**JAVASCRIPT WEEK2**

**String Conversion**

String conversion happens when we need the string form of a value.

For example, alert(value) does it to show the value.

We can also call the String(value) function to convert a value to a string:

let value = true;

alert(typeof value); // boolean

value = String(value); // now value is a string "true"

alert(typeof value); // string

String conversion is mostly obvious. A false becomes "false", null becomes "null", etc.

**Numeric Conversion**

Numeric conversion happens in mathematical functions and expressions automatically.

For example, when division / is applied to non-numbers:



We can use the Number(value) function to explicitly convert a value to a number:

let str = "123";

alert(typeof str); //  string

let num = Number(str); // becomes a number 123

alert(typeof num); // number

Explicit conversion is usually required when we read a value from a string-based source like a text form but expect a number to be entered.

If the string is not a valid number, the result of such a conversion is NaN. For instance:

let age = Number("an arbitrary string instead of a number");

alert(age); // NaN, conversion failed

Numeric conversion rules:

|  |  |
| --- | --- |
| Value | Becomes… |
| undefined | NaN |
| null | 0 |
| true and false | 1 and 0 |
| string | Whitespaces from the start and end are removed. If the remaining string is empty, the result is 0. Otherwise, the number is “read” from the string. An error gives NaN. |

**Examples:**

alert( Number("123") ); // 123

alert( Number("123z") ); // NaN (error reading a number at "z")

alert( Number(true) );  // 1

alert( Number(false) );  // 0

Please note that null and undefined behave differently here: null becomes zero while undefined becomes NaN.

Most mathematical operators also perform such conversion, we’ll see that in the next chapter.