Converting to Strings

In this lesson, we'll learn how to convert various data types to strings.
Let's begin!

WE'LL COVER THE FOLLOWING

- `String()` rules for conversion

Almost all values (including reference values and primitive values) have a `toString()` method that can be used to convert the value to a string. If the value is a `Number`, you can pass an integer argument to `toString()`, which represents the base of the conversion. Its value must be between 2 and 36:

index.js

```
var num = 73;
console.log(num.toString(2));  // 1001001
console.log(num.toString(8));  // 111
console.log(num.toString(16)); // 49
var num1 = 17.25;
console.log(num1.toString(2)); // 10001.01
```

As you can see from this short code snippet, you can invoke `toString()` with a specific radix to convert floating-point numbers, too.

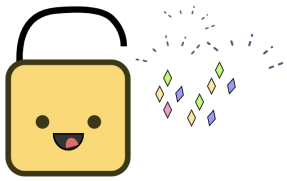
Another way to convert values to strings is to leverage the `String()` casting function. It works similarly to the other casting functions you have already met, such as `Boolean()` and `Number()`.

`String()` rules for conversion

`String()` uses these rules:

> - If the value is null, `"null"` is returned.
> - If the value is undefined, the casting results `"undefined"`.
> - Otherwise, if the value has a `toString()` method, it is called with no arguments, and its result is returned.

Achievement unlocked!



Congratulations! You've learned some really useful stuff about string conversion in JavaScript.

Great work! Give yourself a round of applause! :)

In the *next lesson*, we'll look at some string properties and some useful string functions.

Stay tuned!