

前端Start

使用vscode进行代码编写

1.下载对应的四个插件

vetur/vue-helper/Live Server/Chinese

2.创建工作区

- 现在本地创建文件夹
- 使用vscode打开创建的空文件
- 将文件夹保存成工作区(文件--->工作区另存为--->同名文件夹下保存同名.code-workspace文件)

3.tips

换行是ctrl+enter, 通过ctrl+K&ctrl+s调出控制面板更改成shift+enter

4.live server

可以用于模拟tomcat的ip:port进行测试, 右键相关的html文件--->run with live server

es6:ECMAScript 6

1.let

var声明的变量没有局部作用域

let声明的变量有局部作用域



```
1 <script>
2   //es如何定义变量, 定义变量特点
3   //js定义    var a = "aa"
4   //es6 写法里 定义变量使用let关键字
5   {
6     var a = "aa"
7     let b = "bb"
8   }
9
10  console.log(a)
11  console.log(b)
12
13 </script>
```

aa

Uncaught ReferenceError: b is not defined
at es6demo.html:11

var可以声明多次

let只能声明一次

```
1  <script>
2    var a = 1
3    var a = 2
4
5    let m = 10
6    let m = 20 // Uncaught SyntaxError: Identifier 'm' has already been declared
7
8    console.log(a);
9    console.log(b);
0  </script>
```

2.const

```
<script>
  //1.声明后不可改变
  const PI = "3.1415926"
  PI = 3 // TypeError: Assignment to constant variable.

  //2.一旦声明必须初始化，否则会报错
  const My_Age //SyntaxError: Missing initializer in const declaration.
</script>
```

3.解构赋值

解构赋值是对赋值运算符的扩展

他是一种针对数组或者对象进行模式匹配，然后对其中的变量进行赋值。

在代码书写上简洁且易读，语义更加清晰明了，也方便了复杂对象当中数据字段的读取

```
<script>
  //1.数组解构
  //传统
  let a = 1, b = 2, c = 3
  console.log(a, b, c);
  //es6
  let [x, y, z] = [1, 2, 3]
  console.log(x, y, z)

  //对象解构
  let user = {name: 'Helen', age: 18}
  //传统
  let name1 = user.name
  let age1 = user.age
  //es6
  let {name, age} = user //注意:结构的变量必须是user中的属性
  console.log(name, age)
</script>
```

4.模板字符串

模板字符串相当于加强版的字符串，用反引号`，除了作为普通字符串，还可以用来定义多行字符串，还可以在字符串中加入变量和表达式。

```
<script>
  //1.多行字符串
```

```

let string1 = `Hey,
can you stop angry now?`
console.log(string1);
// Hey,
// can you stop angry now?

//2.字符串插入变量和表达式。变量名写在${}中，${}中可以放入javascript表达式
let name = "Mike"
let age = 27
let info = `My Name is ${name}, I am ${age+1} years old next year.`
console.log(info)
//My name is Mike, I am 28 years old next year

//3.字符串中调用函数
function f(){
    return "have fun!"
}
let string2 = `Game start,${f()}`
console.log(string2)    // Game start,have fun!
</script>

```

5.声明对象简写

```

<script>
    const age = 12
    const name = "amy"

    //传统
    const person1 = { age: age, name: name }
    console.log(person1);

    //ES6
    const person2 = { age, name }
    console.log(person2);
</script>

```

6.定义方法简写

```

<script>
    //传统
    const person1 = {
        sayHi:function(){
            console.log("Hi");
        }
    }

    person1.sayHi()

    //ES6
    const person2 = {
        sayHi(){
            console.log("Hi");
        }
    }

    person2.sayHi()
</script>

```

7.对象拓展运算符

拓展运算符 (...) 用于取出参数对象所有可遍历属性然后拷贝到当前对象。

```
<script>
  //1.拷贝对象
  let person1 = { name: "Amy", age: 15 }
  let someone = { ...person1 }
  console.log(someone);

  //2.合并对象
  let age = { age: 15 }
  let name = { name: "Amy" }
  let person2 = { ...age, ...name }
  console.log(person2);

</script>
```

8.箭头函数

箭头函数提供了一种更加简洁的函数书写方式。基本语法是：

参数 => 函数体

```
<script>
  //传统
  var f1 = function (a) {
    return a
  }
  console.log(f1(1));

  //ES6
  var f2 = a => a
  console.log(f2(1))

  //当箭头函数没有参数或者有多个参数，需要用(括起来)
  //当箭头函数函数体有多行语句，用{}包裹起来，表示代码块
  //当只有一行语句，并且需要返回结果时，可以省略{}，结果会自动返回
  var f3 = (a, b) => {
    let result = a + b
    return result
  }

  console.log(f3(6, 2));    // 8

  //前面代码相当于：
  var f4 = (a, b) => a + b
</script>
```