

BCVD1006 – Full Stack Development – Lab 9

- JavaScript DateTime & Strings

Developer Note:

- Please create a separate JavaScript file for each exercise
- You may use the HTML page to trigger your scripts or you may use the JS Playground Editors (REPL) to program and just submit the JS file.
 - <https://repl.it/languages/javascript>
 - <https://jsfiddle.net/>

Exercise 1:

Create a function named **halloween** that has the following requirements:

- Takes a date as a parameter in the format **yyyy/mm/dd** as an input and
- returns "Happy Halloween!" if the date is October 31, else return "Just another day."
- Hint: you may need to use conditional statements

Hint: You will need to use the build it date methods to compare the date

Expected output is as follows:

```
halloween(new Date("2013/10/31")) // "Happy Halloween!"  
  
halloween(new Date("2012/07/31")) // "Just another day."  
  
halloween(new Date("2011/10/12")) // "Just another day."
```

Exercise 2:

Create a function named **hasFriday13** that has the following requirements:

- Given the **month** and **year** as numbers as parameters
- Return whether that month contains a **Friday 13th**.
- Else will return "Crystal Lake" if there is 13th

Hint: January represent 1, February 2, etc

Expected output is as follows:

```
hasFriday13(3, 2020) // It's Friday the 13th!

hasFriday13(10, 2017) // It's Friday the 13th!

hasFriday13(1, 1985) // Crystal Lake.
```

Exercise 3:

Create a function expression named **starWarsRelations** that has the following requirements:

- Given the **name** as a string as the parameter
- Return the string Star Wars relation to Luke Skywalker

Hint: You may need conditionals or switch to compare the different strings

Expected output is as follows:

```
starWarsRelations("Darth Vader") // "Luke, I am your father."
starWarsRelations("Leia") // "Luke, I am your sister."
starWarsRelations("R2D2") // "Luke, I am your droid."
```

Exercise 4:

Create a function expression named **findCase** that has the following requirements:

- Takes a **string** as a parameter
- returns **"upper"** if all the letters in the word are **uppercase**
- returns **"lower"** if all the letters in the word are **lowercase**
- returns **"mixed"** if all the letters in the word are **mixed**

Hint: You may need to use the string methods to compare strings. Also, ignore punctuation, spaces and numbers.

Expected output is as follows:

```
findCase("whisper...") // "lower"

findCase("SHOUT!") // "upper"

findCase("Quiet Voice") // "mixed"
```

Challenge:

Create a function expression named `findingNemo` that has the following requirements:

- Given a string of `words` as a parameter
- Find the word "Nemo" and return a string like "I found Nemo at [position you found it]!"
- Otherwise, return a string "I can't find Nemo :("

```
findingNemo("I am finding Nemo !"); // "I found Nemo at 4!"  
  
findingNemo("Nemo is me"); // "I found Nemo at 1!"  
  
findingNemo("I Nemo am"); // "I found Nemo at 2!"  
  
findingNemo("Finding Dory"); // I can't find Nemo :(
```