Type Conversion & Coercion

→ Javascript Type Conversion:
→ When we convert one data type to another data type, this process is called type conversion.

→ In Javascript, there are two types of type conversion:

→ Explicit Conversion:
→ The type conversion that we manually do is known as explicit type conversion. In Javascript, explicit type conversion are done using the built-in methods like STRING(), NUMBER(), etc.

→ Number Conversion: To convert Boolean values or numeric strings to numbers, we use the NUMBER() i.e. an in-built method. NUMBER() method in case of empty strings and null values return 0. If a string is an invalid number like having an alphabet in a string, the

result will be NAN (Not a number).

→ ~~The~~ ~~for~~ Rules of the Numeric value:

VALUE	RETURN
→ 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. If the string contain any alphabet like 67a90, it will give NAN error.

Ex:

```
LET RES;
RES = NUMBER('100');    //100
RES = NUMBER(FALSE);    // 0
RES = NUMBER(' ');      // 0
RES = NUMBER('HELLO');  // NAN
RES = NUMBER(UNDEFINED); // NAN
```

→ Boolean Conversion: Boolean type conversion happens in logical operations.

It also follows the rules, but they are mostly obvious:
→ NAN, 0, undefined, NULL, "" are converting to FALSE.
→ everything else, including objects, to true

Ex:
```
REG = BOOLEAN(1);  // TRUE
REG = BOOLEAN(0);  // FALSE
REG = BOOLEAN("HELLO"); // TRUE
REG = BOOLEAN(""); // FALSE
```

→ String Conversion: The method STRING() is used to convert numbers to strings. It can be used on any type of numbers, literals, variables, or expressions. The method toSTRING() does the same.

Ex:
```
LET REG
LET A=90
REG = STRING(A) // returning a string from a
                   number variable A.
STRING(378009) // returning a string from a
                  number literal 378009
A.toSTRING()
(378009).toSTRING()
```

→ PARSEINT and PARSEFLOAT: The in-built

method PARSEINT() and PARSEFLOAT() return numbers from strings that start with numeric data.

Ex:

```
CONSOLE.LOG(PARSEINT('$100 DOLLARS')); //NAN
  "   (  "  ('+10.25990 Kg')); //10
  "   (  "  ('+10.25 Kg')); //10.25
  "   (  "  ('ABC')); //NAN
```

→ Implicit Conversion:

→ Sometimes Javascript automatically converts one data type to another. This type of conversion is known as implicit conversion.

→ Conversion To String: When we add a number into a string, Javascript converts the number to a string before concatenation.

Ex:

```
LET RES;
RES = '3' + 4;
CONSOLE.LOG(RES)    // "34"
RES = '9' + TRUE;
CONSOLE.LOG(RES);   // "9TRUE"
```

→ Conversion To Number: Numeric string used with operators like, +, -, /, * returning number type.

```
LET RES
RES = '4' - '4';
CONSOLE.LOG(RES); // 0
RES = '4' * 5;
CONSOLE.LOG(RES); // 20
```

→ Boolean Conversion to Number: If we use Boolean, true is considered as 1 and false is considered as 0.

```
LET RES;
RES = '5' - TRUE;
CONSOLE.LOG(RES); // 4
RES = 10 + FALSE;
CONSOLE.LOG(RES); // 10
```

→ In Javascript, NULL is considered as 0 when used with numbers. Arithmetic operation of undefined with number, boolean or null returns NAN.

```
RES = 4 + NULL; // 4
RES = 4 - UNDEFINED; // NAN
```

```
LET NUM = 43
CONSOLE.LOG(NUM.TOFIXED(2)) // to get to 2 decimal places.
```