**Javascript ES6 feature review**

|  |
| --- |
| Variable   1. define let vs const, let a = 123, const a = 6 |
| Function   1. arrow function ( ) => {}, add = (a,b) => {return a+b}   Good:   * do not use function keyword again * ‘this’ keyword clear   String function API  Array function API |
| Operator  ... => array,object addition  l = [1,2,3], new\_l = [...l,456] => [1,2,3,4,5,6]  Obj = {name:’Sam’}, newObj = {...Obj, age:10} => {name:’Sam’, age:10}  ... => make other data type as array  (1,2,3) => ...arg = > [1,2,3]  Rest = (a,b,...c)=>{print(a,b,c)}  Arg = [1,2,3,4,5,6]  Rest(arg) => a=1, b= 2 c = [3,4,5,6]  !only one ... in the expression |
| Library   1. import, export   Child.js  Only export one object => export default objName  Const person = {name: ‘max’, age:12} export default person  Export multiple things  Export const fun1() => {}  Export const var1 = 3  Parent.js  Import person from ./Child.js => use person object in Parent.js  Import {var1} from ./Child.js  Import {fun1} from ./Child.js  Import \* as child from ./Child.js => get all thing from child.js |
| Classes => property + method = blueprint of an object  Class Person {  Constructor(){  This.name = ‘Sam’  }  Speak = () =>{console.log(‘hi’)}  }  Inheritance of class  Class A extends B {object crated by A have B’s property and methods}  ! If both class A and B have the constructor  We should add super() in child( here is B) class to make sure B can access A’s property  ES 7 define property and method  Class A {  ! Do not use constructor  name = ‘Sam’  Speak = () => {...}} |