JavaScript Essentials

Strings

Table of Contents

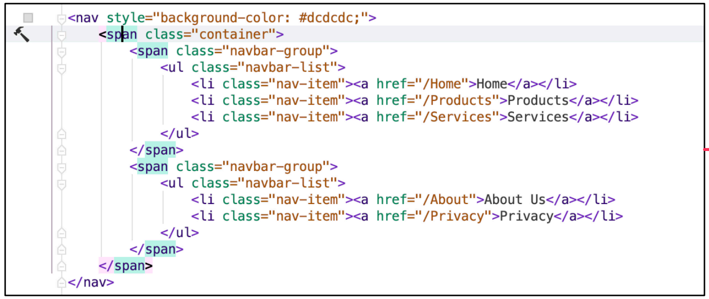
Lesson Objectives

- Understand the power of text in JavaScript and Front-end Development
- Able to create/concatenate strings
- Understand the difference of strings and numbers
- Able to use template strings concatenate strings

Section 1

Overview

Overview – Power of text



```
<nav style="background-color: #dcdcdc;">
    <span class="container">
        <span class="navbar-group">
            <ul class="navbar-list">
                <li class="nav-item"><a href="/Home">Home</a></li>
                <li class="nav-item"><a href="/Products">Products</a></li>
                <li class="nav-item"><a href="/Services">Services</a></li>
            </ul>
        </span>
        <span class="navbar-group">
            <ul class="navbar-list">
                <li class="nav-item"><a href="/About">About Us</a></li>
                <li class="nav-item"><a href="/Privacy">Privacy</a></li>
            </ul>
        </span>
    </span>
</nav>
```

```
h1 { color: white;
background: orange;
border: 1px solid black
padding: 0 0 0 0;
font-weight: bold;
}
/* begin: seaside-theme */

body {
background-color:white;
color:black;
font-family:Arial,sans-serif;
margin: 0 4px 0 0;
border: 12px solid;
}
```

Text everywhere

Overview – Summary

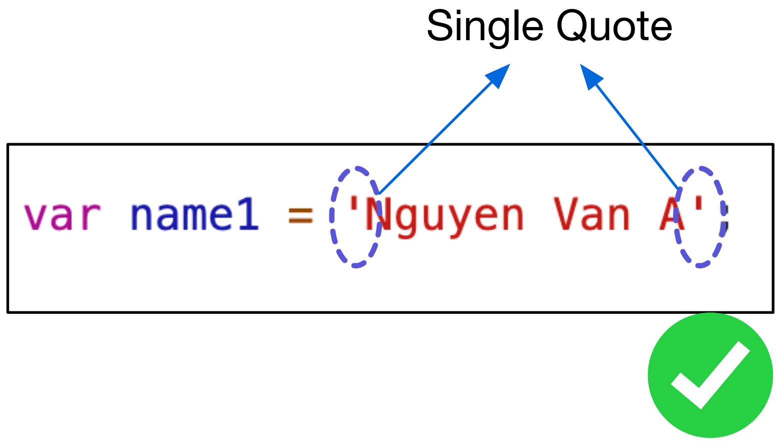
- Pretty much all of the programs we've shown you so far in the course have involved some string manipulation.
- Understand strings give us a huge advantage in Programming

Section 2

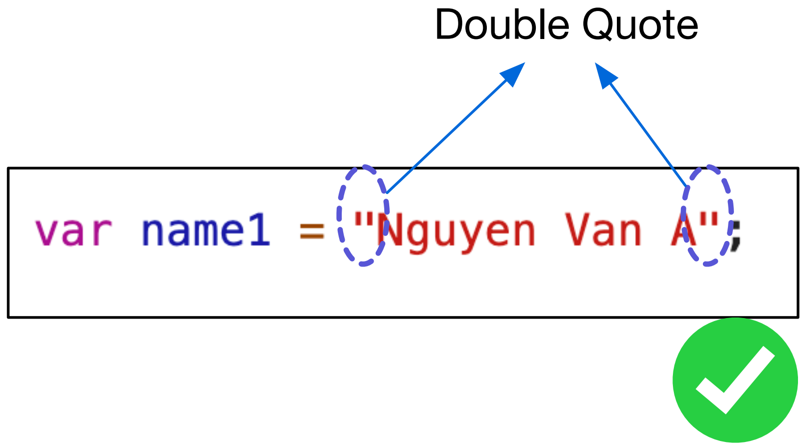
Strings basics

Strings basics – Create a string

- Syntax (correct):



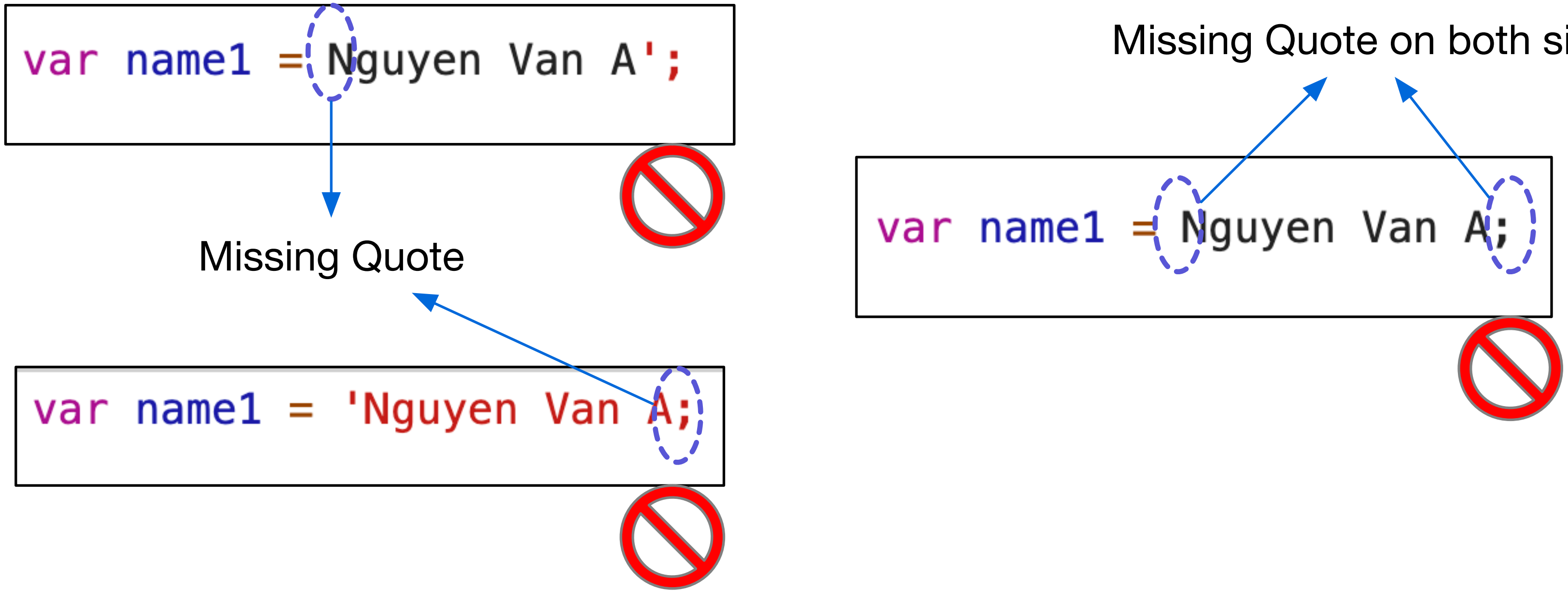
```
var name1 = 'Nguyen Van A';
```



```
var name1 = "Nguyen Van A";
```

Strings basics – Create a string

- Syntax (wrong):



```
var name1 = Nguyen Van A';
```

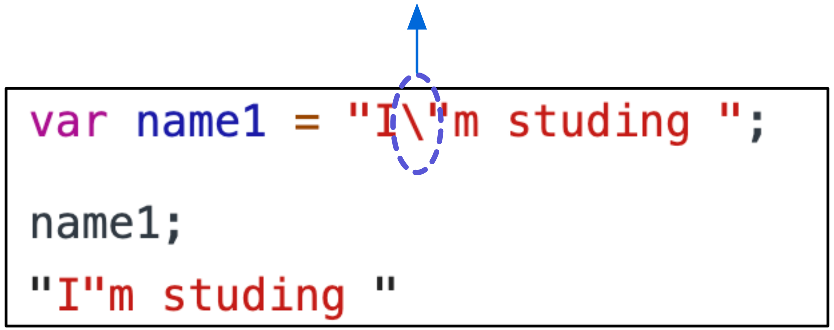
```
var name1 = 'Nguyen Van A;
```

```
var name1 = Nguyen Van A;
```

Strings basics – Escaping characters

- What if we have single quote or double quote in our string ? Like this: I’m studying
- Use Escape notation syntax:

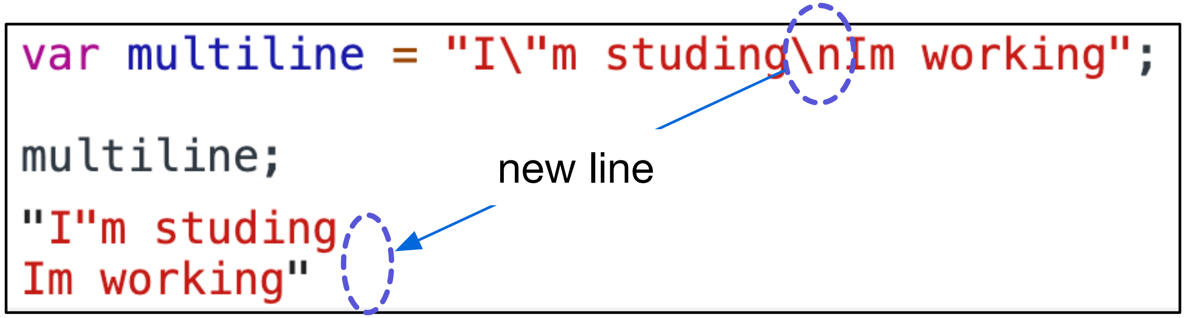
Add \ before quote to escape it



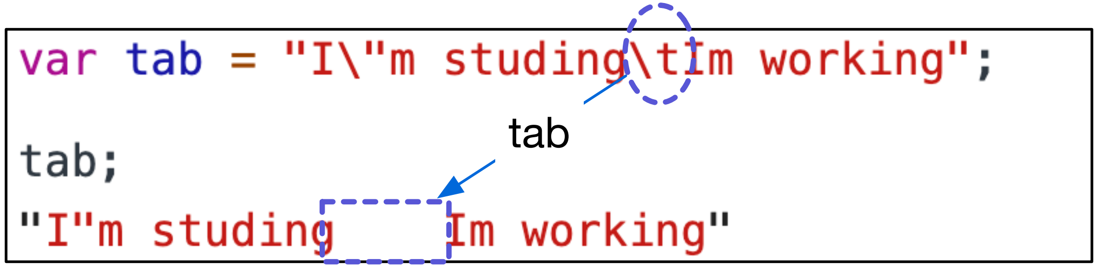
```
var name1 = "I\"m studing ";

name1;
"I"m studing "
```

Strings basics – Escaping characters

- Escape notation usage:



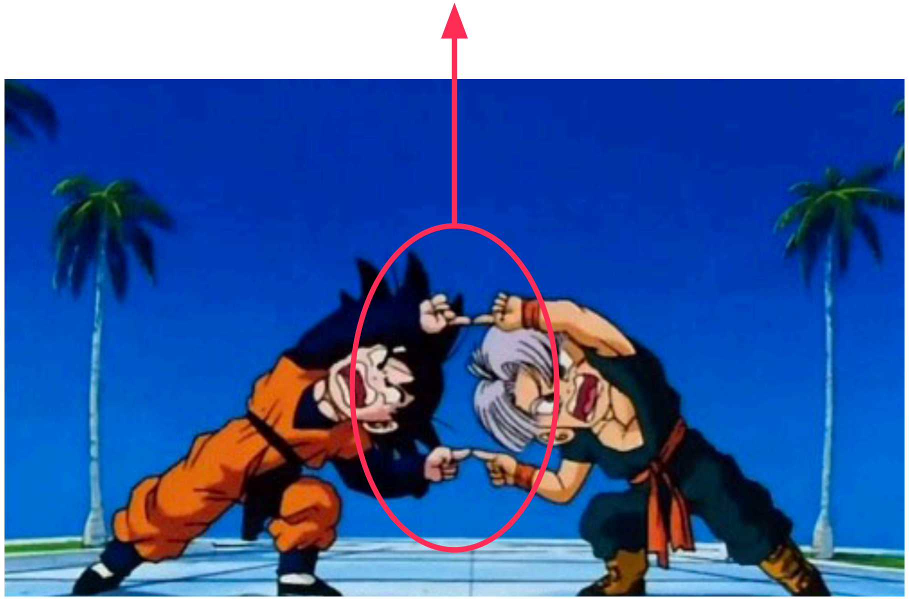
```
var multiline = "I\"m studing\nIm working";

multiline;
"I"m studing
Im working"
```



```
var tab = "I\"m studing\tIm working";

tab;
"I"m studing	Im working"
```

Strings basics – Concatenating strings

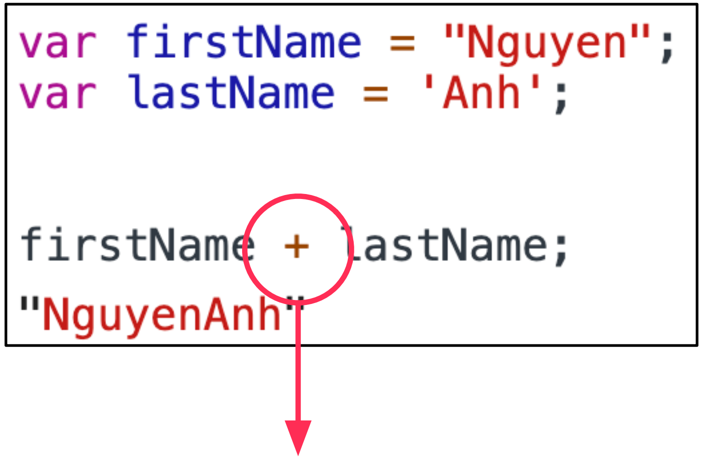
Concatenate (fancy way to call "join" togother)



09e-BM/DT/FSOFT - ©FPT SOFTWARE – Fresher Academy - Internal Use

Strings basics – Concatenating strings

- Joining together strings in JavaScript uses the plus (+) operator, the same one we use to add numbers together, but in this context it does something different

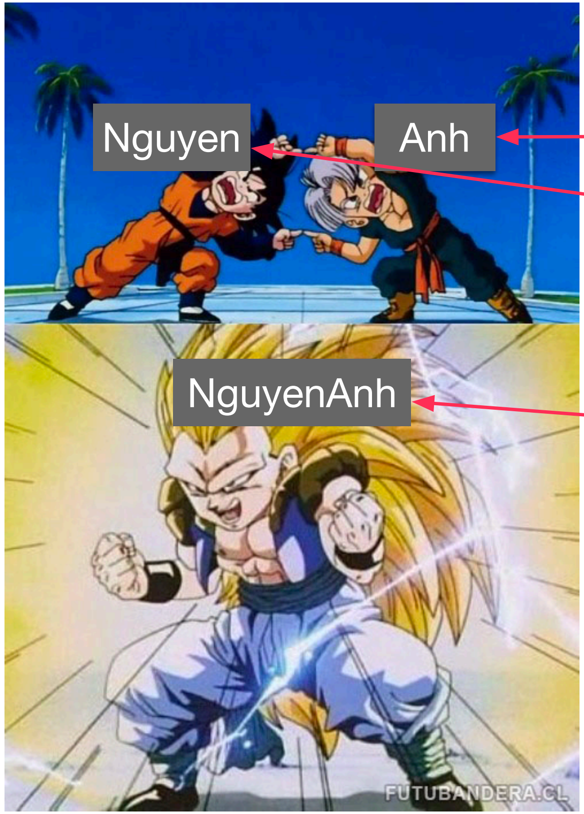


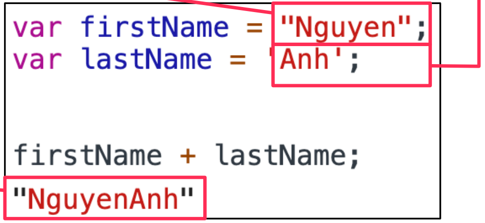
```
var firstName = "Nguyen";
var lastName = 'Anh';


firstName + lastName;
"NguyenAnh"
```

Concatenate

Strings basics – Concatenating strings

- Result:

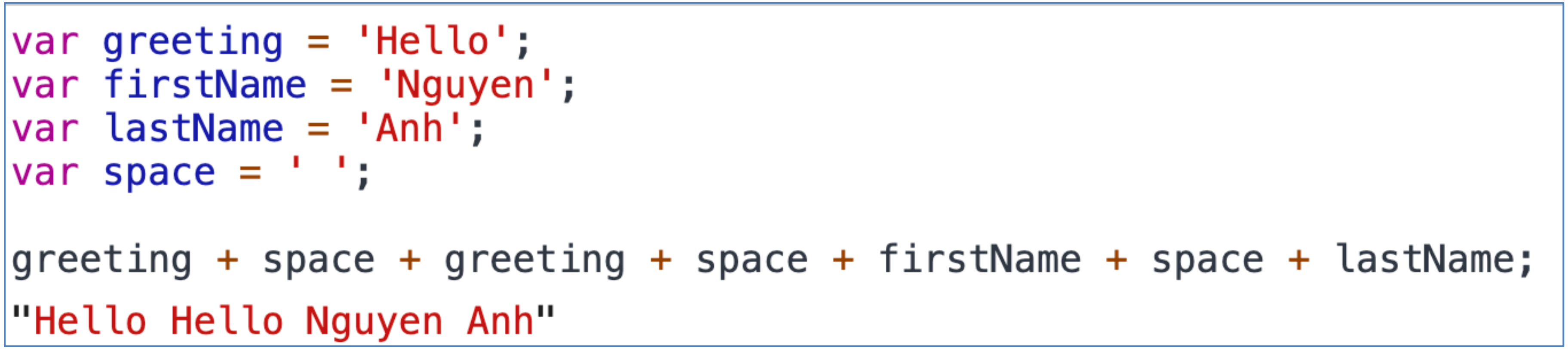




```
var firstName = "Nguyen";
var lastName = 'Anh';


firstName + lastName;
"NguyenAnh"
```

Strings basics – Concatenating strings

- In the last instance, we joined only two strings, but you can join as many as you like, as long as you include a + between each pair. Try this:



```
var greeting = 'Hello';
var firstName = 'Nguyen';
var lastName = 'Anh';
var space = ' ';

greeting + space + greeting + space + firstName + space + lastName;
"Hello Hello Nguyen Anh"
```

Strings basics – Concatenating strings

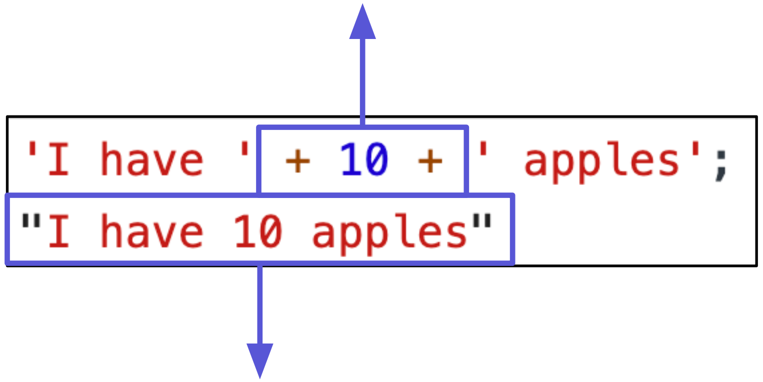
- You can also use a mix of variables and actual strings. Try this:

```
var greeting = 'Hello';
var firstName = 'Nguyen';
var lastName = 'Anh';

greeting + ' ' + firstName + ' ' + lastName + ' from FA';
"Hello Nguyen Anh from FA"
```

Strings basics – Numbers and strings

- So what happens when we try to add (or concatenate) a string and a number? Let's try it in our console:

You might expect this to return an error ?

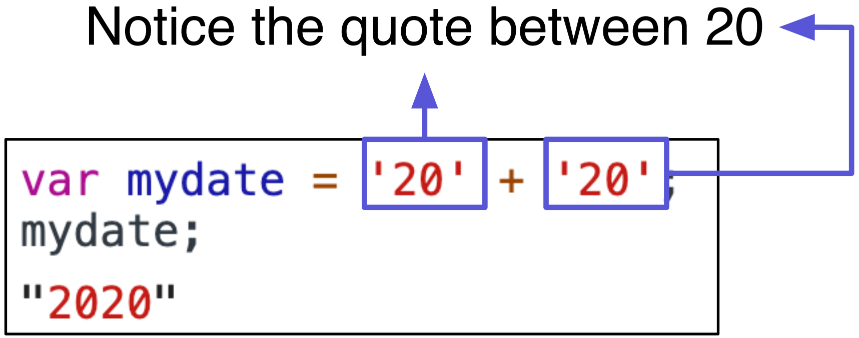


```
'I have ' + 10 + ' apples';
"I have 10 apples"
```

But it works fine!

Strings basics – Numbers and strings

- You can even do this with two numbers — you can force a number to become a string by wrapping it in quote marks. Try the following:

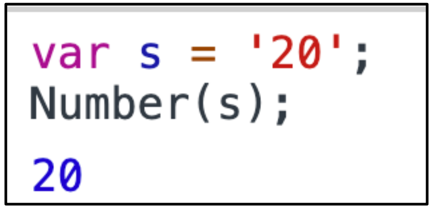


```
var mydate = '20' + '20';
mydate;
"2020"
```

Strings basics – Numbers and strings

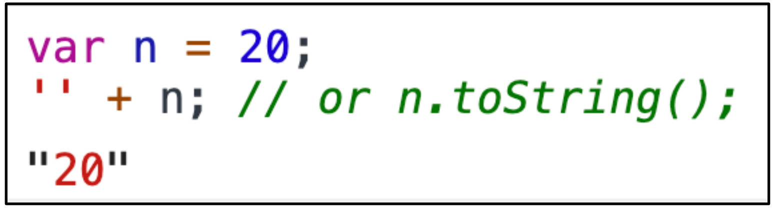
- To convert between number and string:

Convert from string to number



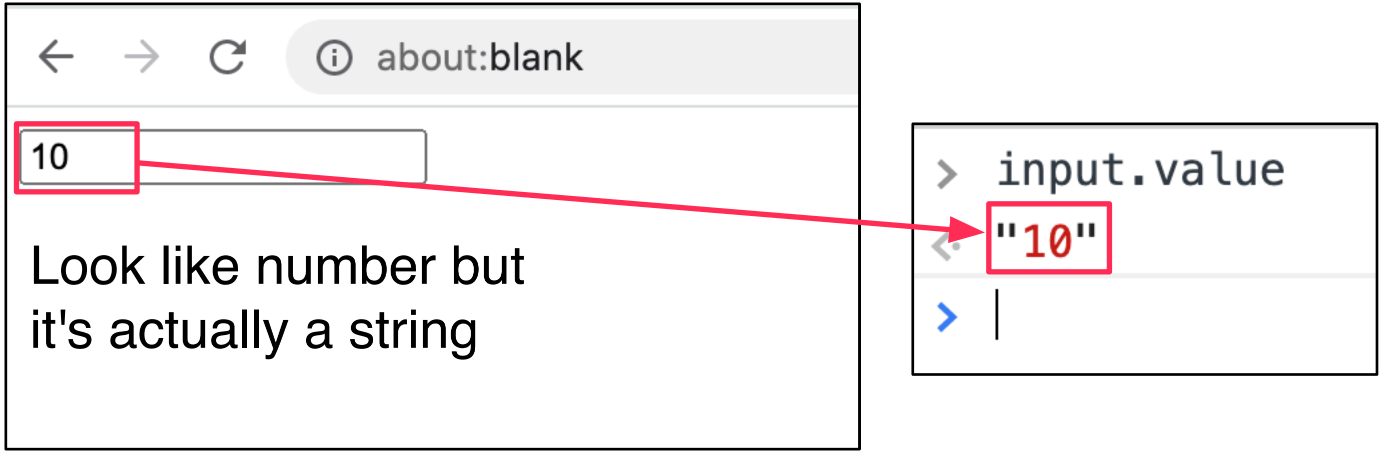
```
var s = '20';
Number(s);
20
```

Convert from number to string



```
var n = 20;
'' + n; // or n.toString();
"20"
```

Strings basics – Numbers and strings

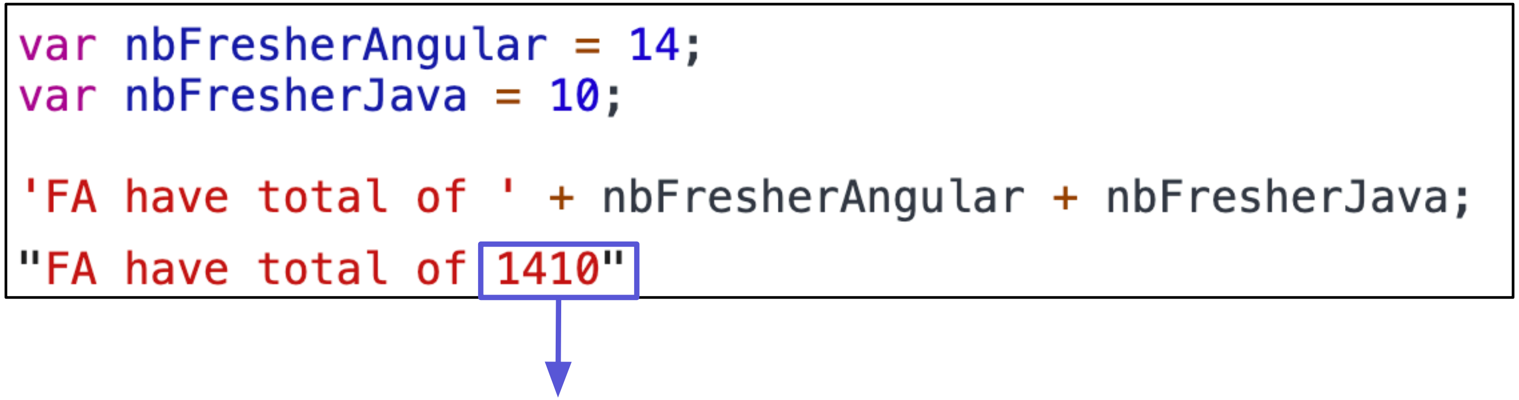


Look like number but it's actually a string

```
> input.value
<· "10"
>
```

Strings basics – Numbers and strings

- Sometimes, it not work the way you wanted

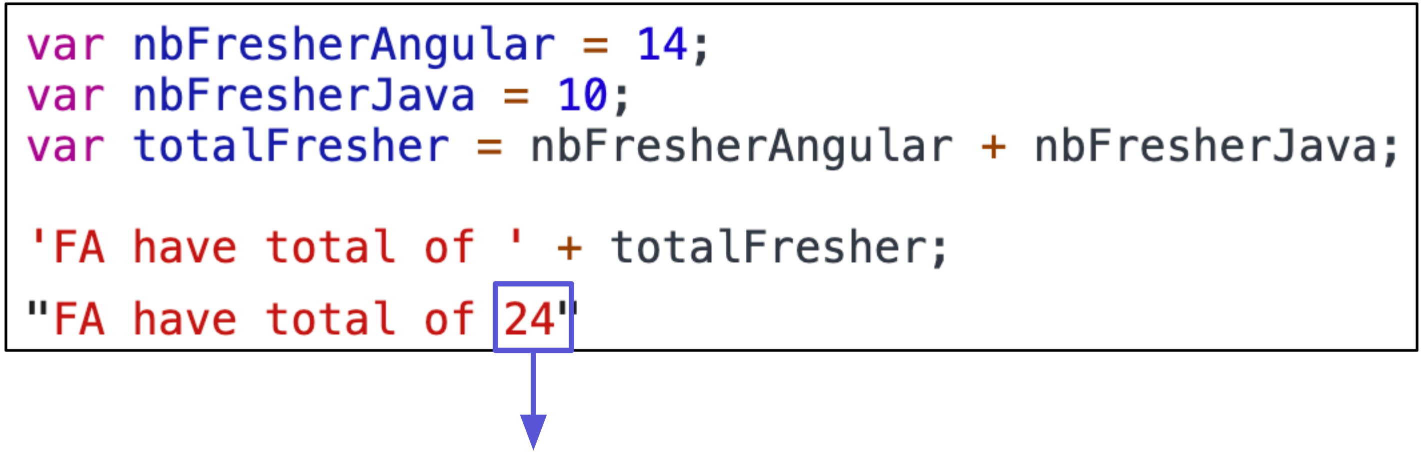


```
var nbFresherAngular = 14;
var nbFresherJava = 10;

'FA have total of ' + nbFresherAngular + nbFresherJava;
"FA have total of 1410"
```

Wrong numbers (must by 24)

Strings basics – Numbers and strings

- To fix it:



```
var nbFresherAngular = 14;
var nbFresherJava = 10;
var totalFresher = nbFresherAngular + nbFresherJava;

'FA have total of ' + totalFresher;
"FA have total of 24"
```

Correct now

Strings basics – Practice Time

Practice Strings manipulation

Strings basics - Summary

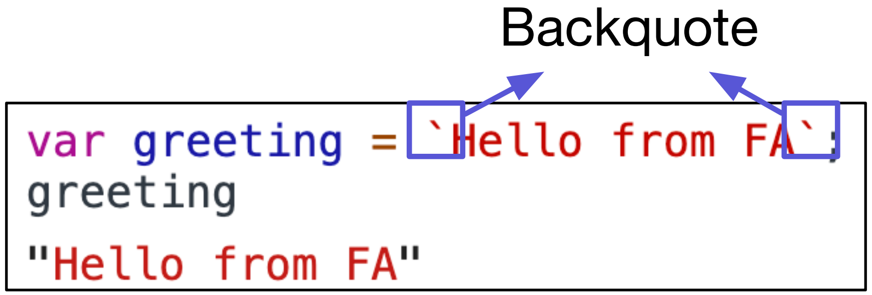
- Create a string using single quote (‘), double quote (“)
- When your string has single quote or double quote, escape it with \’, \”
- Concatenating strings using + operators
- You can concatenate strings and number with + operators
- Take care when you concatenate strings and number it may give wrong answer

Section 3

Template strings

Template strings

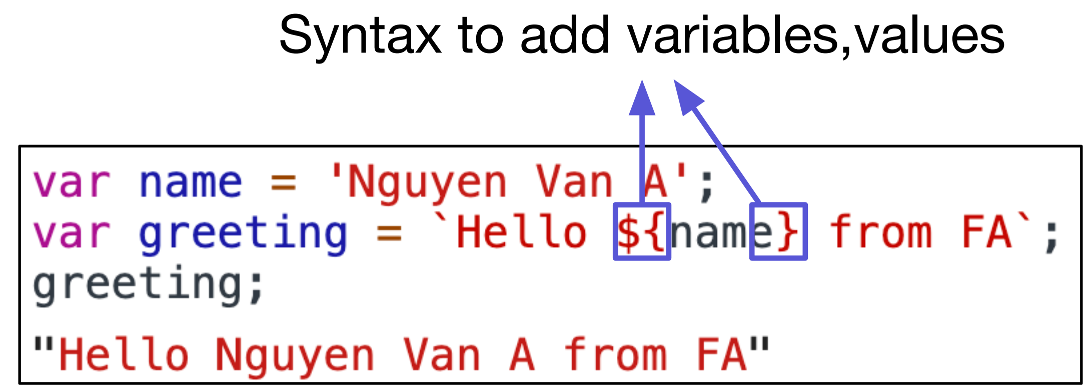
- Another type of string syntax that you may come across is **template literals** (sometimes referred to as template strings). This is a newer syntax that provides more flexible, easier to read strings.



```
var greeting = `Hello from FA`;
greeting
"Hello from FA"
```

Template strings - Concatenation

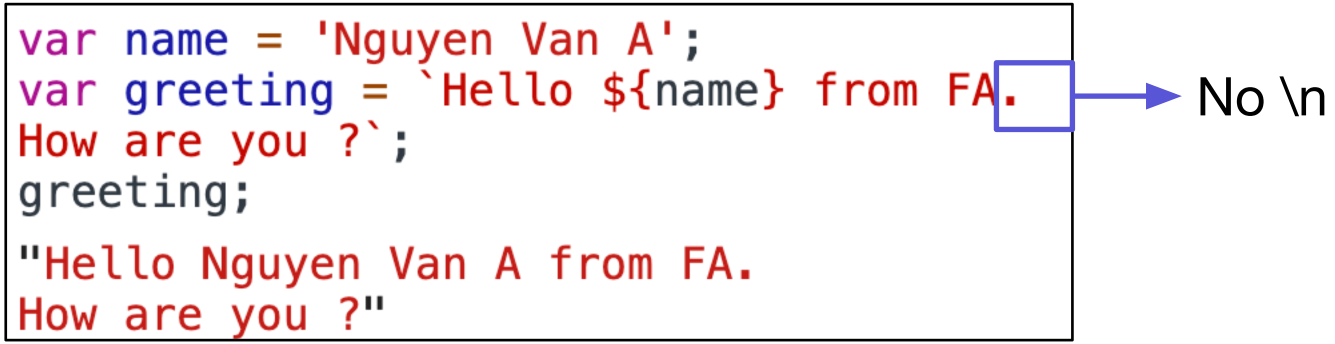
- Template literals simplify concatenating strings:



```
var name = 'Nguyen Van A';
var greeting = `Hello ${name} from FA`;
greeting;
"Hello Nguyen Van A from FA"
```

Template strings - Concatenation

- Template literals work great with multiline:



```
var name = 'Nguyen Van A';
var greeting = `Hello ${name} from FA.
How are you ?`;
greeting;
"Hello Nguyen Van A from FA.
How are you ?"
```

Template strings - Concatenation

- Template literals work great with numbers:

```
var nbFresherAngular = 14;
var nbFresherJava = 10;

`FA have total of ${nbFresherAngular + nbFresherJava}`;
"FA have total of 24"
```

Strings basics – Practice Time

Practice Template Strings

Assignment operators - Summary

- Assignment operators provide useful shortcuts to keep our code cleaner and more efficient
- Can use other variables on the right hand side of each expression as well

Thank you

Q&A