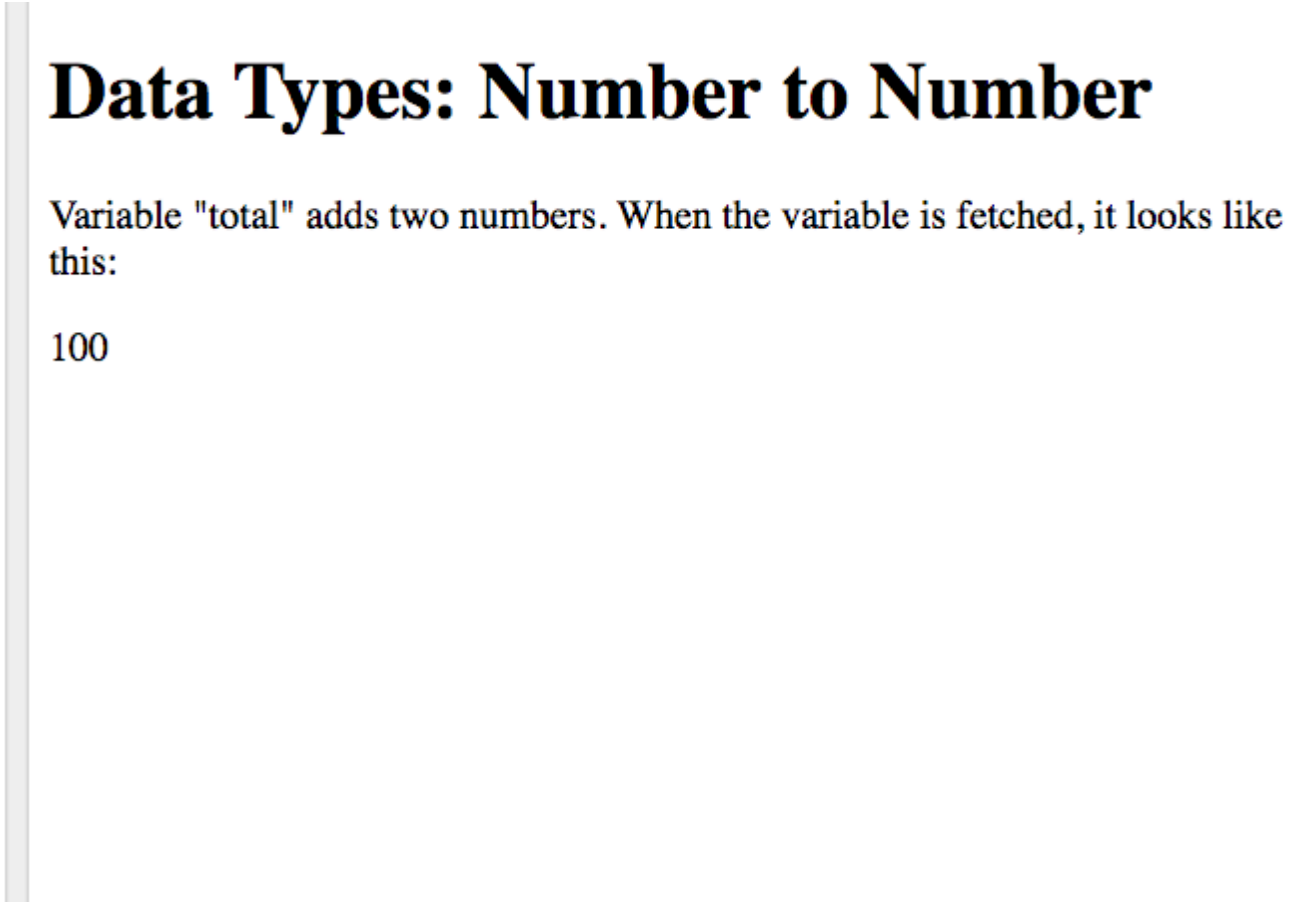
Introduction To Data Types

In JavaScript, data can be numeric. See how the variable in this code adds 60 and 40 to render as 100?

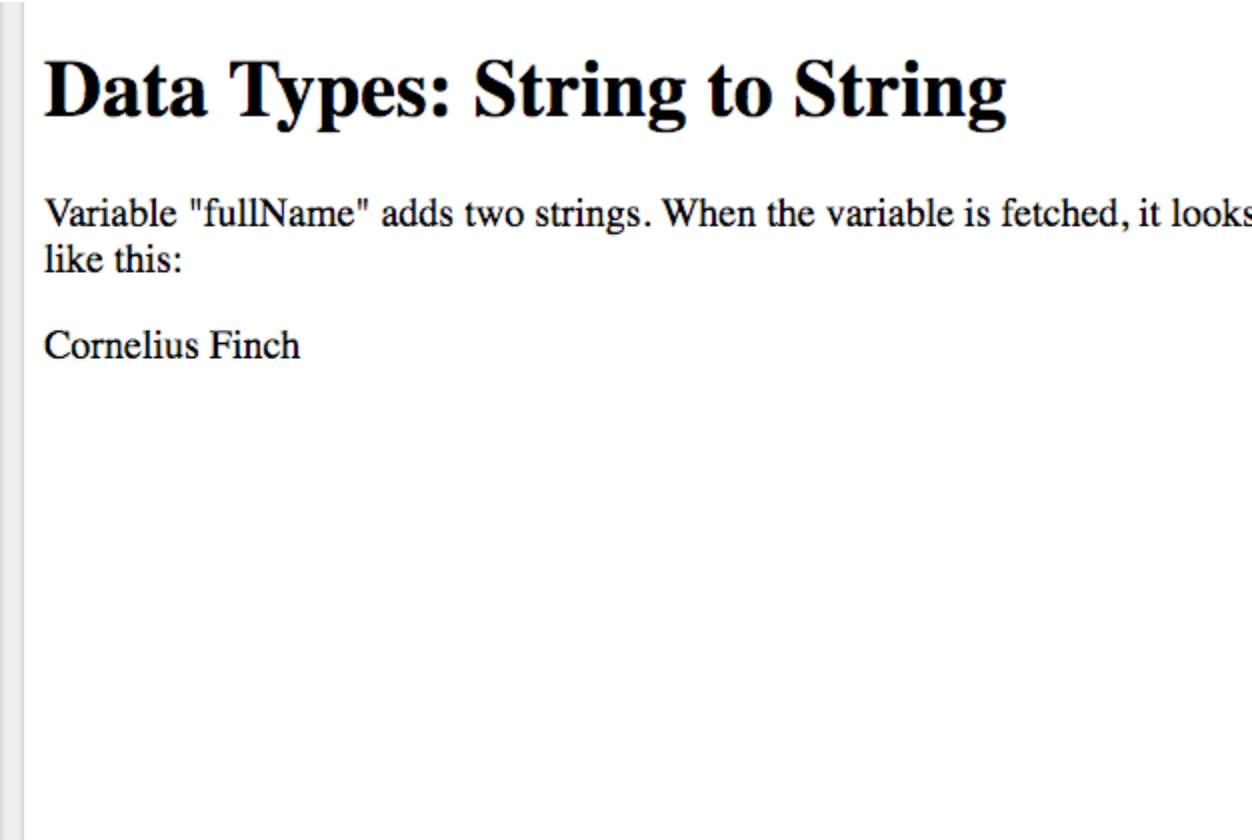
```
<!DOCTYPE html>
<html>
<body>

<h1>Data Types: Number to Number</h1>

<p>Variable "total" adds two numbers. When the variable
is fetched, it looks like this:</p>

<p id="twonumbers"></p>

<script>
var total = 60 + 40;
document.getElementById("twonumbers").innerHTML =
total;
</script>

</body>
</html>
```



Introduction To Data Types (Continued)

- Data can also be strings of characters. See how the variable in this code puts the two names together to render as a full name?

```
<!DOCTYPE html>
<html>
<body>

<h1>Data Types: String to String</h1>

<p>Variable "fullName" adds two strings. When the
variable is fetched, it looks like this:</p>

<p id="twostrings"></p>

<script>
var fullName = "Cornelius" + " Finch";
document.getElementById("twostrings").innerHTML =
fullName;
</script>
```

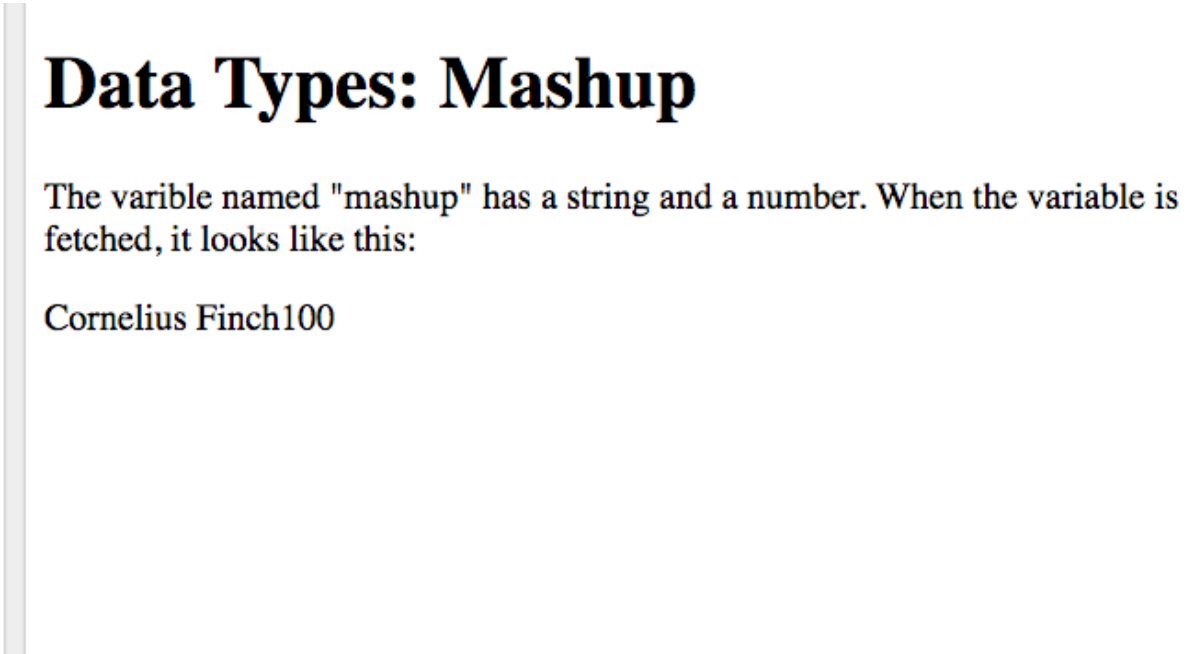


Introduction To Data Types (Continued)

- But, what if we mix up two types of data? What if we declare our variable to be the sum of a name and a number?

  Will our code break? Nah, JavaScript will just treat the number as a string.

```
<!DOCTYPE html>
<html>
<body>

<h1>Data Types: Mashup</h1>

<p>The varible named "mashup" has a string and a
number. When the variable is fetched, it looks like
this:</p>

<p id="mixed-bag"></p>

<script>
var mashup = "Cornelius Finch" + 100;
document.getElementById("mixed-bag").innerHTML =
mashup;
```



Data Types Defined

- In computer science and computer programming, a data type, or simply "type," is a classification that determines possible values for data that match its criteria, the operations that can be performed on these values, the meaning of the data, and the way values can be stored.

Data Types Across Languages

- Data types are really similar across different programming languages. So, once you're well-versed in how and when to use them in JavaScript, you'll be able to cruise into any language with some base knowledge.
- We'll elaborate on all of these types — except Booleans, for now.

1. Numeric	2. String	3. Boolean
Handles numbers	Consists of letters and/or other characters	Handles true or false values
Ex: 200.54 Ex: 893	Ex: 'GA@ga.co' Ex: "How are you user?"	Ex: true Ex: false
Used for tasks that involve counting or calculating	Used when working with any kind of text Written with single or	Used when there are two options for a value (i.e. yes/no, on/off, true/false)

Numbers vs. Strings

- Notice that in JavaScript, you don't need to declare the type of data you're using.
- For example, when you code:
- `var a = 13;`
- `var b = "thirteen";`
-
- JS will know that `var a` is a number and `var b` is a string because strings always have quotation marks. Using quotation marks is a way of communicating to the processor what type of data to expect.
- If you put quotations marks around a number, like so `"9"`, the processor will read it as a string. You should always be mindful of the type of data you use moving forward.
-

The "typeof" Command

- When we start writing more complex code, there may be times when we forget which type of data we're manipulating. Luckily, the computer already knows which type of data we're working with, so we can use the `typeof` command in the console to find the answer.
- `typeof` returns a string that tells us which type of data we're seeing.
- Take a look at some examples:
- `typeof 5;`
- `// => "number"`
-
- `typeof "John Doe";`
- `// => "string"`
-
- `typeof true;`
- `// => "boolean"`
-
-
- Make sure to remember the `typeof` command in case you ever need to find out the type of data with which you are working.

Try It!

- Add each line of code to the JS Bin console and hit return to run that line of code.

```
1.typeof 'hello';
```

- The console returns `string`.

```
2.typeof 24;
```

- The console returns `number`.

```
3.typeof 3.45;
```

- The console returns `number`.

```
4.typeof true;
```

- The console returns `boolean`.

```
5.typeof "JS is fun!";
```

- The console returns `string`.

Strings

- Strings are collections of letters and symbols known as *characters*, and we use them to deal with words and text in JavaScript. They come in two varieties, 'single-quote' and "double-quote."
- Although there are some differences between the two, don't worry about them for now. For the purposes of your work in Web Development Fundamentals, they're interchangeable.
- Strings are typically used to store text for people to read.
- `'John'`
- `"Jane"`
- `'123'`

Numbers

- In JavaScript, numbers can always be divided into two groups:
- 1) **Integers (whole numbers)**
- `..., -1,0, 2, 3, 4, 5, ...`
-
- 2) **Floats (decimals)**
- `2.718, 3.14, .5, .25, etc`
-
- This means that JS numbers can be positive, negative, or have a decimal point.

Exercise

- Write the code to perform the actions listed below in the **JavaScript** panel of the JS Bin editor.
- **Instructions**
- Create a variable `petName`. Assign (give) it the value `"Rover"`.
- Create a variable `age`. Assign it the value `8`.
- Create a variable `favoriteToy`. Assign it the value `"ball"`.
- Hit the "Run" button in the "Console" panel and then check the values of the three variables you created by typing each variable name into the "Console" panel and hitting "return."
- Example: `petName;`

Exercise (Continued)

- Update `petName`. The new value should be `Arthur`.
- Update `age`. The new age should be `5`.
- Update `favoriteToy`. The new favorite toy should be `"yarn"`.
- Hit the "run" button in the "Console" panel and then check the values of the three variables you created by typing each variable name into the "Console" panel and hitting enter/return.
- Example: `petName;`
- In the "Console" panel, use the `typeof` command to find the type of data stored in each variable.
- Example: `typeof petName`
- Stuck? Check out the solutions in the cheat sheet at the end of this lesson.