***Mảng***

**push(x)**:thêm cuối

* Với x là giá trị tham số cần thêm cuối

**join(‘ ’)**: chuyển mảng thành chuỗi, các ptu mảng dc liên kết với nhau bởi ’ ‘ (để hiển thị)

**reverse()**: đảo ngược mảng

**sort()**:Sắp xếp các phần tử của mảng (theo alpha)

**concat():**nối 2 mảng với nhau. Trả về mảng mới là mảng gộp của 2 mảng vừa nối

* Arr1.concat(arr2)

**pop():** xóa cuối

**shift()** : xóa đầu

**unshift():** thêm đầu

* Unshift(x)

**toString() :** biến mảng thành chuỗi

**splice:** trả về mảng đã dc xóa (k có hiển thị)

*2 tham số: splice(start, count)*

* + ts 1: vị trí bắt đầu xóa
  + Ts 2: số lượng phần tử xóa từ vị trí đó

*3 tham số: splice( start, count, item)*

* + Ts 1: vị trí thêm/xóa (vị trí bắt đầu)
  + Ts 2: số lượng xóa (để 0 nếu muốn thêm)
  + Ts 3: giá trị phần tử cần thêm

**indexOf(x):** trả về vị trí đầu tiên của phần tử trong mảng có giá trị = x, nếu k thấy trả về -1

**lastIndexOf(value):** Tương tự indexOf() nhưng trả về vị trí cuối.

**includes(x):** kiểm tra mảng có chứ ptu = x hay không, trả về true hoặc false.

**\*\*\*map(callback**): Tạo một mảng mới với các phần tử được biến đổi bởi hàm callback

**Slice(start,end):** trích xuât một phần của mảng thành một mảng mới // vd 1 2 3 4 5 6 :slice(1,3) -> trả về mảng: 2,3 => chỉ lấy ở vị trí 1,2 nghĩa là index end k lấy.

***Xâu***

*1. Tìm kiếm và thay thế*:

**indexOf(substring):** trả về vị trí đầu tiên của chuỗi con xuất hiện trong xâu, k có trả về -1

**lastIndexOf(substring):** Tương tự indexOf(), nhưng trả về vị trí cuối

**includes(substring)**: kiểm tra xem có chứa chuỗi con k, trả về true false

**startsWith(substring):** kiểm tra chuỗi có bắt đầu bằng chuỗi con không

**endsWith(substring):** kiểm tra chuỗi có kết thúc bằng chuỗi con không

*2. Cắt và trích xuất:*

**slice(startIndex, endIndex):** Trích xuất một phần của chuỗi từ startIndex đến endIndex (không bao gồm endIndex).

**substring(startIndex, endIndex):** Tương tự slice(), nhưng không hỗ trợ chỉ số âm.(trích xuất chuỗi từ chuỗi)

**substr(startIndex, length):** trích xuất chuỗi con bắt đầu từ ptu đầu với độ dài..

*3. Biến đổi*:

**toLowerCase():** Chuyển đổi chuỗi thành chữ thường.

**toUpperCase():** Chuyển đổi chuỗi thành chữ hoa.

**trim():** Loại bỏ khoảng trắng ở đầu và cuối chuỗi.

**concat(...strings):** Nối hai hoặc nhiều chuỗi lại với nhau. A.concat(B)

**split(separator):** tách chuỗi thành mảng dựa trên kí tự phân cách, vd split(‘,’)

*4. Kiểm tra:*

**charAt(index):** Trả về về kí tự có vị trí .