homework 1, version 4

```
md"_homework 1, version 4_"
```

Submission by: ***Emile Tenezakis*** (emileten@mit.edu)

Homework 1 - *convolutions*

18.S191 , fall 2020

This notebook contains *built-in, live answer checks*! In some exercises you will see a coloured box, which runs a test case on your code, and provides feedback based on the result. Simply edit the code, run it, and the check runs again.

For MIT students: there will also be some additional (secret) test cases that will be run as part of the grading process, and we will look at your notebook and write comments.

Feel free to ask questions!

```
student = (name = "Emile Tenezakis", kerberos_id = "emileten")
```

```
# edit the code below to set your name and kerberos ID (i.e. email without @mit.edu)

student = (name = "Emile Tenezakis", kerberos_id = "emileten")

# press the ▶ button in the bottom right of this cell to run your edits
# or use Shift+Enter

# you might need to wait until all other cells in this notebook have completed running.
# scroll down the page to see what's up
```

Let's create a package environment:

```
begin
    import Pkg
    Pkg.activate(mktempdir())
end
```

We set up Images.jl again:

```
begin
    Pkg.add(["Images", "ImageMagick"])
    using Images
end
```

Exercise 1 - *Manipulating vectors (1D images)*

A `Vector` is a 1D array. We can think of that as a 1D image.

```
example_vector = [0.5, 0.4, 0.3, 0.2, 0.1, 0.0, 0.7, 0.0, 0.7, 0.9]
```

```
colored_line(example_vector)
```

Exerise 1.1

👉Make a random vector `random_vect` of length 10 using the `rand` function.

```
random_vect =
[0.355274, 0.0936927, 0.0515978, 0.194034, 0.796445, 0.629625, 0.84069, 0.84646, 0.269016
```

```
random_vect = rand(Float64, 10)
```

Got it!

Awesome!

Hint

[illegible]

```
    for i in x
        x_sum = x_sum + i
    end
    return x_sum/length(x)
end
```

```
2.0
```

```
mean([1, 2, 3])
```

Got it!

Keep it up!

👉Define `m` to be the mean of `random_vect`.

```
m = 0.4618349491719069
```

```
m = mean(random_vect)
```

Got it!

Great!

👉Write a function `demean`, which takes a vector `x` and subtracts the mean from each value in `x`.

```
demean (generic function with 1 method)
```

```
function demean(x)
    return x .- mean(x)
end
```

Let's check that the mean of the `demean(random_vect)` is 0:

Due to floating-point round-off error it may not be exactly 0.

```
end
```

Got it!

Awesome!

Exercise 1.3

👉Write a function that turns a `Vector` of `Vector`s into a `Matrix`.

```
vecvec_to_matrix (generic function with 1 method)
```

```
function vecvec_to_matrix(vecvec)
    nrow, ncol = length(vecvec), length(vecvec[1])
    mat = zeros(nrow, ncol)
    for i in 1:nrow
        mat[i,:] = vecvec[i]
    end
    return mat
end
```

```
2×2 Matrix{Float64}:
 1.0  2.0
 3.0  4.0
```

```
vecvec_to_matrix([[1,2], [3,4]])
```

Got it!

Exercise 2 - *Manipulating images*

In this exercise we will get familiar with matrices (2D arrays) in Julia, by manipulating images. Recall that in Julia images are matrices of `RGB` color objects.

Let's load a picture of Philip again.

```
philip_file = "/var/folders/wr/kdh2bp89031fm4fj10t6b12w0000gn/T/jl_IzNV9x"
```

```
philip_file = download("https://i.imgur.com/VGPeJ6s.jpg")
```

```
philip =
```

Exercise 2.1

👉Write a function **`mean_colors`** that accepts an object called `image`. It should calculate the mean (average) amounts of red, green and blue in the image and return a tuple `(r, g, b)` of those means.

```
md"""
#### Exercise 2.1
👉 Write a function **`mean_colors`** that accepts an object called `image`. It should calculate the mean (average) amounts of red, green and blue in the image and return a tuple `(r, g, b)` of those means.
"""
```

```
mean_colors (generic function with 1 method)
```

```
function mean_colors(image)
    sum = RGB(0,0,0)
    elements = 0
    vecImage = matrix_to_vecvec(image)
    for vec in vecImage
```

```
    end

    function quantize(image::AbstractMatrix)
        return broadcast(quantize, image)
    end
end
```

```
(0.2, 0.9)
```

```
quantize(0.267), quantize(0.91)
```

Got it!