Web Programming

In-class activities
Week 3 – Introduction to JavaScript

Prof. Josué Obregón

Department of Industrial Engineering- ITM

Seoul National University of Science and Technology

In-class activities

- 11:05 – 5 minutes quiz (available on eclass)
  - You can check score and answers at 11:11
- 11:10 - Q&A
- (adapted) Pair-programming written exercise
  1. Write a short program/web page according to the problem (on paper)
  2. Switch paper with your partner
  3. Check your partner's code (marking errors)
  4. Write your partner's code on your computer and verify it works
  5. Discuss errors and each others' solutions
- Hands-on exercises

Starter: Pair Programming Exercise

- Body Mass Index is a measure of body fat based on height and weight.

$$BMI = \frac{weight\ (kg)}{height^2(m)}$$

- Write a small script that:
  - Define a `bmi(weight, height)` function computes and return
    - Example: `bmi(60, 1.7)` → 20.76
  - Define 2 `person` objects, with `weight` and `height` properties
  - Define an array `people` containing your defined `person` objects
  - Using the `forEach` array method, apply the `bmi` function to each element of `people` and print in in console

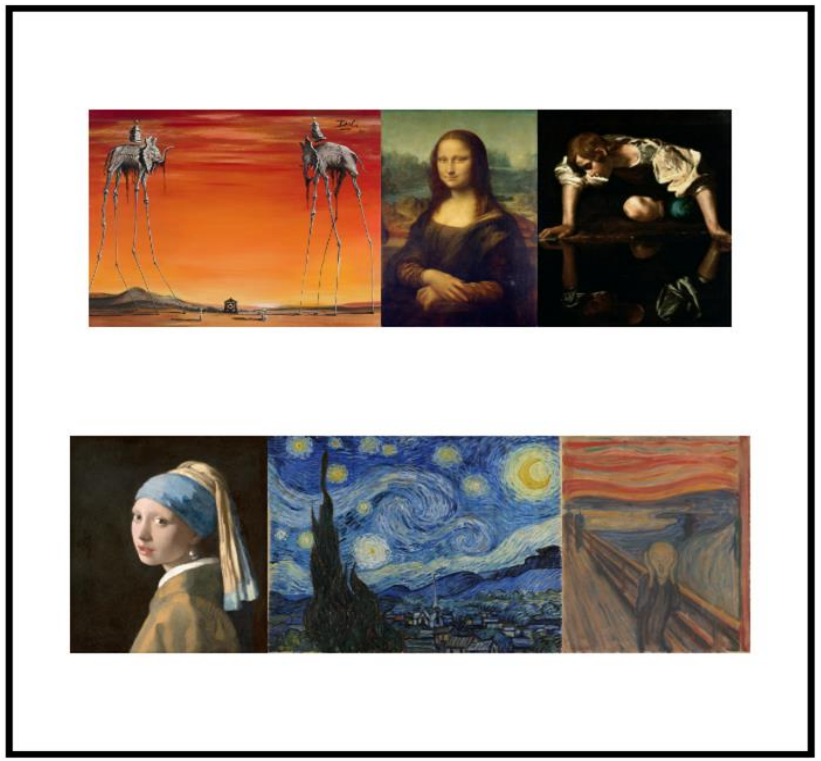
Exercise 1: Picture Gallery

Given the html, gallery.html (download from eclass), add some css to your website to match the appearance below:

- Each image should be **120px** tall
- Each image should have a **20px** **antiquewhite** mat between the images and the frame
- Each image should have a solid black frame with a width of **10px**.
- There should be **10px** of space around the frame

Exercise 2: Picture Collage

Given the html in collage.html (download from eclass), write collage.css to match the appearance below:



- The images should have a **height** of **100px**
- The image container should:
  - Have **20px** of **padding** on all sides
  - Have a **height** and **width** of **300px**
  - Center align its children on the main and cross axes (hint: use flex!)
  - **Wrap** any overflowing content
  - Have a **3px solid black** border

Exercise 3: Analyze test score

1. Create a function called `analyzeScores` that takes an array of numerical test scores as a parameter.
2. The function should return an object containing three properties:
   - `highest`: the highest score in the array
   - `lowest`: the lowest score in the array
   - `average`: the average score (sum of all scores divided by the number of scores)
3. Define an array of at least five test scores (for example, `[78, 92, 85, 100, 67]` or any other scores).
4. Call the `analyzeScores` function with your array of scores, store the returned object in a variable, and print the result in the console.