/\*

16 \* 将其他的数据类型转换为String

17 \* 方式一：

18 \* - 调用被转换数据类型的toString()方法

19 \* - 该方法不会影响到原变量，它会将转换的结果返回

20 \* - 但是注意：null和undefined这两个值没有toString()方法，如果调用他们的方法，会报错

21 \*

22 \* 方式二：

23 \* - 调用String()函数，并将被转换的数据作为参数传递给函数

24 \* - 使用String()函数做强制类型转换时，

25 \* 对于Number和Boolean实际上就是调用的toString()方法

26 \* 但是对于null和undefined，就不会调用toString()方法

27 \* 它会将 null 直接转换为 "null"

28 \* 将 undefined 直接转换为 "undefined"

29 \*

30 \*/

31

32 //1.调用被转换数据类型的toString()方法

33 var a = 123;

34 a = a.toString();

35 console.log(typeof a); //string

36

37

38 a = true;

39 a = a.toString();

40 console.log(typeof a); //string

41

42 a = null;

43 a = a.toString();

44 console.log(typeof a); //报错，Uncaught TypeError: Cannot read property 'toString' of null

45

46 a = undefined;

47 a = a.toString();

48 console.log(typeof a); //报错，Uncaught TypeError: Cannot read property 'toString' of undefined

49

50 //---------------------------------------------------------------------------------------------

51

52 //2.调用String()函数，并将被转换的数据作为参数传递给函数

53 a = 123;

54 a = String(a);

55 console.log(typeof a); //string

56

57 a = null;

58 a = String(a);

59 console.log(typeof a); //string

60

61 a = undefined;

62 a = String(a);

63 console.log(typeof a); //string

64

65 //3.我用Java中的方法，发现也是可以的

66 var b = 123;

67 b = "" + b;

68 console.log(typeof b); //string