|  |  |  |
| --- | --- | --- |
| *Метод* | *Описание* | |
| let newArr = [ ] | Създаване на нов масив |  |
| console.log(arr[arr.length-1]) | Взима последния елем. от масивът | |
|  |  |  |
| arr.push(‘пример ’) | Добавя нова ст-ст накрая на масивът | |
| arr.toString( ) | Обръща масивът в стринг | |
| arr.join(“ - ”) | Събира елементите в стринг и определя как да бъдат разделени. Повече се използва,когато принтираме на конзолата. | |
| arr.includes(‘Gogo’) | Проверява дали има даден параметър в масива.Този метод връща **True** или **False** | |
| let remove = arr.shift() | Този метод изрязва елем.от масива.**Винаги взима нулевият индекс!** | |
| let type =arr.map(Number) | С този метод задаваме какъв тип да ни изкара от масива. | |
| let arr = name.split() или (“ ”) | Превръща стринг в масив.Може да раздели също и по даден символ. | |
| arr.reverse() | Обръща масива. |  |
| *arr.join(‘’)* | *Прави масива на стринг* |  |
|  |  |  |
| ***Основни Методи за Масиви*** | Описание | |
| arr.**push**( ) | Добавя нова ст-ст накрая на масивът. | |
| arr.**pop** ( ) | Изрязва последния елемент от масива.Може да го запазим в променлива. | |
| arr.**unshift**( ) | Добавя нов елем. в масивът като нулев индекс. | |
| arr.**shift**( ) | Изрязва първия елемент от масива. Може да го запазим в променлива. | |
|  |  | |
| let newArr = arr.**slice**() | Копира масива | |
| let newA = arr.**slice**(0,2) | Копира масива от зададения индекс (0), и колко елемента да копира (2) от масива. | |
|  |  | |
| let newArr = arr.**splice** (1,3) | Променя ориг.масив,като изрязва дадени елем.от него и ги запазва в променлива.  Пр: От 1 индекс изрязва 3 елемента от масива на дясно. | |
| let newArr = arr.**splice**(1,3,’6’,’Gogo’) | Пр: От 1 индекс изрязва 3 елемента от масива на дясно и добавяме нови елемменти в масива. | |
|  |  | |
| let newArr = arr.map(x=>x\*2) | Map методът създава нов масив с трансформирани елементи от стария | |
| arr.filter() | Създава нов масив с филтрирани елементи: let evenNum = arr.filter(arr=>arr%2===0) | |  |
| arr.sort() | Сортира елем. от масива като го пренарежда по ASCII кодовете на елементите. | |  |
| arr.sort((a,b)=> a-b) | Сортира елементите по възходящ ред. | |  |
| arr.**sort((a,b)=>a.localCompare(b))** | Сортира стрингов масив по азб.ред | |  |
|  |  | |  |
| let repeated = arr.**indexOf**(‘name’) | Запазва в променливата No на индекса с този стринг.**Така можем да изтрием точен индекс от масива.** | |  |
| arr.**forEach**(elem => {console.log(elem)}) | Завърта всеки един елемент от масива и го принтира по отделно. | |  |
| arr.**concat**(arr2) | Конкатенира два масива един за друг. | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

*1.Масиви*

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

*2.Обекти*

|  |  |
| --- | --- |
| *метод* | *описание* |
| *let obj = {name:’Gogo’}* | *Създава обект! { key:value} на обекта.* |
| *console.log (obj.name)* | *Принтира на конзолата данните на обекта с избрания ключ.* |
| *delete obj.age* | *Изтрива избран ‘ключ’ и неговата ст-ст от дадения обект.* |
| *Object.keys(obj)* | *Показва ключовете на обекта като масив.* |
| *Object.values(obj)* | *Показва ст-стите на ключовете като масив.* |
| *Object.entries(obj)* | *Показва масив(и) ,които съдържат key/value ст-сти. пр:(name ,‘Gogo’)* |
| ***for - in*** | *Цикъл ,който се използва САМО за обекти.* |
| *obj.hasOwnProperty(променлива)* | *Методът проверява дали даден ключ го има в обекта.* |
| *obj.find(el=>el.name===name)* | *Метод ,който намира първият елемент отговарящ на даденото условие.* |
| *let obj=JSON.parse(string)* | *Превръща JSON стринг в обект.* |
| *let objTwo=JSON.strringify(obj)* | *‘Стрингосваме’обект в JSON формат.Прави референцията на стринг.* |
|  |  |
| *object[curentProduct]=arr[index+1]* | *Така добавяме ст-ст в обект (с нейния ключ и value) .виж зад.5 от упр.* |
| *let movie = movies.find (el=> el.name === name);* | *отговаря за първия елемент,който оотг на даденото условие. (проверява дали има такоа име в обекта)* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

*3.Числа*

|  |  |
| --- | --- |
| *функция* | *описание* |
| *let res = Math.pow (2,3)* | *Тази функция пресмята степенуване 2 на 3 степен* |
| *let res = Math.min(n1,n2,n3,и т.н.)* | *Тази ф-я взима най-малкото число от избраните* |
| *let res = Math.max(n1,n2,n3)* | *Тази ф-я взима най-голямото число от изброените.* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

*4.Стрингове*

|  |  |
| --- | --- |
| *функция* | *описание* |
| *let range = пром.charCodeAt()* | *Показва No. на символа в ASCII таблицата* |
| *console.log(String.fromCharCodeAt(61))* | *Показва символа с No от ASCII таблицата – „=”* |
| *arr.join(‘’)* | *Прави масива на стринг* |
| *let …..= input.split(‘ // ’)* | *Разделя стринга по зададеното в скобите,като го прави на масив!* |
| *let …=string.lastIndexOf(‘.’)* | *Намира последният индекс с този символ в стрингът.Връща число!* |
| *let … = пром.substring(0,10)* | *Изрязва от дадения стринг зададеното в (),като взима от 0-9!*  *Ако има само едно число-изрязва до края на стринга!* |
| *let … = text.toLowerCase()* | *Конвертира целия стринг с малки букви* |
| *let … = text.toUpperCase()* | *Конвертира целия стринг с големи букви* |
| *let … = text.match(‘word’)* | *Сравнява дали има такава дума в текста.При първото съвпадение спира да проверява! string.match => променливата става масив!!!* |
| *let … = text.matchAll(‘word’)* | *Намира всички съвпадения с дадената дума! string.match => променливата става масив!!!* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

*RegExp*

|  |  |
| --- | --- |
| ***Когато използвам РегЕкс,след като разделя всички данни в масив,завъртам масива. Създавам променлива,която мачва всеки елемент от масива дали съвпада с модела!След това проверявам (if(пром !== null)) дали променливата има съвпадение.Ако има, (тя е масив),който го завъртам и вече мога да взимам данни от групите на модела!*** | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |