|  |  |
| --- | --- |
| **CODEGYM** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc** |

**BÀI HỌC NGÀY 16.07.2021**

**Học viên:** Trần Tiến Dũng

**Lớp:** C0621G1

|  |  |
| --- | --- |
| **MẢNG** | |
| 1. **Mảng**   - *Mục tiêu*   * *Định nghĩa* * *Thêm phần tử vào mảng* * *Xóa phần tử của mảng* * *Sắp xếp mảng* * *Duyệt các phần tử của mảng*  1. **Mảng một chiều**  * *Mục tiêu*  1. **Cú pháp khai báo mảng**  * *Cách 1* * *Cách 2* * *Lưu ý hàm so sánh*  1. **Mảng hai chiều**  * Mục tiêu * Định nghĩa * Số chiều của mảng * Mảng hai chiều | **+** Mô tả được mảng  + Tạo được mảng  + Sử dụng được các thuộc tính và phương thức cơ bản của mảng  + Thao tác được với các phần tử của mảng  + Thêm được phần tử vào mảng  + Mô tả được cách sử dụng vòng lặp for để duyệt mảng  + Sử dụng được phương thức forEach để duyệt mảng  + Là một loại biến đặc biệt, có thể lưu được nhiều giá trị thay vì chỉ một giá trị như các biến thông thường  + Mỗi giá trị trong mảng được gọi là một phần tử  + Các phần tử được lưu trữ ở các vị trí kế tiếp nhau trong bộ nhớ  + Các khải niệm của mảng:   1. Tên mảng: Tuân thủ theo quy tắc đặt tên của biến 2. Phần tử: Các giá trị được lưu trữ trong mảng 3. Chỉ số: Vị trí của các phần tử trong mảng. Chỉ số bắt đầu từ 0. 4. Độ dài: Số lượng các phần tử trong mảng   + Phương thức Push() cho phép thêm phần tử vào cuối mảng.  + Cú pháp tên mảng.push(giá trị thêm vào mảng).  + Có thể thêm phần tử vào mảng bằng cách gán trị cho phần tử ở vị trí length  VD: var fruits = [“Banana”, “Orange”, “Apple”, “Mango” ];  fruits[fruits.length] = “Lemon”  + Sử dụng hàm pop() sẽ xóa đi phần tử cuối cùng của mảng  VD: var fruits = [“Banana”, “Orange”, “Apple”, “Mango”];  Fruits.pop() = [“Banana”, “Orange”, “Apple”, ]  + Sử dụng hàm shift() sẽ xóa đi phần tử đầu tiên của mảng  VD: var fruits = [“Banana”, “Orange”, “Apple”, “Mango”];  Fruits.shift() = “Orange”, “Apple”, “Mango”]; ]  + Hàm sort() sẽ giúp sắp xếp mảng theo một trật tự nhất định  VD: let arr = ["Toyota", "Subaru", "BMW"]  arr.sort() // [ "Toyota", "Subaru", "BMW" ]  + Hàm sort() sẽ đảo ngược trật tự của một mảng  VD: let arr = ["Toyota", "Subaru", "BMW"]  arr.sort() // [ "BMW", "Subaru", "Toyota" ]  arr.reverse() // [ "Toyota", "Subaru", "BMW" ]  + Sử dụng vòng lặp for để duyệt qua tất cả các phần tử của mảng  VD: var fruits = [“Banana”, “Orange”, “Apple”, “Mango” ];  For (i = 0; i < fruits.length; i++){  text += fruits[i] + “<br/>”  + Phương thức forEach của mảng cho phép thao tác lần lượt với từng phần tử  Var numbers = [4, 9, 16, 25];  Number.forEach(function(item){  Document.write(item + “</br>”);  });  + Khai báo và sử dụng được mảng một chiều  + Sử dụng vòng lặp để làm việc với mảng  + Sử dụng dấu ngoặc vuông [] để khai báo mảng:  VD: var arr = [element1, element2, element3]  + Sử dụng từ khóa new:  VD: var arr = new Array(element1, element2, element3);  + Hàm so sánh là một hàm có mục đích giúp so sánh 2 phần tử  + Hàm so sánh có 2 tham số  + Hàm so sánh trả về giá trị kiểu số: âm (nhỏ hơn 0), dương (lớn hơn 0) và 0.   1. âm có nghĩa là phần tử đầu nhỏ hơn phần tử sau 2. dương có nghĩa là phần tử đầu lớn hơn phần tử sau 3. 0 có nghĩa là 2 phần tử bằng nhau   + Mô tả được mảng đa chiều  + Khai báo được mảng hai chiều  + Duyệt được mảng hai chiều sử dụng vòng lặp lồng nhau  + Là mảng có các phần tử là các mảng khác  + Có thể có mảng 2 chiều, 3 chiều… hoặc nhiều hơn  + Mảng càng nhiều chiều thì độ phức tạp khi xử lý càng cao  + Mảng 1 chiều chỉ cần 1 chỉ số để xác định vị trí của phần tử mảng  + Mảng 2 chiều cần 2 chỉ số để xác định vị trí của phần tử mảng  + Mảng 3 chiều cần 3 chỉ số để xác định vị trí của phần tử mảng  + Là mảng nhiều chiều được sử dụng phổ biến  + Bao gồm các phần tử, mỗi phần tử là một mảng một chiều  + là một bảng gồm n dòng và m cột |
|  | |