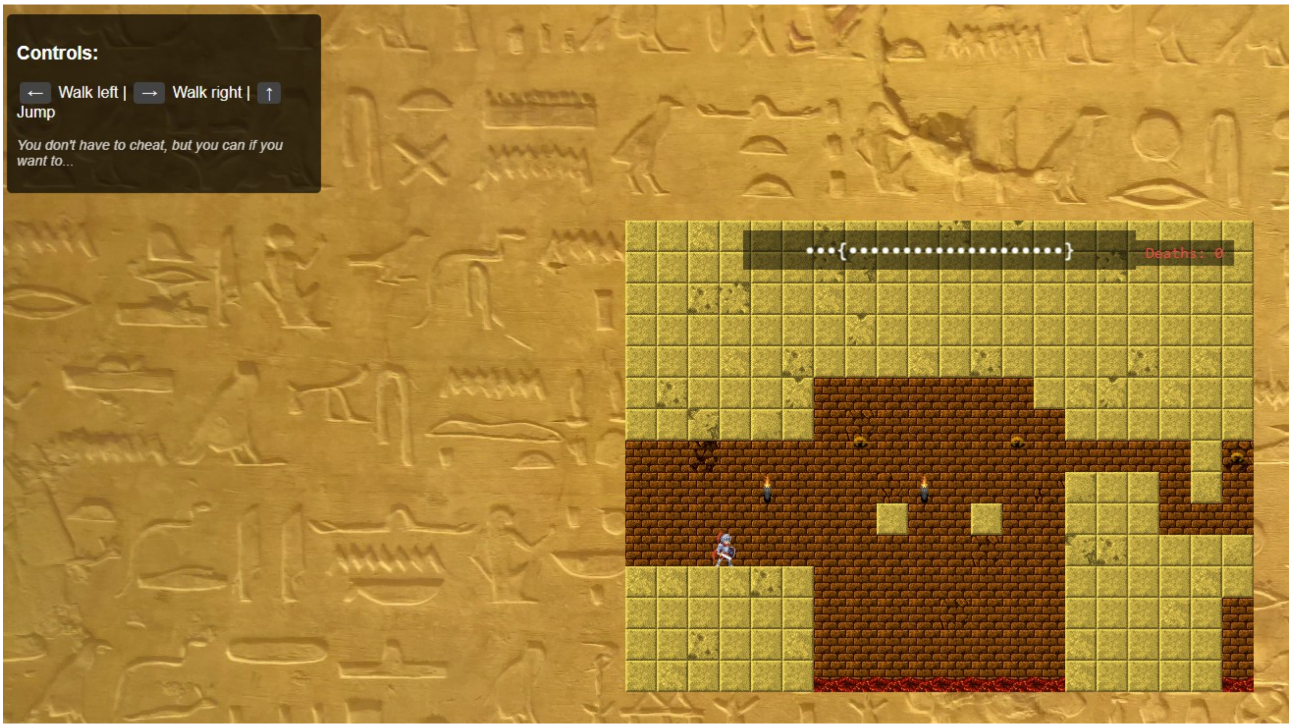
Tomb Raider

DarkTemplar

Kleopatras_Klantarslen



Controls:

← Walk left | → Walk right | ↑ Jump

You don't have to cheat, but you can if you want to...

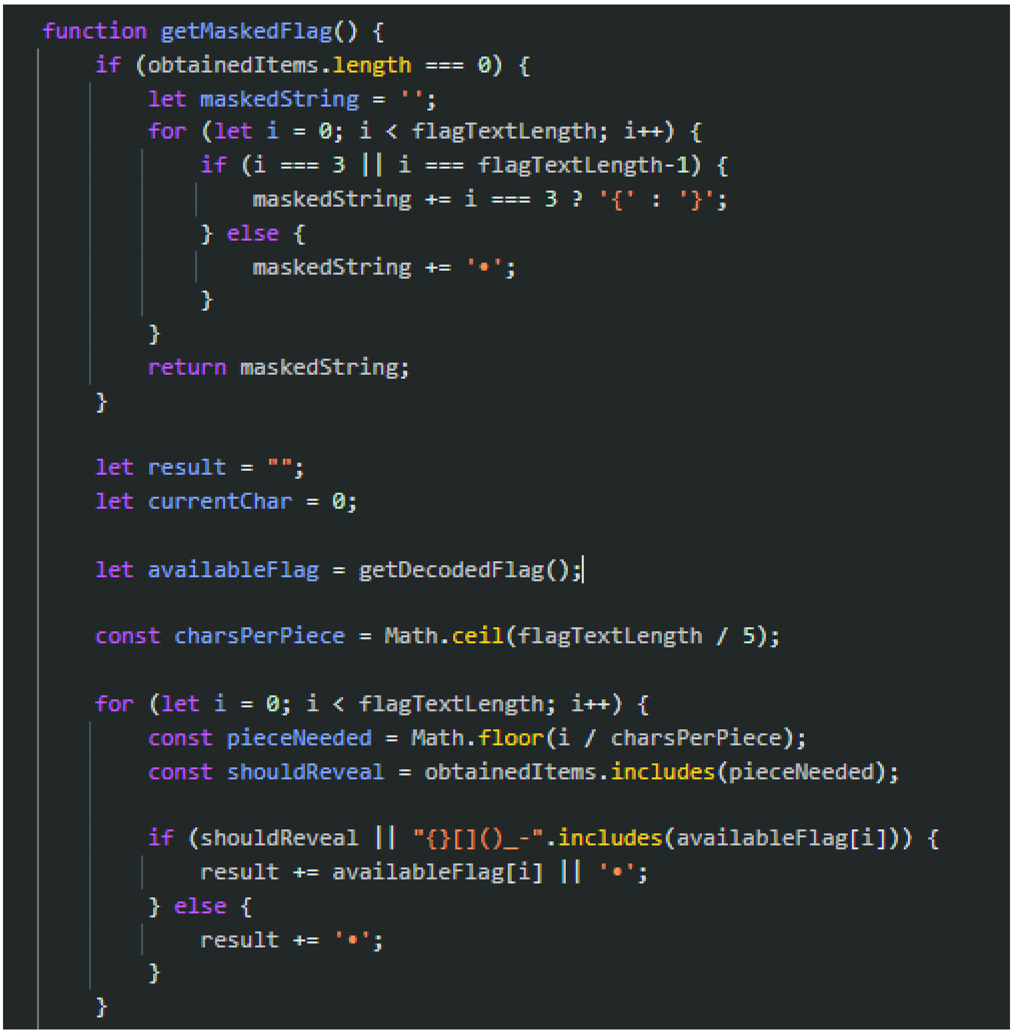
Deaths: 0

Upon entering the site we get this screen. Seems to be a game of sorts. I like the little line: ”You don’t hav eot cheat, but you can if you want to…” We’ll try without cheating. I’m a gamer, how hard can it be?

Deaths: 1

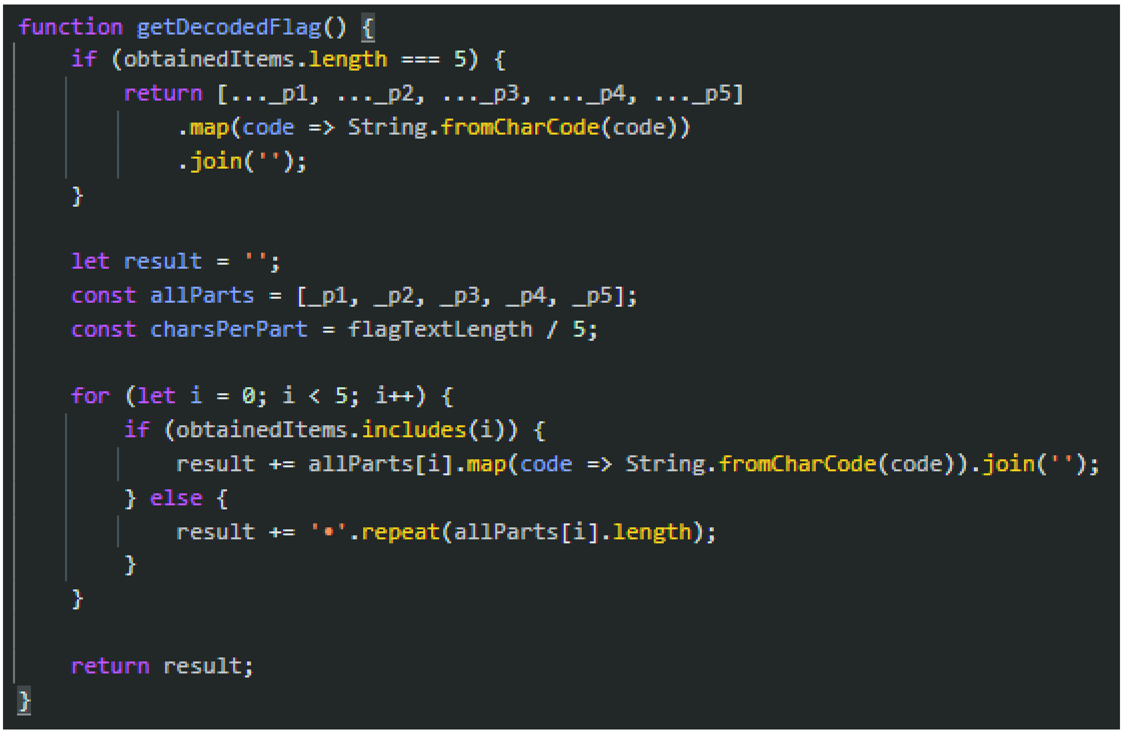
Screw this, I’m cheating.

So what can we do? Let’s start by checking out the source. Not much there. The only thing of interest is the script-tag that points us to what seems to be the game logic. We’ll open up the dev tools in the browser with F12 to inspect what is happening in the code.



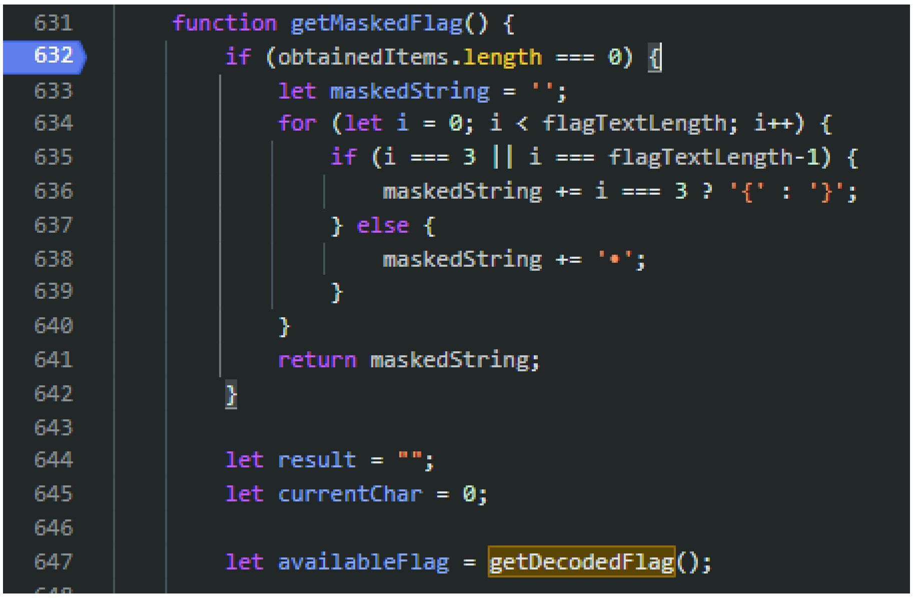
```
function getMaskedFlag() {
    if (obtainedItems.length === 0) {
        let maskedString = '';
        for (let i = 0; i < flagTextLength; i++) {
            if (i === 3 || i === flagTextLength-1) {
                maskedString += i === 3 ? '{' : '}';
            } else {
                maskedString += '•';
            }
        }
        return maskedString;
    }

    let result = "";
    let currentChar = 0;

    let availableFlag = getDecodedFlag();

    const charsPerPiece = Math.ceil(flagTextLength / 5);

    for (let i = 0; i < flagTextLength; i++) {
        const pieceNeeded = Math.floor(i / charsPerPiece);
        const shouldReveal = obtainedItems.includes(pieceNeeded);

        if (shouldReveal || "{}[]()_-".includes(availableFlag[i])) {
            result += availableFlag[i] || '•';
        } else {
            result += '•';
        }
    }
```

I found this interesting function. It seems to call the function getDecodedFlag() if obtainedItems.length is not equal to 0. Let’s look at the function getDecodedFlag().



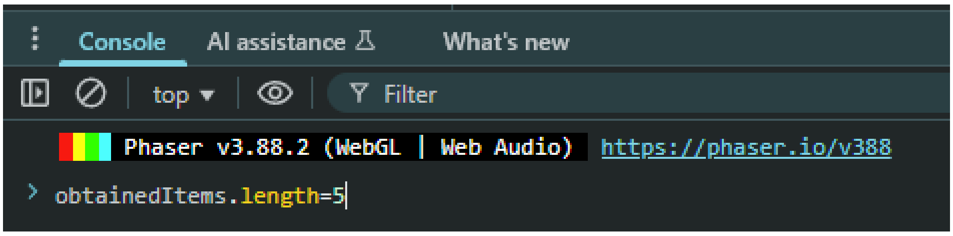
```
function getDecodedFlag() {
    if (obtainedItems.length === 5) {
        return [..._p1, ..._p2, ..._p3, ..._p4, ..._p5]
            .map(code => String.fromCharCode(code))
            .join('');
    }

    let result = '';
    const allParts = [_p1, _p2, _p3, _p4, _p5];
    const charsPerPart = flagTextLength / 5;

    for (let i = 0; i < 5; i++) {
        if (obtainedItems.includes(i)) {
            result += allParts[i].map(code => String.fromCharCode(code)).join('');
        } else {
            result += '•'.repeat(allParts[i].length);
        }
    }

    return result;
}
```

This seems to convert an array with CharCodes to their respective characters if obtainedItems.length is equal to five. Let’s try to set a breakpoint in the code and se if we can manually set the value to 5 and be able to get the decoded flag.



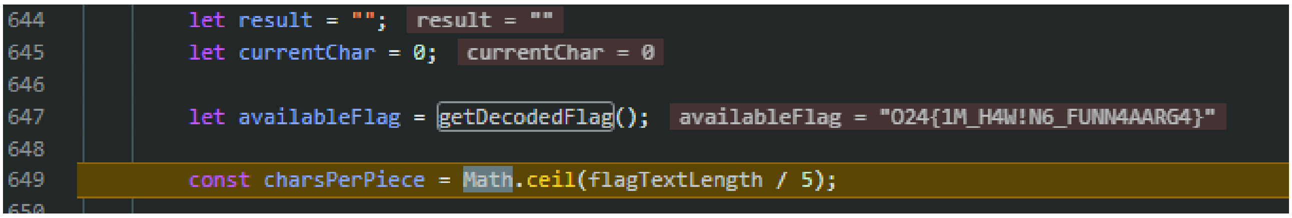
```
    function getMaskedFlag() {
        if (obtainedItems.length === 0) {
            let maskedString = '';
            for (let i = 0; i < flagTextLength; i++) {
                if (i === 3 || i === flagTextLength-1) {
                    maskedString += i === 3 ? '{' : '}';
                } else {
                    maskedString += '•';
                }
            }
            return maskedString;
        }

        let result = "";
        let currentChar = 0;

        let availableFlag = getDecodedFlag();
```

We’ll set the breakpoint there to see if we can make it skip the if expression. Let’s reload the site.



```
Phaser v3.88.2 (WebGL | Web Audio) https://phaser.io/v388
> obtainedItems.length=5
```

Now that it has reloaded and stopped at the breakpoint. We’ll use the console to set the value to 5 manually. And let’s step forward in the execution a bit.

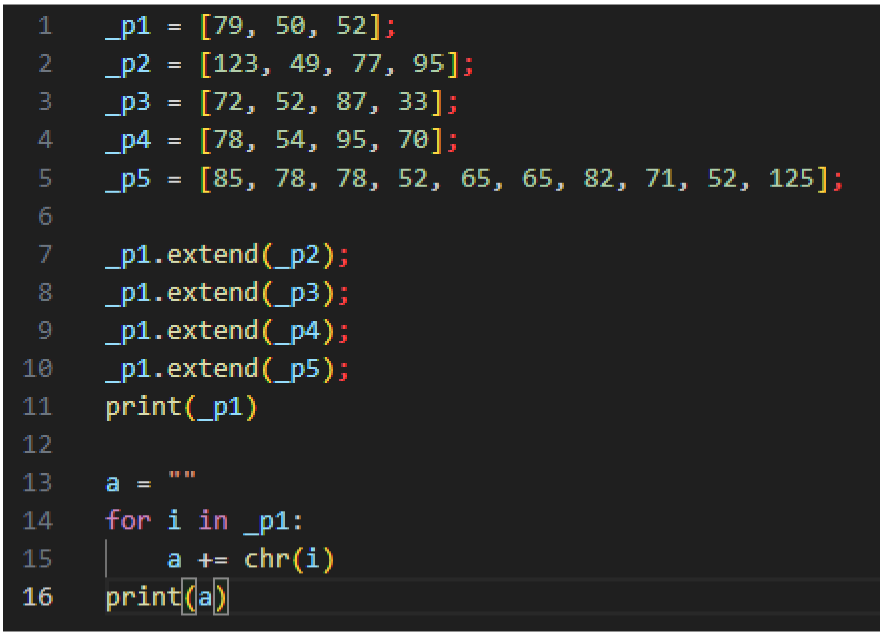


```
        let result = "";  result = ""
        let currentChar = 0;  currentChar = 0

        let availableFlag = getDecodedFlag();  availableFlag = "O24{1M_H4W!N6_FUNN4AARG4}"

        const charsPerPiece = Math.ceil(flagTextLength / 5);
```

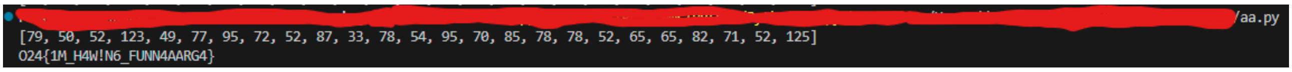
Now we can see our flag as the value of the variable availableFlag
O24{1M_H4W!N6_FUNN4AARG4}

Alternate solution:

Take the arrays and copy them into a python script. Make it into a single long list. Then do a for loop which takes each value and turns it into a character which it then appends to a string and print out the string.



```
_p1 = [79, 50, 52];
_p2 = [123, 49, 77, 95];
_p3 = [72, 52, 87, 33];
_p4 = [78, 54, 95, 70];
_p5 = [85, 78, 78, 52, 65, 65, 82, 71, 52, 125];

_p1.extend(_p2);
_p1.extend(_p3);
_p1.extend(_p4);
_p1.extend(_p5);
print(_p1)

a = ""
for i in _p1:
    a += chr(i)
print(a)
```



```
[redacted]
[79, 50, 52, 123, 49, 77, 95, 72, 52, 87, 33, 78, 54, 95, 70, 85, 78, 78, 52, 65, 65, 82, 71, 52, 125]
O24{1M_H4W!N6_FUNN4AARG4}
```