Type Conversions

This lesson will cover the concept of Type Conversion and its use.

An expression's evaluation can result in type conversions. These are called *implicit* conversions, as they happen automatically without the programmer's intervention. For example, using the `+` operator between a string and a number causes the concatenation of the two values into a string result.

```javascript
const f = 100;
// Show "Variable f contains the value 100"
console.log("Variable f contains the value " + f);
```

▷   🖫   ↤   ⛶

JavaScript is extremely tolerant in terms of type conversion. However, sometimes conversion isn't possible. If a value fails to convert into a number, you'll get the result `NaN` (Not a Number).

```javascript
const g = "five" * 2;
console.log(g); // Show NaN
```

▷   🖫   ↤   ⛶

Sometimes you'll wish to convert the value of another type. This is called *explicit* conversion. JavaScript has the `Number()` and `String()` commands that convert the value between the parenthesis to a number or a string.

```javascript
const h = "5";
console.log(h + 1); // Concatenation: show the string "51"
const i = Number("5");
console.log(i + 1); // Numerical addition: show the number 6
```

▷   🖫   ↤   ⛶