

UnderTheCTF

this is the part where we  
write the flag on the board

# cyphercon

- milwaukee
- april
- sign up:

[https://docs.google.com/forms/d/1\\_0UMjMwvVCbgZRKNAkUAPbh5IfCcQy\\_5GmNotwZyt3g/edit?usp=drive\\_web](https://docs.google.com/forms/d/1_0UMjMwvVCbgZRKNAkUAPbh5IfCcQy_5GmNotwZyt3g/edit?usp=drive_web)

## quick aside: textsheet.com

- free chegg (probably scrapes it using stolen / bought accounts
  - that or chegg doesn't actually check for abuse
- they display annoying surveys bc the person who made it wants money
- then they un-hide the answer
- fun snippet of the source code:

```
/ We look at whether FuckAdBlock already exists.  
if(typeof fuckAdBlock !== 'undefined' || typeof FuckAdBlock !== 'undefined')  
    // If this is the case, it means that something is wrong  
    // So, considering that it is a detection  
    adBlockDetected();  
else {  
    // Otherwise, you import the script FuckAdBlock
```

## quick aside: textsheet.com

- anyway it's trivial to bypass: the content you want has an id="content"
- so

### Edit bookmark

Name

fuck textsheet

URL

javascript:document.write(document.getElementById('content').innerHTM

# quick aside: textsheet.com

```
// ==UserScript==  
// @name    Scrape textsheet  
// @match   *textsheet.com/*  
// @match   *www.textsheet.com/*  
// @require http://ajax.googleapis.com/ajax/libs/jquery/2.1.0/jquery.min.js  
// @require https://gist.github.com/raw/2625891/waitForKeyElements.js  
// @grant   GM_addStyle  
// ==/UserScript==  
//- The @grant directive is needed to restore the proper sandbox.
```

```
waitForKeyElements("#content", showContent);  
function showContent () {  
    document.write(document.getElementById('content').innerHTML);  
}
```

i have one gripe: i couldn't figure out the search box & captcha; i'll give u **pwnypoints** if you do  
document.getElementById('search-box').innerHTML +

<https://underthectf.com/>

# Music

- Musical notes!
- Find how musical notation works for the flag
- Crypto can be virtually anything
  - Usefulness is a different measure

## The Hex





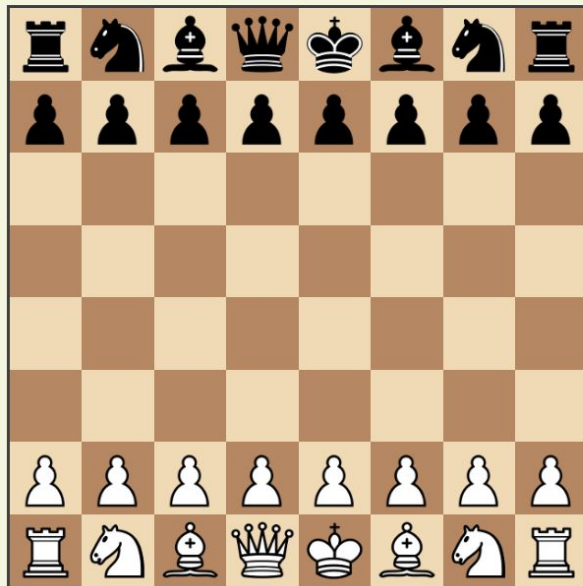
# Algebraic

- Understand bad chess notation to solve the challenge
- Rows are 1-8 from bottom to top, columns are a-h from left to right
- Can't really cheat the flag as the javascript sends a GET request with the current "position" of the board
  - However, can play moves in any order: all it wants is a GET request with the following query string parameter:
  - fen: 1b1k2r/ppppqppp/2n5/8/1PP2B2/3n1N2/1P1NPPPP/R2QKB1R
  - which represents the state of the board at the end

The following string encodes a chess game:

*d2d4g8f6c2c4e7e5d4e5  
f6g4c1f4b8c6g1f3f8b4b1d2  
d8e7a2a3g4e5a3b4e5d3*

Once you play out all of the moves, a flag will be revealed.

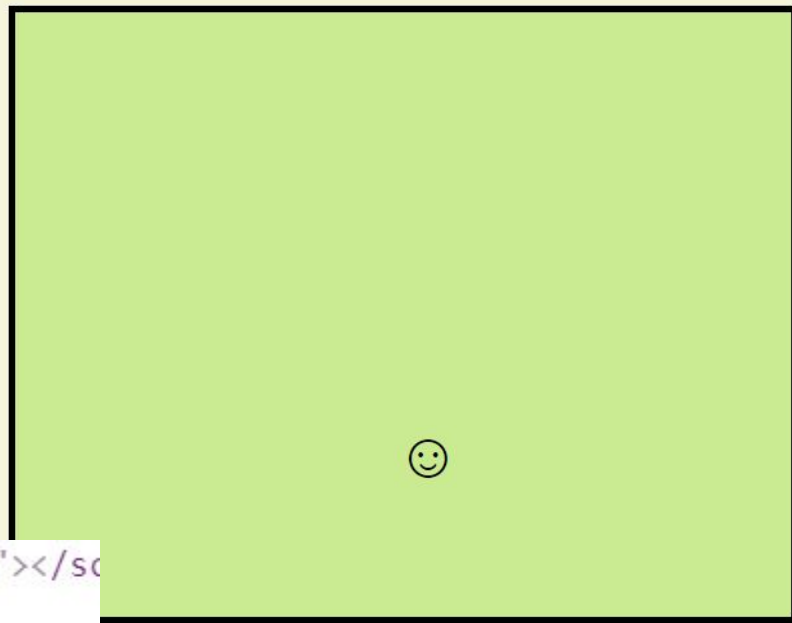


# Clicks

- Flag looks like `flag{...}`
- Inspect Element
  - HTML element
  - CSS Top and Left permuted
  - Probably JS
- What's moving it?
  - Event Listeners
  - Or notice, literally `flag.js`

```
<script src="jquery-1.12.1.min.js"></script>  
<script src="flag.js"></script>  
</body>
```

If you click the face enough, the flag will surely appear.



̄vOyḠBw⌘OyΨvцwV̂⌘w⌘ψΨvĈIE

# Clicks

- Flag looks like `flag{...}`
- XOR with integer key
- `charCodeAt` / `fromCharCode`
  - Key between 0 and 0xFFFF
  - Brute force
- Remember flag format
- Borrow their function

```
1 (function () {
2   var key = 0;
3   var text = '\u047f\u0475\u0478\u047e\u0462\u0461\u04
4     '\u0470\u0477\u0446\u0461\u0476\u046b\u0
5     '\u0470\u0477\u047c\u0464';
6
7   var decrypt = function (key) {
8     var i, newChar, decrypted = '';
9     for (i = 0; i < text.length; i++) {
10       newChar = text.charCodeAt(i) ^ key;
11       decrypted += String.fromCharCode(newChar);
12     }
13     return decrypted;
14   };
15
16   var showDecrypted = function () {
17     var result = decrypt(key++);
18     S('#output') text(result);
```

```
> for (let i=0; i<65536; i++) if (temp1(i).startsWith("flag")) console.log(temp1(i));
```

```
flag{[REDACTED]}
```

```
< undefined
```

```
> |
```