

# Primitive Data Types and Operators

# Data Types

- *Every value*, or piece of data, has a specific **type**
- In JavaScript and TypeScript there are three *primitive data types*
  1. **number** (numerical data)
  2. **string** (textual data)
  3. **boolean** (logical data)
- **Primitive data types** are the *most basic* types built into a language
  - Later you will compose primitive data types together to form composite data types
- **A value's type informs its capabilities**
  - If you have two number values, then you can add them, for example.

# Numerical Type - number

- **number Literal** examples: **0**, **1**, **2**, **3.14**, **110.110**
- We tend to use numerical data in two ways:
  - As **integers**, which are useful for counting
  - As **decimals**, which are useful in simulations, modeling, and so on
- Lower-level programming languages like Java and C have specific types for integer (i.e. `int`) and decimal data (i.e. `double`). In TypeScript, it's just **number**.

# number Operators

	Name	Operator Symbol	Example
1	Exponentiation	**	2 ** 8 (is the same as 2 <sup>8</sup> )
2	Multiplication	*	10 * 3
	Division	/	100 / 5
	Remainder	%	18 % 5 (remainder of 18 divided by 5)
3	Addition	+	1 + 1
	Subtraction	-	111 - 1

- Complex expressions can be formed of multiple operators and parenthesis:  
**4 / ((1 + 1) \*\* 2) is 1**
- Rules of precedence determine the order each operator is evaluated  
**4 / 1 + 1 \*\* 2 is 5**
- The groupings show operators tiered from high-to-low precedence.
  - A sequence of operators of the same tier will be evaluated left-to-right.
  - For example, 8 / 4 \* 2 is 4.

# Textual Type - `string`

- `string` is short for "string of characters"
- Literal examples: `"abc"`, `"123"`, `"~() @#z2"`
- Useful for **all textual data**.

# Concatenation

- When you "add" two Strings together it is called **Concatenation**

`"Hello" + "World" // evaluates to "HelloWorld"`

- Concatenation means, simply, to "join" one string to another. When the program *concatenates* two strings, the result is a *single* String value.
- You can also concatenate numerical types to a String. This is very useful!

`"The answer is " + 42 // Evaluates to "The answer is 42"`

# Concatenation Gotcha

- Be careful **concatenating** two strings containing numbers together!
- What is:

"1" + "2"

- **It is "12"!**

# Logical Type - `boolean`

- Literal examples: `true`, `false`
- A `boolean` can only be **one of two possible values**, either `true` or `false`.
- An upcoming lesson will focus on `boolean` operators:
  - `not`
  - `and`
  - `or`