⬚ Exercises

Exercise 1

Write a function `displayName` that takes two arguments, a name and a surname, and displays them James Bond-style.

Look at the test cases to understand, your function must work the same way!

Here are a few test cases:

Test: displayName("James", "Bond")
Output:
```My name is Bond, James Bond
```

Test: displayName("Ada", "Lovelace")
```My name is Lovelace, Ada Lovelace
```

Hint: When we say *display*, you need to use `console.log`.

Exercise 2

You probably know the game "Rock, paper, scissors" (if you don't, look up the rules and have a couple of games with your neighbour!).

Write a function that announces the result of the game.

Your function `rockPaperScissors` will take two arguments:

- the first is the move of the first player, between `"rock"`, `"paper"` and `"scissors"`
- the second is the move of the second player, between `"rock"`, `"paper"` and `"scissors"`

Your function will print `First player wins` if the first player wins, `Second player wins` if the second player wins, and `Draw` if nobody wins.

Here are a few test cases:

Test: rockPaperScissors("rock", "rock")
Output:
```Draw
```

Test: rockPaperScissors("rock", "paper")
Output:
```Second player wins
```

Test: rockPaperScissors("rock", "scissors")
Output:
```First player wins
```

Test: rockPaperScissors("paper", "scissors")
Output:
```Second player wins
``

## Exercise 3

Modify the `rockPaperScissors` function so that it *returns* the result rather than *displays* it.

You will also change your function to handle errors. If one of the argument given to the function is neither `"rock"`, `"paper"` or `"scissors"`, the function will return `Argument error`.

Here are a few test cases:

Test:   rockPaperScissors("paper", "scissors")
Return:
```'Second player wins'
```

Test:   rockPaperScissors("rock", "potatoes")
Return:
```'Argument error'
```

Test:   rockPaperScissors("glue", "scissors")
Return:
```'Argument error'
```

Test:   rockPaperScissors("rock", "rock")
Return:
```'Draw'
```

## Exercise 4

Write a function `printVowels` that takes a string as argument and displays only the vowels of the string.

1/3

Vowels are `a`, `e`, `i`, `o`, `u`.


Here are a few test cases:

Test:   printVowels("hello world")
Output:
```eoo
```


Test: printVowels("kangaroo")
Output:
```aaoo
```


Test: printVowels("cheeseburger")
Output:
```eeeue
```


Test: printVowels("rhythm")
Output:
```
```

Exercise 5

Write a function `abbrev` that takes a string as argument, and returns an abbreviation of the string.

The abbreviation will be made of the first 3 characters of the string, followed by a space and the total number of characters in the string.

If the string is shorter than 3 characters, it will be padded with spaces. (Look at the examples!)

Here are a few test cases:

Test: abbrev("Nice")
Return:
```'Nic 4'
```


Test: abbrev("I am your father")
Return:
```'I a 16'
```


Test: abbrev("Supercalifragilisticexpialidocious")
Return:
```'Sup 34'
```


Test: abbrev("!")
Return:
```'!  1'
```


Test: abbrev("Yo")
```'Yo  2'
```

Exercise 6

Write a function `pyramid` that takes the size of the base as argument, and draws a pyramid of `#`.

Test: pyramid(9)
Output:
```    #
   ###
  #####
 #######
#########
```


Test: pyramid(1)
Output:
```#
```


Test: pyramid(5)
```  #
 ###
#####
```


Test: pyramid(6)
```  ##
 ####
######
```

Exercise 7

Write a function `subDigits` that takes an integer as argument, and returns it after substracting each digit to that integer.

For example, if the integer is 12, it returns 12 - 1 - 2 => 9.

If the integer is negative, the function returns "Argument Error".

Here are a few test cases:

Test: subDigits(12)
Return:
```9
```

Test: subDigits(4000000)
Return:
```3999996
```

Test: subDigits(0)
Return:
```0
```

Test: subDigits(451)
Return:
```441
```

Test: subDigits(-12)
Return:
```'Argument Error'
```

To see this cheat sheet and its attached documents, please visit https://inco-academy.360learning.com/course/play/5e820154bc3ed73d824986f4