Phần 1 :ES6 :

1. ES6 : những điều cần biết:
   1. Do hiệp hội các nhà sản xuất máy tính Châu Âu đề xuất làm tiêu chuẩn của ngôn ngữ JavaScript
   2. Là bản phát triển của JS
2. Variable – template String – Spread – Destructing
   1. Var : tạo ra 1 biến có phạm vi truy cập xuyên suốt function chứa nó
   2. Let : tạo ra 1 biến chỉ có thể truy cập trong block-scope bao quanh nó
   3. Ngoài ra khi ở global scope : tức là k nằm trong 1 function nào cả thì var tạo cho global object this còn let thì k
   4. Trong ES6 : dùng let trong mọi trường hợp
   5. Let : là khai báo 1 biến tạm thời trong 1 block scope nào đó
   6. Const: tao ra hằng số
3. Template String : xuất ra 1 chuỗi
   1. Cú pháp : `${}`
4. For: duyệt mảng có 3 cách
   1. For
   2. For in
   3. For of
5. Đối với duyệt đối tượng
   1. thì dùng for in cũng phải truy cập y hệt như arr
6. Tìm hiểu về Spread :
   1. Bản chất của nó cũng là 1 toán tử (operator)
   2. Đưa dấu … trước mảng mà chúng ta muốn chèn
7. Destructuring : phép gán
   1. Đối với arr thì dùng để nối id vs giá trị với nhau
   2. Đối với đối tượng thì nếu ghi sai thuộc tính thì sẽ báo lỗi còn nếu k có thì để giá trị mặc định cũng ok
8. Function
   1. Default params : giá trị mặc định khi viết hàm
   2. Rest params : giá trị truyền vào tham số có quá nhiều ví dụ như truyền a ,b ,c => sinh ra rest params để truyền đưa vào dấu …values = > values chính là 1 arr
9. Arrow Function:
   1. Viết 1 function như bình thường
   2. Có 1 cách viết function khác gọi là arrow function : bỏ luôn từ khóa function và thêm kí hiệu => sau tham số truyền vào
   3. Nếu mà ở dưới chỉ có 1 câu lệnh thì bỏ cả function và phần {}, bỏ được cả return nữa
   4. Nếu function mà chỉ truyền vào 1 tham số thì có thể bỏ luôn phần ()
   5. Nếu funtion k có tham số thì kệ
   6. Method map(). Tạo 1 mảng mới dựa trên kết quả của hàm đã cho
   7. This : đại diện cả cho đối tượng đó
10. Enhanced :
    1. Properti value shorthand : có thể bỏ các giá trị key đi
    2. Method propertive : sử dụng nhiều method trong 1 đối tượng
    3. Computed property names: thay đổi các thuộc tính: sẽ là [] bên trong là tên thuộc tính
11. Kế thừa :
    1. Extends : để kế thừa
    2. ở class con viết lại constructor thì dùng super() cho ngắn gọn

Phần 2 : TypeScript

1. TypeScript và những điều cần biết :
   1. Ngôn ngữ mã nguồn mở được phát triển bởi Microsoft
   2. Mô hình hướng đối tượng
   3. Phát triển ứng dụng chạy phía client hoặc phía server (Node.js)
   4. Typescript sinh ra mã javascript hoạt động tốt trên các môi trường hỗ trợ js
   5. Nguồn tài liệu : typescriptlang.org
2. Cấu trúc thư mục
   1. Giống hệt bootstrap
3. Cài đặt : đọc ở doccument
   1. Tải node.js về rồi sử dụng câu lệnh : npm install -g typescript
   2. Câu lệnh biên dịch : tsc greeter.ts
   3. Tuy nhiên cách biên dịch như trên tốn nhiều thời gian vì thế k dùng cách này mà sử dụng tập tin tsconfig để biên dịch cho nhanh : doccument : jsconfig:
      1. outDir : vị trí file biên dịch ra js
      2. watch : true : bật chế độ theo dõi
      3. exclude : chứa mảng những foder mà k muốn biên dịch
      4. target : biên dịch theo chuẩn nào thường là ES6
      5. removeComments : xóa đi comment
   4. sử dụng câu lệnh tsc để biên dịch
4. variable và Data Type
   1. array có 2 kiểu viết :
      1. let arrNumber : Array<number> = [1,5,8];
      2. let arrString : string[] = ["java", "android", "es6"];
   2. Tuple : giúp ràng buộc kiểu dữ liệu
   3. Enum:
   4. Any : kiểu dữ liệu bất kỳ
   5. Void : thường sử dụng quyết định trong kiểu trả về của 1 function
   6. Type assertions : chuyển đổi qua lại kiểu dữ liệu với nhau: <string>buến
   7. Interface object
   8. Interface Array: quy định kiểu dữ liệu của phần tử
5. Function
   1. Sự trả về của hàm : viết hàm xong cần ghi rõ kiểu trả về
   2. Truyền tham số vào trong hàm
   3. Overload:
6. OOP