1. Write the **getStats(arr)** function which receives an Array as a parameter and prints a message regarding the types of elements found inside it.

var arr = [1, 2, 'hello', NaN, {

city: 'IasI',

zip: null

}, [11, 12], undefined, undefined, undefined]

getStats(arr); // 2 numbers, 1 string, 1 NaN, 1 Object, 1 Array, 3 undefined

1. Write a unary function addF which accepts 1 param:

* **x** - number

and returns another **unary function** with which, when called, returns the sum between it's param and x.

let add = addF(13);

add(10); // 23

add(-5); // 8

1. Write a limit function with 2 params:

* **fn** - a function
* **max** - number

and returns another function which behaves just like fn but can be called just max number of times. After that it will return undefined.

let limitAdd = limit(add, 2);

limitAdd(3, 5); // 8

limitAdd(11, 23); //34

limitAdd(5, 10); // undefined

function add(x, y) {

return x + y;

}

1. *Make a function that looks through an****array of objects****(first argument) and****returns an array of all objects****that have****matching name and value pairs****(second argument). Each name and value pair of the source object has to be present in the object from the collection if it is to be included in the returned array. You are not allowed to use a for loop. You should instead use iterable/iterator methods.*

For example, if the first argument is [{ first: “Romeo”, last: “Montague” }, { first: “Mercutio”, last: null }, { first: “Tybalt”, last: “Capulet” }], and the second argument is { last: “Capulet” }, then you must return the third object from the array (the first argument), because it contains the name and its value, that was passed on as the second argument.

**Examples:**

whatIsInAName([{ first: "Romeo", last: "Montague" }, { first: "Mercutio", last: null }, { first: "Tybalt", last: "Capulet" }], { last: "Capulet" }) should return [{ first: "Tybalt", last: "Capulet" }]

whatIsInAName([{ "apple": 1 }, { "apple": 1 }, { "apple": 1, "bat": 2 }], { "apple": 1 }) should return [{ "apple": 1 }, { "apple": 1 }, { "apple": 1, "bat": 2 }]

whatIsInAName([{"a": 1, "b": 2, "c": 3}], {"a": 1, "b": 9999, "c": 3}) should return []

1. Given a string that contains space-separated words, write a function that returns a map that counts occurrences of each word in the input string.
2. Extend the **Array.prototype** with a new function, **transform**, which behaves just like [Array.prototype.map](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map" \o "https://developer.mozilla.org/en-us/docs/web/javascript/reference/global_objects/array/map" \t "_blank)

Array.prototype.transform = function ArrayTransform(…) {

…

}

1. Extend the **Array.prototype** with a new function, **keep**, which behaves just like [Array.prototype.filter](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter" \o "https://developer.mozilla.org/en-us/docs/web/javascript/reference/global_objects/array/filter" \t "_blank)

Array.prototype.keep = function ArrayKeep(…) {

…

}