vue详解--- es5和es6的基本语法

1. es6的基本语法

let：

特点：

1.a是局部作用域的 function xx(){let a = 'xxoo';} if(){let a = 'ss';}

2.不存在变量提升

3.不能重复声明（var可以重复声明） ,var声明的不能用let再次声明,反之也是

4 let声明的全局变量不从属于window对象，var声明的全局变量从属于window对象。

关于第4个特点的简单说明：

ES5声明变量只有两种方式：var和function。

ES6有let、const、import、class再加上ES5的var、function共有六种声明变量的方式。

还需要了解顶层对象：浏览器环境中顶层对象是window.

ES5中，顶层对象的属性等价于全局变量。var a = 10; window.a

ES6中，有所改变：var、function声明的全局变量，依然是顶层对象的属性；let、const、class声明的全局变量不属于顶层对象的属性，也就是说ES6开始，全局变量和顶层对象的属性开始分离、脱钩，目的是以防声明的全局变量将window对象的属性造成污染，因为window对象是顶层对象，它里面的属性是各个js程序中都可以使用的，不利于模块化开发，并且window对象有很多的自有属性，也为了防止对window的自由属性的污染，所以在ES6中将顶层对象和全局变量进行隔离。

举例：

```
var a = 1;
console.info(window.a);  // 2

let b = 2;
console.info(window.b); //undefined

// var a;  -- undefined
console.log(a); -- undefined
var a = 10;

var b = 'xx';
console.log(c);  -- 报错
let c = 'xx';
```

const： 特点： 1.局部作用域 2.不存在变量提升 3.不能重复声明 4.一般声明不可变的量

```
const pi = 3.1415926;
//pi = 'xx' -- 报错
```

模板字符串：tab键上面的反引号，${变量名}来插入值　类似于python的三引号"""adfadsf"""，可以写多行文本

另外还可以通过它来完成字符串格式化

示例：

```
let bb = 'jj';
var ss = `你好${bb}`;
console.log(ss); -- '你好jj'
```

2. es5和es6的函数对比

```
//ES5写法
function add(x){
    return x
}
add(5);
//匿名函数
var add = function (x) {
    return x
};
add(5);
//ES6的匿名函数
let add = function (x) {
    return x
};
add(5);
//ES6的箭头函数,就是上面方法的简写形式
let add = x => {
    console.log(x);
    return x
};
console.log(add(20));
//更简单的写法，但不是很易阅读
let add = x => x;
console.log(add(5));
```

多个参数的时候必须加括号，函数返回值还是只能有一个，没有参数的，必须写一个()

```
let add = (x,y) => x+y;
```

3. 自定义对象中封装函数的写法

```
 //es5对象中封装函数的方法
var name = 'xx';
var person1 = {
    name:'xx',
    age:18,
```

```
    f1:function () {  //在自定义的对象中放函数的方法
        console.log(this);
        console.log(this.name)
    }
};

//es5对象中封装函数的方法
var name = 'xx';
var person1 = {
    name:'xx',
    age:18,
    f1:function () {  //在自定义的对象中放函数的方法
        console.log(this);
        let aa = {
        aaa:'xx',
        af:()=>{console.log(this)}
        }
aa.af()
    }
};
//<h1 id='d1'>xxx</h1>
//document.getElementById('d1').onclick = function(){this.innerText;};
person1.f1();  //通过自定对象来使用函数

//ES6中自定义对象中来封装箭头函数的写法
let username = 'xx'; //-- window.username
let person2 = {
    name:'xx',
    age:18,
    //f1:function(){};
    f1: () => {  //在自定义的对象中放函数的方法
        console.log(this); //this指向的不再是当前的对象了，而是指向了person的父级对象(称为上下文)，而此时的父级对象是我们的window对象，Window {postMessage: ƒ, blur: ƒ, focus: ƒ, close: ƒ, frames: Window, ...}
        console.log(window);//还记得window对象吗，全局浏览器对象，打印结果和上面一样：Window {postMessage: ƒ, blur: ƒ, focus: ƒ, close: ƒ, frames: Window, ...}
        console.log(this.username)  //'子俊'
    }
};
person2.f1(); //通过自定对象来使用函数
person2 -- window.person2
//而我们使用this的时候，希望this是person对象，而不是window对象，所以还有下面这种写法
let person3 = {
    name:'xx',
    age:18,
    //f1:function(){};
```

```
    //f1(){}
    f1(){  //相当于f1:function(){},只是一种简写方式,称为对象的单体模式写法，写起来也简单，vue里面会看用到这种方法
        console.log(this);//this指向的是当前的对象,{name: "超", age: 18, f1: ƒ}
        console.log(this.name)  //'超'
    }
};
person3.f1()



let name2 = 'xx';
let person2 = {
    name2:'xx',
    age:18,
    f1:() => {
        console.log(this);
        console.log(this.name2)
    }
};
person2.f1();
```

4. es5和es6的类写法对比（了解）

```
<script>

    //es5写类的方式
    function Person(name,age) {
        //封装属性
        this.name = name;  //this--Python的self
        this.age = age;
    }
    //封装方法，原型链
    Person.prototype.f1 = function () {
        console.log(this.name);//this指的是Person对象, 结果：'超'
    };
    //封装方法，箭头函数的形式写匿名函数
    Person.prototype.f2 = ()=>{
        console.log(this); //Window {postMessage: ƒ, blur: ƒ, focus: ƒ, close: ƒ, frames: Window, …}  this指向的是window对象
    };

    var p1 = new Person('xx',18);
    p1.age

    p1.f1();
```

```
    p1.f2();
    //其实在es5我们将js的基本语法的时候，没有将类的继承，但是也是可以继承的，还记得吗，那么你想，继承之后，我们是不是可以通过子类实例化的对象调用父类的方法啊，当然是可以的，知道一下就行了，我们下面来看看es6里面的类怎么写
    class Person2{
        constructor(name,age){ //对象里面的单体模式，记得上面将函数的时候的单体模式吗，这个方法类似于python的__init__()构造方法，写参数的时候也可以写关键字参数 constructor(name='超2',age=18)
            //封装属性
            this.name = name;
            this.age = age;
        } //注意这里不能写逗号

        showname(){  //封装方法
            console.log(this.name);
        } //不能写逗号
        showage(){
            console.log(this.age);
        }
    }
    let p2 = new Person2('x2',18);
    p2.showname()  //调用方法  'x2'
    //es6的类也是可以继承的，这里咱们就不做细讲了，将来你需要的时候，就去学一下吧，哈哈，我记得是用的extends和super

</script>
```