

TypeScript 联合类型

联合类型 (Union Types) 可以通过管道(`|`)将变量设置多种类型，赋值时可以根据设置的类型来赋值。

注意：只能赋值指定的类型，如果赋值其它类型就会报错。

创建元组的语法格式如下：

```
Type1|Type2|Type3
```

实例

声明一个联合类型：

TypeScript

```
var val:string|number
val = 12
console.log("数字为 "+ val)
val = "Runoob"
console.log("字符串为 " + val)
```

编译以上代码，得到以下 JavaScript 代码：

JavaScript

```
var val;
val = 12;
console.log("数字为 " + val);
val = "Runoob";
console.log("字符串为 " + val);
```

输出结果为：

```
数字为 12
字符串为 Runoob
```

如果赋值其它类型就会报错：

```
var val:string|number
val = true
```

也可以将联合类型作为函数参数使用：

TypeScript

```
function disp(name:string|string[]) {
  if(typeof name == "string") {
    console.log(name)
  }
}
```

```
} else {  
  var i;  
  for(i = 0; i < name.length; i++) {  
    console.log(name[i])  
  }  
}  
}  
}  
disp("Runoob")  
console.log("输出数组....")  
disp(["Runoob", "Google", "Taobao", "Facebook"])
```

编译以上代码，得到以下 JavaScript 代码：

JavaScript

```
function disp(name) {  
  if (typeof name == "string") {  
    console.log(name);  
  }  
  else {  
    var i;  
    for (i = 0; i < name.length; i++) {  
      console.log(name[i]);  
    }  
  }  
}  
disp("Runoob");  
console.log("输出数组....");  
disp(["Runoob", "Google", "Taobao", "Facebook"]);
```

输出结果为：

```
Runoob  
输出数组....  
Runoob  
Google  
Taobao  
Facebook
```

联合类型数组

我们也可以将数组声明为联合类型：

TypeScript

```
var arr: number[] | string[];  
var i: number;  
arr = [1, 2, 4]  
console.log("**数字数组**")  
for(i = 0; i < arr.length; i++) {  
  console.log(arr[i])  
}
```

```
arr = ["Runoob", "Google", "Taobao"]
console.log("**字符串数字**")
for(i = 0; i < arr.length; i++) {
  console.log(arr[i])
}
```

编译以上代码，得到以下 JavaScript 代码：

JavaScript

```
var arr;
var i;
arr = [1, 2, 4];
console.log("**数字数组**");
for (i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}
arr = ["Runoob", "Google", "Taobao"];
console.log("**字符串数字**");
for (i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}
```

输出结果为：

```
**数字数组**
1
2
4
**字符串数字**
Runoob
Google
Taobao
```

[← TypeScript 元组](#)

[TypeScript 接口 →](#)

[✎ 点我分享笔记](#)

