**ĐỐI TƯỢNG MẢNG VÀ HÀM**

**A – LÝ THUYẾT**

**I - ĐỐI TƯỢNG MẢNG**

**1. Định nghĩa**

Mảng là một đối tượng (hay có thể coi nó là một kiểu dữ liệu) được sử dụng để lưu trữ nhiều giá trị trong một biến duy nhất.

**2. Khai báo một mảng**

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ]; |

**Trong đó:**

* Giá trị của các phần tử trong mảng được phân cách với nhau bởi dấu “,”
* Các phần tử trong mảng được ngầm đánh số theo thứ tự bắt đầu từ 0, tiếp theo là 1, 2,…

**3. Xuất giá trị của một phần tử mảng**

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  document.write(courses[2]); // NodeJS |

**4. Xuất giá trị của tất cả các phần tử vào mảng**

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  document.write(courses); // Javascript, ReactJS, NodeJS |

**5. Thêm một phần tử vào mảng**

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  courses[3] = "jQuery";  document.write(courses); // Javascript, ReactJS, NodeJS, jQuery |

**6. Duyệt mảng**

Duyệt mảng là cách thức truy cập vào từng phần tử của mảng để lấy ra vị trí cũng như giá trị tương ứng của phần tử đó trong mảng

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  for(key in courses){  document.write(courses[key]+"<br/>");  }  // Javascript  // ReactJS  // NodeJS |

**III - HÀM TRONG JAVASCRIPT**

**1. Định nghĩa Hàm**

Hàm hay còn được gọi là Function được sử dụng để đóng gói nhóm mã xây dựng nên một chức năng nhất định

**2. Hàm tự định nghĩa không có tham số**

**Code:**

|  |
| --- |
| function tinhTong(){  var st1 = 5;  var st2 = 10;  var kq = st1 + st2;  document.write(kq);  }  tinhTong(); // 15 |

**Trong đó:**

* Quy tắc đặt tên hàm giống với cách đặt tên biến
* Tên là những từ ghép thì viết hoa các ký tự đầu tiên của các từ tiếp theo
* Thực thi hàm (chạy hàm) bằng cách gọi tới tên của hàm kèm theo cặp dấu ngoặc tròn

**3. Hàm tự định nghĩa có tham số**

**Code:**

|  |
| --- |
| function tinhTong(st1, st2){  var kq = st1 + st2;  document.write(kq);  }  tinhTong(100, 150); // 250 |

**4. Hàm tự định nghĩa có giá trị trả về**

**Code:**

|  |
| --- |
| function tinhTong(st1, st2){  var kq = st1 + st2;  return kq;  }  document.write(tinhTong(100, 150) + 250); // 500 |

**III – CÁC HÀM & PHƯƠNG THỨC ĐƯỢC XÂY DỰNG SẴN TRONG JAVASCRIPT**

**1. Các phương thức thao tác mới mảng**

**a. Phương thức length**

Phương thức **length** dùng để đếm số phần tử có trong mảng

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  document.write(courses.length); // 3 |

**b. Phương thức concat()**

Phương thức concat() dùng để gom tất cả các phần tử của nhiều mảng riêng rẽ tập trung vào một mảng

**Code:**

|  |
| --- |
| var courses1 = [  "Javascript",  "jQuery"  ];  var courses2 = [  "ReactJS",  "NodeJS"  ];  var courses = courses1.concat(courses2);  document.write(courses); // Javascript, jQuery, ReactJS, NodeJS |

**c. Phương thức join()**

Phương thức join() giúp chúng ta tạo ra một chuỗi từ các phần tử trong một mảng và được ngăn cách bằng một ký tự bấy kỳ

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  var str = courses.join("-")  document.write(str); // Javascript-ReactJS-NodeJS |

**d. Phương thức pop()**

Phương thức pop() dùng để xóa phần tử cuối cùng trong một mảng

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  courses.pop();  document.write(courses); // Javascript, ReactJS |

**e. Phương thức push()**

Phương thức push() dùng để thêm một hoặc nhiều phần tử vào cuối mảng

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  courses.push("VueJS", "AngularJS");  document.write(courses); // Javascript,ReactJS,NodeJS,VueJS,AngularJS |

**f. Phương thức reverse()**

Phương thức reverse() trả lại một mảng với thứ tự các phần tử bị đảo ngược

**Code:**

|  |
| --- |
| var courses = [  "Javascript",  "ReactJS",  "NodeJS"  ];  courses.reverse();  document.write(courses); // NodeJS,ReactJS,Javascript |

**2. Các phương thức của đối tượng toán học Math**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương Thức | Chức Năng | Ví Dụ |
| 1 | Math.round(x) | Làm tròn | Math.round(5.6) = 6 |
| 2 | Math.floor(x) | Làm tròn xuống | Math.round(4.5) = 4 |
| 3 | Math.random() | Lấy số ngẫu nhiên 0 < x < 1 | Math.round(x) = 0.73… |
| 4 | Math.max(x,y,z,…N) | Lấy số có giá trị lớn nhất trong chuỗi số | Math.max(1,5,3,7) = 7 |
| 5 | Math.min(x,y,z,…N) | Lấy số có giá trị nhỏ nhất trong chuỗi số | Math.max(1,5,3,7) = 1 |
| 6 | Math.abs(x) | Trả về giá trị số dương | Math.abs(-9.5) = 9.5 |
| 7 | Math.PI | Hằng số PI | Math.PI = 3.14… |
| 8 | Math.sqrt(x) | Lấy căn bậc 2 của một số | Math.abs(9) = 3 |

**3. Các hàm được sử dụng phổ biến trong Javascript**

**a. Hàm isNaN**

Hàm isNaN() kiểm tra giá trị. Nếu không phải số trả về true, ngược lại thì nếu là số thì trả về là false

**Code:**

|  |
| --- |
| var data1 = 10, data2 = "Vietpro Academy";  document.write(isNaN(data1)); // false  document.write(isNaN(data2)); // true |

**b. Hàm eval**

Hàm eval() biến chuỗi thành biểu thức tính toán được hoặc biến chuỗi thành lệnh thi hành được như là mã lệnh của JS

**Code:**

|  |
| --- |
| var str = "10\*2 + 30";  document.write(str); // 10\*2 + 30  document.write(eval(str)); // 50 |

**c. Hàm setTimeOut**

Hàm SetTimeOut() có tác dụng gọi một hàm sau thời gian định trước, và chỉ thực thi một lần duy nhất

**Code:**

|  |
| --- |
| function messages(){  alert("Vietpro Academy");  }  setTimeOut("messages()", 3000); |

**d. Hàm setInterval**

Hàm SetInterval() có tác dụng gọi một hàm sau những thời gian định trước, được thực thi lặp đi lặp lại trong suốt quá trình chạy file hiện hành

**Code:**

|  |
| --- |
| function messages(){  alert("Vietpro Academy");  }  setInterval("messages()", 1000); |

**B – THỰC HÀNH**

**I - BÀI TẬP THỰC HÀNH**

**Bài 1:** Áp dụng kiến thức về Mảng & Hàm để Xây dựng Hàm cho phép người dùng sử dụng để đếm số phần tử có Key chẵn và số phần tử có Key lẻ của một mảng bất kỳ

**Bài 2:** Áp dụng kiến thức về Hàm để Viết Ứng dụng sau:

* Chạy ứng dụng thì xuất hiện hộp thoại Prompt yêu cầu nhập vào Số thứ nhất (Số bất kỳ)
* Sau khi bấm OK thì xuất hiện hộp thoại Prompt tiếp theo yêu cầu nhập vào một trong 5 ký tự sau (+, -, x, :, %)
* Sau khi bấm OK thì xuất hiện hộp thoại Prompt tiếp theo yêu cầu nhập vào Số thứ hai (Số bất kỳ)
* Sau lần bấn OK này thì xuất hiện hộp thoại Alert thông báo kết quả của Số thứ nhất phép toán với Số thứ hai
* Chạy file bài giải để hiểu rõ chương trình

**II - BÀI TẬP VỀ NHÀ**

**Bài 3:** Viết ứng dụng tạo mảng tự động

* Với các Giá trị của các phần tử mảng do người dùng tự nhập vào từ hộp thoại Prompt.
* Người dùng muốn kết thúc việc nhập giá trị các phần tử mảng thì chỉ cần nhập giá trị của phần tử mảng bằng 0.
* Sau khi kết thúc sẽ thấy thông báo tất cả các giá trị của tất cả các phần tử trong mảng vừa được người dùng nhập vào
* Chạy file bài giải để hiểu rõ chương trình

**Bài 4:** Viết ứng dụng tự động tách Mảng Lẻ - Mảng Chẵn

* Yêu cầu người dùng nhập vào 10 Số Nguyên Dương bất kỳ, mỗi Số Nguyên Dương được nhập vào từ hộp thoại Prompt.
* Sau khi người dùng nhập đầy đủ và bấm OK thì hiển thị tất cả giá trị của các phần tử thuộc mảng Lẻ (Bao gồm chính các Số Nguyên Dương lẻ mà người dùng nhập vào ở trên) và tất cả giá trị của các phần tử thuộc mảng Chẵn (Bao gồm chính các Số Nguyên Dương chẵn mà người dùng nhập vào ở trên)
* Chạy file bài giải để hiểu rõ chương trình

**Bài 5:** Viết ứng dụng Random Image

* Hiển thị hình ảnh ngẫu nhiên, mà cứ mỗi lần người dùng tải lại trang web (F5) thì sẽ lại được hiển thị một bức ảnh khác.
* Chạy file bài giải để hiểu rõ chương trình

**Bài 6:** Viết ứng dụng tạo Bảng (Table) tự động

* Cho phép người dùng tạo Bảng và nhập Dữ liệu trực tiếp cho Bảng thông qua hộp thoại Prompt
* Yêu cầu nhập số lượng dòng (tr) từ Prompt
* Yêu cầu nhập số lượng cột (td) từ Prompt
* Yêu cầu nhập nội dung của từng cột từ Prompt
* Chạy file bài giải để hiểu rõ chương trình