Let's break down the code step-by-step to understand what will be output to the console:

1. `var arr1 = "john".split('');`: This line creates an array `arr1` by splitting the string `"john"` into individual characters. So, `arr1` will be `['j', 'o', 'h', 'n']`.

2. `var arr2 = arr1.reverse();`: This line assigns the reference of `arr1` to `arr2` and then reverses `arr2` in place. The `reverse()` method modifies the original array and also returns a reference to the reversed array. So, both `arr1` and `arr2` will point to the same reversed array, which is `['n', 'h', 'o', 'j']`.

3. `var arr3 = "jones".split('');`: This line creates another array `arr3` by splitting the string `"jones"` into individual characters. So, `arr3` will be `['j', 'o', 'n', 'e', 's']`.

4. `arr2.push(arr3);`: This line appends `arr3` as a single element to the end of the array referred to by `arr2`. Since `arr2` and `arr1` are referencing the same array, this also affects `arr1`. So, after this line, `arr1` and `arr2` will both be `['n', 'h', 'o', 'j', ['j', 'o', 'n', 'e', 's']]`.

5. `console.log("array 1: length=" + arr1.length + " last=" + arr1.slice(-1));`: This line logs the length of `arr1` (5) and the last element of `arr1` (which is `['j', 'o', 'n', 'e', 's']` due to the nested array). So, the output will be: `array 1: length=5 last=j,o,n,e,s`.

6. `console.log("array 2: length=" + arr2.length + " last=" + arr2.slice(-1));`: This line logs the length of `arr2` (5) and the last element of `arr2`, which is also `['j', 'o', 'n', 'e', 's']` due to both `arr1` and `arr2` pointing to the same array. So, the output will be: `array 2: length=5 last=j,o,n,e,s`.

In summary, both `arr1` and `arr2` are referencing the same array, which contains the elements `['n', 'h', 'o', 'j', ['j', 'o', 'n', 'e', 's']]`. Therefore, when you modify one array, the other array will be affected as well.