

PIC 40A: Homework 2 (due 10/19 at 10pm)

Okay... so I lied - oops. Just for the homework(s) on pure JavaScript, no submission to the PIC server is necessary! In this assignment you will make one file called `HW2.js` and all you need to do is submit this file to CCLE before the deadline.

1. For this question I need to make a definition. I promise it is not as complicated as it looks!

I have not made this definition as general as possible because otherwise you would hate me:
<https://stackoverflow.com/questions/1969232/>.

Definition. A **cookie** is a special type of **string**.

It has the form `"name1=value1; name2=value2; ...; lastName=lastValue"`.

The **names** and **values** can be any **non-empty** sequence of ASCII characters which are:

- (a) alphanumeric characters: `a, b, c, ..., z, A, B, C, ..., Z, 0, 1, 2, ..., 9`, or
- (b) a character appearing in the following string: `"_.!+-*'`|&%#~^"`.

In particular, the following characters are **not allowed**:

- white space
- `;` (semicolons)
- `,` (commas)
- `=` (equals)
- control characters `'\x00'` to `'\x1F'`, and `'\x7F'`

For example, the following are examples of cookies:

- `"first_name=Michael; last_name=Andrews; username=mjandr"`
- `"username=mjandr; first_name=Michael; last_name=Andrews"`
- `"_ga=GA1.2.34.56; dwf_sg_task_complete=False; lux_uid=888; _gid=GA1.2.88.88"`
- `"__stripe_mid=c4d6a-723-3ee-54d-640e5af; csrftoken=Kger31Gtcvyt%2F2ILWuQJoJ"`

The following is **not** a cookie `"name=Michael Andrews; position=PIC;assistant,adjunct"` because of the space between `Michael` and `Andrews`, as well as the semi-colon and comma in `PIC;assistant,adjunct`.

Finally, here is the actual question...

Define a function whose function comment reads as follows.

```
/**
This function extracts from a given cookie
the 'value' corresponding to the 'name' "username".

For example, both of the following function calls return "mjandr":
. extract_username("first_name=Michael; last_name=Andrews; username=mjandr");
. extract_username("username=mjandr; first_name=Michael; last_name=Andrews");

If the given cookie has no 'name' called "username",
then the function returns the empty string.

For example, we have
. extract_username("common_error=Micheal; " +
                  "another_one=Andrew; another=Andrew_Michaels") === ""

@param {string} cookie : the cookie to extract information from
@return {string} the 'value' corresponding to the 'name' "username"
                  the empty string if "username" is not a 'name'
*/
```

Just like `string::find` in C++, JavaScript's `String.prototype.indexOf` has an optional second argument. You might find this useful.

2. Define a function whose function comment reads as follows.

```
/**
Returns the result of rolling two dice.
Here are some possible return values.
. '1 + 2 = 3'
. '6 + 4 = 10'
. '3 + 5 = 8'
. '2 + 2 = 4'

The probability distribution of the return values
is the same as rolling two dice in real life.

@return {string} the result of rolling two dice
*/
```

3. Define a function whose function comment reads as follows.

```
/**
For a specified array of booleans,
the function returns the sum of the indices for which
the corresponding value in the array is true.

For example,
sum_checked_indices([false, true, false, true, true]) === 1 + 3 + 4 === 8.

@param {Array} arr : an array consisting of booleans
                    (whenever 0 <= i < arr.length, arr[i] !== undefined)
@return {number} the sum of the indices i for which arr[i] === true
*/
```