

# JavaScript Functions Tasks:

Task01:

Define a function called `rant` which accepts a string argument called `message`. **The function should print out an uppercased version of `message` 3 times** (with 3 separate calls to `console.log`). For example, `rant("I hate beets")` should print out:

1. I HATE BEETS
2. I HATE BEETS
3. I HATE BEETS

Task02:

In some dice games like Craps, a roll of two 1's is called "Snake Eyes". It's generally not a good roll. Please write a function called `isSnakeEyes`, which accepts two numbers as inputs, representing two dice. If the two numbers are both 1's, please print "Snake Eyes!" otherwise print "Not Snake Eyes!"

1. `isSnakeEyes(1,1)` //Snake Eyes!
2. `isSnakeEyes(1,5)` //Not Snake Eyes!
3. `isSnakeEyes(4,5)` //Not Snake Eyes!

**Hint:** Normally a function will `return` a value, but for the sake of having this work with the Udemy interpreter we will be using `console.log()` instead of `return`, to print the output.

Task03:

## Return Value Practice

It's time to practice returning values from a function! Write a simple function `multiply` which accepts two numerical arguments and returns their product (multiply them together). **Make sure to return the value instead of printing it!**

1. `multiply(2,3)` // 6
2. `multiply(9,9)` // 81
3. `multiply(5,4)` // 20

Task04:

## isShortsWeather Function

I often struggle to know whether I should wear shorts or pants on a given day. (this is a complete lie. It's really not that hard to decide.) Please help me decide by writing me a function called `isShortsWeather`.

- It should accept a single number argument, which we will call `temperature` (but you can name it whatever you want, of course).
- If `temperature` is **greater than or equal to 75**, return `true`.

- Otherwise, return `false`.
  - This exercise assumes temperature is in Fahrenheit. I apologize to all my Celsius-using students!
1. `isShortsWeather(80)` //true
  2. `isShortsWeather(48)` //false
  3. `isShortsWeather(75)` //true

Task05:

### Last Element Exercise

Please write a function called `lastElement` which accepts a single array argument. The function should return the last element of the array (without removing the element). If the array is empty, the function should return null.

1. `lastElement([3,5,7])` //7
2. `lastElement([1])` //1
3. `lastElement([])` //null

Task06:

### Capitalize Exercise

Define a function called `capitalize` that accepts a string argument and **returns a new string with the first letter capitalized** (but the rest of the string unchanged). For example:

1. `capitalize('eggplant')` // "Eggplant"
2. `capitalize('pamplemousse')` // "Pamplemousse"
3. `capitalize('squid')` // "Squid"

### Hints:

- Remember that strings are immutable, meaning that you cannot simply change the first letter in the original string. You will need to make a new string that you return.
- Single out the first letter and capitalize it. (use a string method to help!)
- Add that first letter to the rest of the original string, sliced to omit the original first letter (use a string method to help!)
- For example: 'eggplant' becomes 'E' + 'ggplant'

Task07:

### Sum Array Exercise

Write a function called `sumArray` which accepts a single argument: an array of numbers. It should return the sum of all the numbers in the array.

1. `sumArray([1,2,3])` // 6
2. `sumArray([2,2,2,2])` // 8
3. `sumArray([50,50,1])` // 101

**Hints:**

- You'll need a variable to keep track of the total. It should start out as zero.
- Loop over the array and for each element, add it to the total variable.
- After you have added every number to total, return total.

Task08:

**Days Of The Week Exercise**

Write a function called `returnDay`. this function takes in one parameter (a number from 1-7) and returns the day of the week (1 is Monday, 2 is Tuesday, etc.) If the number is less than 1 or greater than 7, the function should return `null`. In some countries Sunday is treated as the first day of the week, but for this exercise we will use Monday as the first day.

Hint: store the days of the week in an array, or use an object with numerical keys. When the function is called, plug the number into the array/object you've created to retrieve the corresponding day name and then return that value.

1. `returnDay(1) // "Monday"`
2. `returnDay(7) // "Sunday"`
3. `returnDay(4) // "Thursday"`
4. `returnDay(0) // null`

task09:

**Function Expression Exercise**

Define a function that returns the square of a number. **Define the function as a function expression, stored in a variable called `square`.**

e.g  
`square(4) //16`

1. `square(3) //9`

Task10: