open_windows(_12, _4) {  
    _4--;  
    popupWindow = window.open('https://' + reverse_string(_12[_4  
]), 'popUpWindow', 'height=137,width=137,left=137,top=137');  
    setTimeout(() => {  
        open_windows(_12, _4)  
    }, 0x3e8)  
}  
  
function reverse_string(_14) {  
    var _15 = _14.split('');  
    var _16 = _15.reverse();  
    var _13 = _16.join('');  
    return _13  
}  
  
function chunkString(_17, _18) {  
    return _17.match(new RegExp('.{1,' + _18 + '}', 'g'))  
}  
  
function enc1(_10) {  
    nchunk = [];  
    for (var _6 = 0x0; _6 < _10.length; _6++) {  
        nchunk.push(String.fromCharCode(_10[_6].charCodeAt() + 0  
x14))  
    }  
    return nchunk.join('')  
}  
  
function enc2(_9) {  
    nchunk = [];  
    for (var _5 = 0x0; _5 < _9.length; _5++) {  
        nchunk.push(String.fromCharCode(_9[_5].charCodeAt() - 0x  
14))  
    }  
    return nchunk.join('')  
}
```

```

function enc3(_19) {
    nchunk = reverse_string(_19);
    return nchunk
}

function encode(_3) {
    functs = [enc1, enc2, enc3];
    for (var _2 = 0x0; _2 < _3.length; _2++) {
        _3[_2] = functs[_2 % 0x3](_3[_2])
    }
    return _3
}

links = ['moc.elgooG', 'moc.ebutuoY', 'moc.llamT', 'moc.koobecaF',
', 'moc.udiaB', 'moc.qQ', 'moc.uhoS', 'moc.llamt.nigoL', 'moc.oa',
'boat', 'nc.063', 'moc.dJ', 'gro.aidepikiW', 'moc.oohaY', 'moc.no',
'zamA', 'nc.moc.anis', 'moc.obieW', 'moc.llamt.segaP', 'su.mooZ',
'moc.eviL', 'moc.xilfteN', 'moc.tiddeR', 'moc.tenauhniX', 'moc.k',
'V', 'moc.tfosorciM', 'moc.enozekO', 'moc.eciffO', 'moc.topsgolB',
', 'ten.ndsC', 'moc.yapilA', 'moc.margatsnI', 'pj.oc.oohaY', 'vt.',
'hctiwT', 'moc.smacagnoB', 'kh.moc.elgooG', 'moc.enilnotfosorciM',
', 'moc.nimsajeviL', 'moc.gniB', 'moc.swennubirT', 'moc.revaN', 'vt.',
'adnaP', 'vt.iqnahZ', 'pj.oc.nozamA', 'moc.wolfrevokcatS', 'nc',
'.aynaiT', 'moc.sserpxeilA', 'ni.oc.elgooG', 'moc.rettiwt', 'moc.',
'yabE', 'moc.yfipohsyM', 'ofni.sretemodlrOW'];
open_windows(links, Math.floor(Math.random() * MAXN + MAXN / 0x2));
console.log(encode(FLAGS));

```

Now, we need to reverse the encoding of the flag:

```

def dec1(s):
    return ''.join(chr(ord(c) - 20) for c in s)
def dec2(s):
    return ''.join(chr(ord(c) + 20) for c in s)
def dec3(s):
    return s[::-1]

def decode(encoded_flag):
    functs = [dec1, dec2, dec3]
    decoded = encoded_flag[:]
    for i in range(len(decoded)):
        decoded[i] = functs[i % 3](decoded[i])
    return decoded

FLAG = ['YWg', '/gQ', '0b3', 'wHH', '-%$', '1cf', 'WEH', '%MR', '4fb', 'wLM',
'%\"R',

```

```
        '29b', 'HFK', 'MQ ', '4e1', 'JHM', 'N%', 'ac4', 'HMI', '%%', '87b',  
'IFv',  
        '$!!', '}]']  
  
decoded_flag = decode(FLAG)  
concatenated_flag = ''.join(decoded_flag)  
print("Decoded FLAG:", concatenated_flag)
```

We basically only need to reverse enc1 enc2 and enc3 to apply the opposite operation on each character of FLAG.

Made with love by: AndreiCat