

08数据类型转换

使用**表单**、**prompt** 获取过来的数据默认是**字符串类型**的，此时就不能直接简单的进行加法运算，而需要转换变量的数据类型。通俗来说，**就是把一种数据类型的变量转换成另外一种数据类型**。

我们通常会实现3种方式的转换：

- 转换为字符串类型
- 转换为数字型
- 转换为布尔型

转换为字符串型

方式	说明	案例
toString()	转成字符串	var num = 1; alert(num.toString());
String()强制转换	转成字符串	var num = 1; alert(String(num));
加号拼接字符串	和字符串拼接的结果都是字符串	var num =1; alert(num+"我是字符串");

```
1      <script>
2          var a = 123;
3          console.log(a.toString());
4          var b = 456;
5          console.log(String(b));
6          var c = 789;
7          console.log('我是一个字符串'+c);
8      </script>
```

- **toString()** 和 **String()** 使用方式不一样
- 三种转换方式，我们更喜欢用第三种**加号拼接字符串转换**方式，这一方式也称为**隐士转换**

转换为数字型

parseInt(string)函数

将**string类型**转成 **整数数值型**

```

1      <script>
2          //parseInt(string)函数,将string类型转成整数数值型
3          var a = '字符串';
4          console.log(parseInt(a));    //NaN
5          var b = '123123';
6          console.log(parseInt(b));    //123123,数值类型
7          var c = '3.1415926';
8          console.log(parseInt(c));    //3,因为parseInt是取整
9          console.log(parseInt('120px')); //120,会去掉单位值
10         console.log(parseInt('px120px')); //NaN
11     </script>

```

parseFloat(string)函数

将string类型转成浮点数数值型

```

1      <script>
2          // parseFloat(string)函数,将string类型转成浮点数数值型
3          var a = '3.141592612312312312312312312331212312321334';
4          //3.141592612312312
5          console.log(parseFloat(a));
6          var b = '0.1111111';
7          console.log(parseFloat(b));
8          //0.1111111
9      </script>

```

Number()强制转换函数

将string类型转换为数值型

```

1      <script>
2          // Number()强制转换函数,将string类型转换为数值型
3          var a = '123';
4          console.log(Number(a)); //123
5      </script>

```

js 隐式转换(- * /)

利用算术运算隐式转换为数值型

```

1      <script>
2          // 字符串通过算术运算符进行隐式转换
3          var a = '12'-0;
4          var b = '10' * 2;
5          var c = '10' /5;
6          var d = '100' - '80'; //20,数值类型
7          console.log(a); //12
8          console.log(b); //20
9          console.log(c); //2
10     </script>

```

方式	说明	案例
parseInt(string)函数	将string类型转成整数数值型	parseInt('78')
parseFloat(string)函数	将string类型转成浮点数数值型	parseFloat('78.21')
Number()强制转换函数	将string类型转换为数值型	Number('12')
js 隐式转换(- * /)	利用算术运算隐式转换为数值型	'12'-0

- 1.注意 parseInt 和 parseFloat ， 这两个是重点。
- 2.隐式转换是我们在进行算数运算的时候，JS自动转换了数据类型。

案例一：计算年龄案例

流程：

- 弹出一个输入框，让用户输入出生年份
- 用现在的年份减去用户输入的年份，就是现在的年龄
- 弹出警示框，把计算的结果输出

```

1  <script>
2      //弹出一个输入框，让用户输入出生年份
3      var age = prompt('你出生的年份');
4      //用现在的年份减去用户输入的年份，就是现在的年龄
5      var year = 2021 - age;
6      //弹出警示框，把计算的结果输出
7      alert(year);
8  </script>

```

案例二：简易加法处理

计算两个数的值，用户输入第一个值后，继续弹出第二个输入框并输入第二个值，最后通过弹窗显示出两个值相加的结果。

流程：

- 弹出第一个输入框，输入第一个值
- 弹出第二个输入框，输入第二个值
- 把这两个值相加
- 弹出警示框，并在警示框输出结果。

```

1  <script>
2      var a = prompt('输入的第一个值');
3      var b = prompt('输入的第二个值');
4      //因为字符串算术运算符中没有设置加号，所以无法通过隐式转换。所以需要进行函数进行转换。
5      var c = parseFloat(a)+parseFloat(b);
6      console.log(c);    //2
7  </script>

```

转换为布尔型

方法	说明	案例
Boolean()函数	其他类型转成布尔值	Boolean('true');

- 代表空，否定的值会被转换为false，如 '' , 0, NaN , null , undefined
- 其余的值都会被转换为true

```
1 console.log(Boolean('')); //false
2 console.log(Boolean(0));  //false
3 console.log(Boolean(NaN)); //false
4 console.log(Boolean(null)); //false
5 console.log(Boolean(undefined)); //false
6 console.log(Boolean('小白')); //true
7 console.log(Boolean(12));    //true
```