Types & Coercion Cheatsheet

Type Conversion

It means that we convert the data type of a variable explicitly by ourselves, like using the Number method to convert a string to a number or using the + unary operator to do the same thing

Type Coercion

It means that the variable's data type is converted from one type to another implicitly by JavaScript itself.

This happens when we, for example, concatenate between a string and a number using the + binary operator ⇒ here the number is converted to a string implicitly.

It is important to mention that other arithmetic operators like -, *, / also performs type coercion when they encounter an operand that is not a number, they try to convert it to a number before evaluating the mathematical operations.

Another example for type coercion happens when we use == to compare between values, in this case, JavaScript converts different data types to number, then compare the values (except for null/undefined they are equal to themselves and to each other only without any type conversion, and they are not equal to anything else).

That's why it is recommended to use the === for the equality comparisons because it does not allow the type coercion to take place.

Common Coercion Cases are summarized here:

Example	Result	Comment
"10" + 2	102	Number → String
"10" - 2	8	String → Number
false + true	1	Boolean → Number
"10" > 2	true	String → Number
true > 0	true	Boolean → Number (true = 1)
if(0 || null || undefined || NaN || "") { //code to be executed }	The if statement will not be executed	Types → Boolean (Falsy Values)
3 > 2 > 1	false	3 > 2 → true, true > 1 → false (Boolean → Number)