Exercise on the for-of loop

We will use the for-of loop to search through data and obtain our desired values.

Exercise 1:

Open the developer tools on any website. Locate the first character of all headings, and log the concatenation of the first characters.

Exercise 1 | Solution

```
let text = '';
let nodes = // Write code here

for ( let node of nodes ) {
    // Write code here
};

console.log( text );
```

Explanation:

The data from the headings can be obtained in several ways. The solution uses

`document.querySelectorAll('h1', 'h2', 'h3', 'h4', 'h5', 'h6')`

This query returns those 6 headings in the form of nodes. The data is in the child of the node. Hence, we use `node.childNodes[0].textContent[0]` where `textContent[0]` refers to the first character of the text.

The rest is plain concatenation in a for-of loop.

Exercise 2:

Assemble a string containing all emojis between the hexadecimal codes `1F601` and `1F64F` in the respective order. Use the for-of loop.

Reference: http://apps.timwhitlock.info/emoji/tables/unicode

Reference: http://apps.timwhitlock.info/emoji/tables/unicode

`String.fromCodePoint` converts a decimal number into a character, even if it is a 4-byte long number.

Explanation:

In the first 4 lines of our code, we define our prefixes and ranges because we know all of them and we know how they progress:

```
let prefix = '1F6';
let digits4 = '01234';
let digits5 = '01234567890ABCDEF';
let emojis = '';
```

Then, using a nested loop, we traverse each element of `digits4`, appending it to the prefix, and then further concatenating each element from `digits5` to this hexadecimal value, `hex`.

This value is converted to string using `String.fromCodePoint`.