**JS basic**

* In HTML, JS must inserted between <script> and </script> tags
* JS can be placed in <body>, <head> or external files
* innerHTML: write into a HTML element
* document.write(): writing into the HTML output (never use it after the document has finished loading)
* window.alert(): writing into an alert box
* console.log(): writing into the browser console
* JS statements are: Values, Operators, Expressions, Keywords, and Comments
* Semicolons ; separate JS statements
* var : declare variables

Arrow function

Before:

hello = function(){

return "Hello World!";

}

After:

hello = () =>"Hello World!";

With Parameters:

hello = (val) => "Hello" + val;

* This inside arrow function is different compare to regular functions
* In regular function, this represented the object that called the function
* In arrow function, this always represents the object that defined the arrow function
* After arrow function defined, "this" keyword represents the object that owns the arrow function, no matter who call the function (arrow function里有object那么指向那个object，不然指向object window)

Use Strict

* "use strict" defines that JS should be executed in "strict mode"
* Declared at the beginning of a script, all code in the script will execute in strict mode
* strict mode cannot use not declared variable
* cannot use not declared object
* deleting a variable or object is not allowed
* deleting a function is not allowed
* duplicating parameter name is not allowed
* Octal numeric literals are not allowed
* Octal escape character are not allowed
* writing to a read-only property is not allowed
* writing to a get-only property is not allowed
* Deleting an undeletable property is not allowed
* word "eval" and "arguments" cannot be used as a variable
* with statement is not allowed
* eval() is not allowed to create variables in the scope from which is was called (eval() function evaluates JS code represented as a string)
* this keyword in strict mode refers to the object that called the function

Length syntax for JS

const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];

const result = words.filter(word => word.length > 6);

console.log(result);

// expected output: Array ["exuberant", "destruction", "present"]

ES 5 Features:

* The "use strict" Directive
* String.trim() : remove whitespace from both ends of a string
* Array.isArray(): return boolean which determines whether the pass value is an Array
* Array.forEach() : execute a provided function once for each array element
* Array.map(): creates a new array populated with the results of calling a provided function on every element in the calling array
* Array.filter(): creates a new array with all elements that pass the test implmented
* Array.reduce(): executes a reducer function(that you provide) on each element of the array, result in a single output value
* Array.reduceRight(): applies a function against an accumulator and each value of the array(from right to left) to reduce it to a single value
* Array.every(): test whether all elements in the typed array pass the test implemented by the provided function (全通过才return true)
* Array.some():test whether some elements in the typed array pass the test implemented by the provided function (只要有一个通过就ruturn true)
* Array.indexOf(): return the first index at which a given element can be found in the array, -1 is not founded
* Array.lastIndexOf():return the last index at which a given element can be found in the array, -1 is not founded, searching backwards
* JSON.parse(): 解析 JSON string, constructing the JS value or object described by the string (not allow trailing commas and single quotes)
* JSON.stringify(): converts a JS object or value to JSON string
* Date.now(): returns the number of milliseconds elapsed since Jan 1st, 1970 00:00:00 UTC
* Property Getters and Setters: let you define object methods with a syntax that looks like getting or setting property
* New Object Property Methods: Object.defineProperty(), let you define an object property and/or change a property's value and/or metadata

ES6 Features:

* JavaScript let: let statement allows you declare a variable with block scope
* JavaScript const: allows you to declare a constant with block scope (similar to let statement but cannot change)
* JavaScript Arrow Functions
* JavaScript Classes: replace function keyword into class keyword. Properties are assigned inside a constructor() method (default setter)
* Default parameter values: ES6 allows function parameters to have default values
* Array.find(): return the value of the first array element that passes a test function
* Array.findIndex(): return the index of the first array element that passes a test function
* New Number Properties: EPSILON(ε), MIN\_SAFE\_INTEGER(-9007199254740991), MAX\_SAFE\_INTEGER(9007199254740991)
* Number.isInteger(): return true if the argument is an integer
* Number.isSafeInterger():return true if the argument is an safe integer ( safe integer: -(253 - 1) to +(253 - 1) range)
* isFunite(): returns false if the argument is infinity or not a number
* isNaN(): returns true if the argument is not a number
* Exponentiation Operator \*\*: same as power function ie: x\*\*2 = Math.pow(x,2)