CAA 1

Web Programming

Alejandro Pérez Bueno

Nov 16, 2023

Table of Contents

Question 1 2
Question 2 2
Question 3 3
Question 4 3

Question 1

```
let test = 0 || 'default';

if (!test) {
  console.log('A');
  } else if (test === 0) {
    console.log('B');
  } else {
    console.log(test);
}
```

The answer is `C: default`. The value of the `test` variable is `default`, because the logical OR operation in the first line interprets the first zero as `false`, and thus assigns the second variable, as it is not considered a 'false' value. The `if` statements below do the following:

1. `!test`: returns true if `test` is falsy, i.e. if it is false, zero, an empty string, null, undefined, or NaN. In the code this condition **is not met**.
2. `test === 0` returns true if `test` is zero and `test` is an integer (both sides must be of the same type). In the code this condition **is not met**, because `test` is a string containing `default`.
3. The last condition is an `else` statement, which **is met** because all prior conditions are not met. This part of the code prints out the value of `test`.

Question 2

```
let items = [1, 2, 3, 4, 5];

items.forEach((item, index) => {
  if (index % 2 === 0) {
    continue;
  }
  console.log(item);
});
```

The answer is `C: an error`, because the `forEach` statement does not allow for a `continue` statement. A `for of` loop could have been used. I understand that if a proper loop was used, the code would have instead printed the odd numbers in the `items` array: `1`, `3` and `5`.

Question 3

```
let items = ['hall', 'desire', 'low', 'bill', 'own'];

const res = items.reduce((msg, current, index) => {
  const pos = (index % 2) ? current.length - 1 : 0;
  return msg + current.charAt(pos);
});
console.log(res);
```

The answer is (again) `C: hello`. The code iterates over the elements of the `items` array and creates a string with this criteria:

- Even elements (starting from zero) in the array: add the first character of the string (index `0`) to the output
- Odd elements in the array: add the last character of the string (index `current.length - 1`) to the output

Thus, here is the array highlighting the selected characters that create the output string:

'**h**all', 'desir**e**', '**l**ow', 'bil**l**', '**o**wn' -> `hello`

Question 4

The full code would be:

```
if (!Object.entries) {                                                    ①
  Object.entries = function(obj) {
    var ownProps = Object.keys(obj),
        i = ownProps.length,
        resArray = new Array(i);
    while (i--)
      resArray[i] = [ownProps[i], obj[ownProps[i]]];
    return resArray;
  };
}

var o = {                                                                 ②
  first: 'first',
  second: 'second',
  third: 'third',
  last: ' '
};
```

```
Object.entries(o).forEach(function([key, value]) {                    ③
  console.log(key + ': ' + value);
});
```

① If `Object.entries` does not exist (not implemented in older browsers), a new polyfill function is created, which creates an array of arrays containing key-value pairs from an object.

② Same values for the first second third and last attributes (from the CAT subject).

③ Calling the Object.entries to print out every key and value in the `o` variable, regardless of if you use an old browser.