EECS 1012: LAB #3 – HTML+CSS+JS (**January 21-27, 2019**)

#Important reminders

1) You **must** attend **your assigned** lab session (we will be marking your submission in the lab).
2) Do the mini-quiz prelab quiz. It will open on Wednesday by 13:00 and will expire on Thursday by 12:59. It's part of your lab grade. Each lab counts up to 2% of your overall grade.
3) You must arrive on time – anyone later than 15 minutes may not be admitted to the lab.

#Important pre-lab works you need to do before going to the lab

1) Download this lab files and read them carefully to the end.
2) You should have a good understanding of

- Events (such as `onclick`, `ondblclick`)
- `document.getElementById().innerHTML`
- `document.getElementById().setAttribute()`
- `document.getElementById().style`
- `document.getElementById().style.display`

1. GOALS/OUTCOMES FOR LAB

- To learn how to change the *behaviour* of an HTML document using JavaScript,

2. LAB – TASK

1) You are provided a simple `myLearningKit.HTML` document and supporting files such as `myLearningKit.CSS` and `myLearningKit.js` as well some images. Your first job will be to improve the presentation of the HTML file slightly. Then, you major task is to improve the behaviour of the HTML file by providing `JavaScript` codes.
2) You will generate at least five HTML and JS files in this process. You should demo each HTML file to the TA. For that, please, have each HTML file open in a different tab so you can show the progression.
3) See next pages for details on how to modify your HTML and JS files.

3. SUBMISSIONS

1) Manual verification by TA

You will need to have one of the TAs verify your lab before submission. The TA will look at your various files in their progression. The TA may ask you to make minor modifications to the lab to demonstrate your knowledge of the materials.

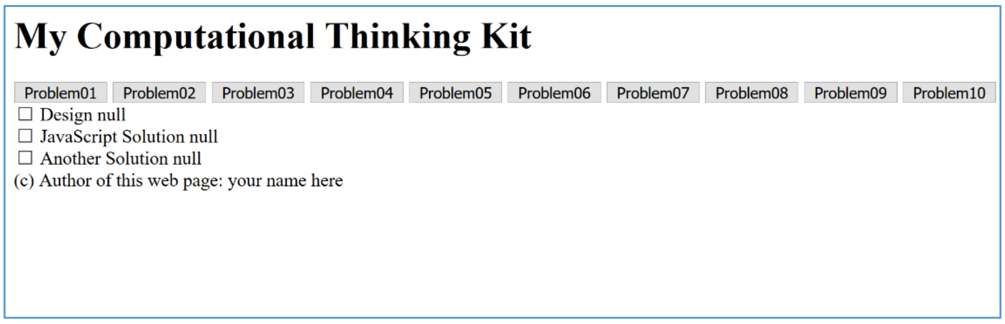
The TA will mark your name off a list and ask you to sign that you have been verified.

2) Moodle submission

You will see an assignment submission link on Moodle.

1) Create a **folder** named "**Lab3**" and copy all of your lab materials inside (image folder, `myLearningKit_Ex{1,2,3,4,5}.HTML` and `myLearningKit_Ex1.css`, and `myLearningKit_Ex{2,3,4,5}.js`). This folder should be compressed (or tar.gz on the VirtualBox machines) and the compressed file submitted. Please follow the submission instructions in the following video that we used for previous labs: https://www.youtube.com/watch?v=stEOh6ntV5o

STARTING POINT: **myLearningKit**.**html**, **myLearningKit**.**css**, **myLearningKit**.**js**, and images in the **image folder**. You are given the following HTML file.



My Computational Thinking Kit

Problem01 Problem02 Problem03 Problem04 Problem05 Problem06 Problem07 Problem08 Problem09 Problem10

☐ Design null
☐ JavaScript Solution null
☐ Another Solution null
(c) Author of this web page: your name here

This file is not connected to a CSS file. In Exercise 1, you improve the *presentation* of this HTML document by modifying the HTML and CSS files slightly.

Before proceeding further, open **myLearningKit.HTML** with an editor and carefully learn about its content and structure.

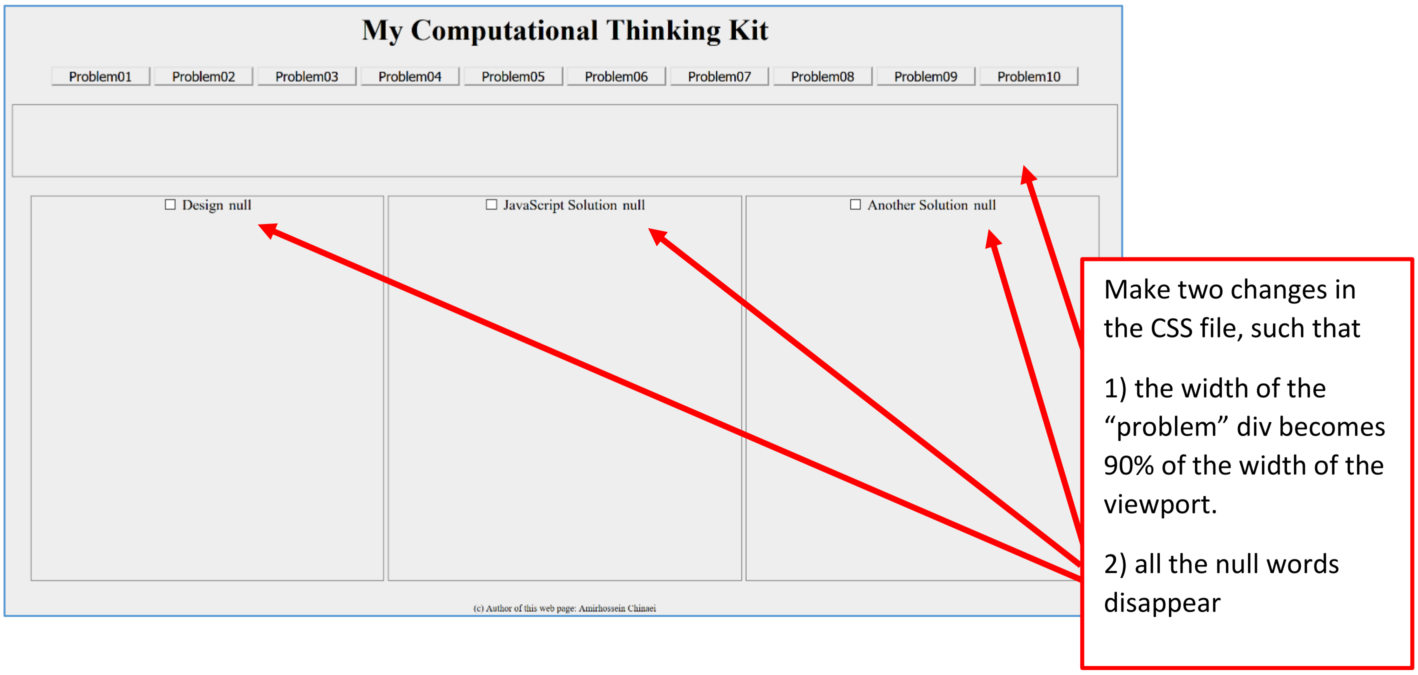
(Note that this document eventually is going to be an interface (in particular, a webpage) that contains some basic computational problems together with your flowcharts and implementations. However, it takes a couple of weeks until we get there. So, for now we (ab)use the document towards demonstrating foods.)

Exercise 1: (CREATE: **myLearningKit_Ex1.HTML** and **myLearningKit_Ex1.css)**
Copy **myLearningKit.HTML** to a new file named **myLearningKit_Ex1.html**. Copy **myLearningKit.CSS** to a new file named **myLearningKit_Ex1.css**.

Make two changes to **myLearningKit_Ex1.html**, as follows:

1) Connect it to **myLearningKit_Ex1.CSS** by adding a link in the head element.
2) In the footer element, replace “your name here” with your name.

When you open your HTML file with browser, you should see the following result:



My Computational Thinking Kit

Problem01 Problem02 Problem03 Problem04 Problem05 Problem06 Problem07 Problem08 Problem09 Problem10

☐ Design null ☐ JavaScript Solution null ☐ Another Solution null

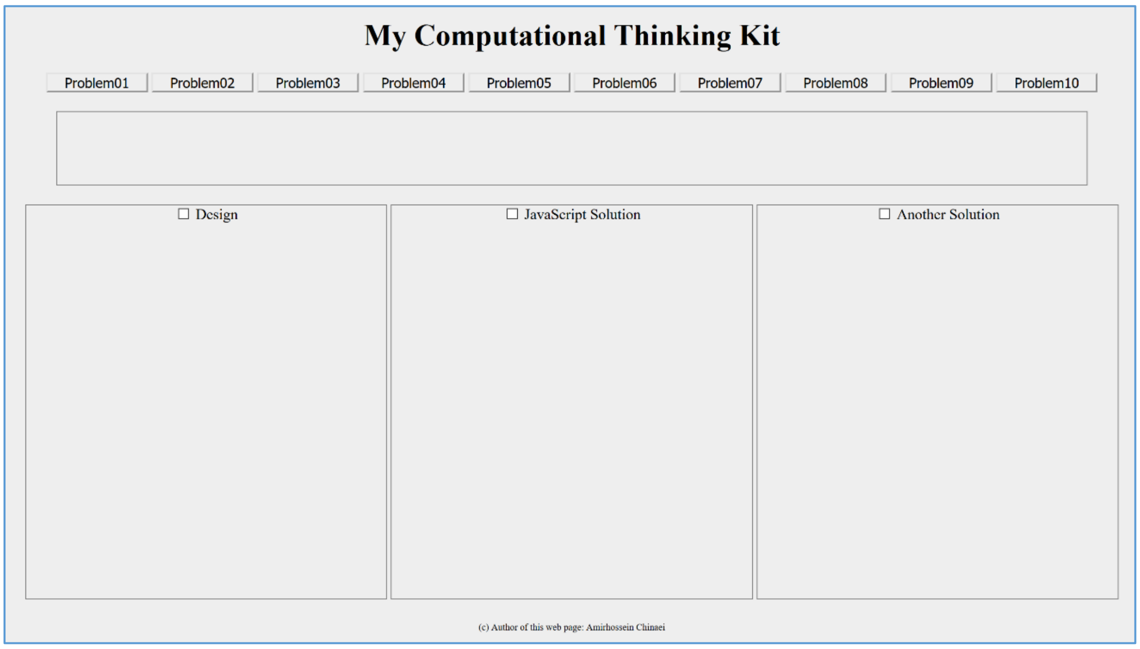
(c) Author of this web page: Amirhossein Chinaei

Make two changes in the CSS file, such that

1) the width of the “problem” div becomes 90% of the width of the viewport.

2) all the null words disappear

Now, as stated in the red box above, make two changes to **myLearningKit_Ex1.css**, as follows. Currenlty, the width of the “problem” division is 100% of its parent (which is the body element). We want to have the width of the “problem” division set to 90% of the width of the viewport. (Find it in w3schools how we can use width of viewport.) Also, as you can see in the HTML file, the source file of the images as well as the alt attributes are all set to *null*. In other words, we do not want yet these img tags to display anything. But, in the picture above, the word "null" is displayed. If, in the HTML file, we remove the attributes src and alt from the images, our HTML code is no longer valid. Hence, we need to make a change in the CSS file such that these images do not display anything for now. After making these two changes in the CSS file, you should have the following result:



Recall: you should figure out how to make such changes by yourself (or, by some quick searches in the web). Asking for help from your friends or from TAs should be a last resort.

Optional: you may want to change the coloring and borders of some of the elements above to make your kit page more appealing. We do not grade it though.

Exercise 2: (CREATE: **myLearningKit_ex2.html** and **myLearningKit_Ex2.js)**
Copy **myLearningKit_Ex1.html** to a new file named **myLearningKit_Ex2.html**. Copy **myLearningKit.js** to a new file named **myLearningKit_Ex2.js**.

When you open the **myLearningKit_Ex2.html** in your browser, if you click on any of the buttons, nothing happens. In other words, we have not defined any behaviour for our page yet. Let's do it step by step.

1) Add an attribute `onclick` to the button designated for Problem01 such that when clicked function `p01Func` is called.

```
onclick="p01Func()"
```

2) The body of `p01Func`() can be provided in the HTML file internally. In particular, we can add a script element and write our `JavaScript` code inside it. But, similar to what we do for CSS, we encourage you to have your `JavaScript` code in an external file. To do so, add the following line to your HTML file:

```
<script src="myLearningKit_Ex2.js"></script>
```

3) Open **myLearningKit_Ex2.js**. We already provided you with the skeleton of `p01Func`().

```
Function p01Func(){
…
}
```

The body of the function is what we write in the pair of curly braces. We want to update the content of the "problem" division with the following paragraph:

I'm looking for a type of pancake originating from the Indian subcontinent, made from a fermented batter. It is somewhat similar to a crepe in appearance.

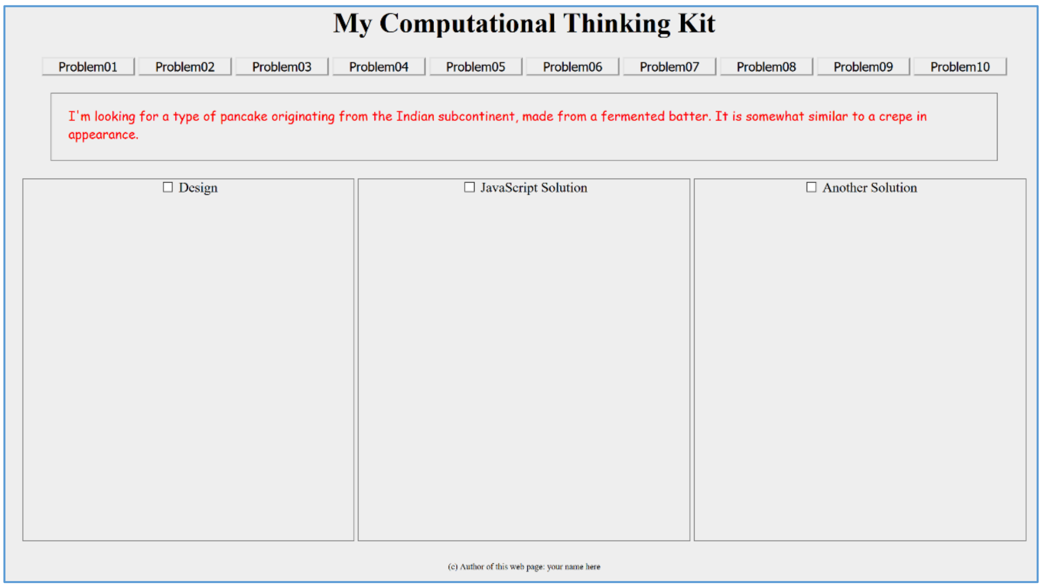
To do so, we need to get from our HTML file the element that its ID is "problem", by the following `JavaScript` code:

```
document.getElementById("problem")
```

This would get us a construct, that we know it as an **object**.
Then, we need to assign the paragraph above to the `innerHTML` of this object. Here, it's the `JavaScript` code:

```
document.getElementById("problem").innerHTML="<p>…</p>";
```

where **...** is replaced with the red paragraph above. Now, when we click on the button representing `Problem01`, we see the following result.



My Computational Thinking Kit

I'm looking for a type of pancake originating from the Indian subcontinent, made from a fermented batter. It is somewhat similar to a crepe in appearance.

(c) Author of this web page: your name here

4) (At this point, you may want to revisit the **myLearningKit_Ex2.html** file and look up the `img` elements and their IDs.) The idea is that for `Problem01`, `dosaDesign.jpg` is assigned to the image that its ID is "flowchart" and `dosa1.jpg` is assigned to the image that its ID is "js".
So, you need to add more `JavaScript` code to what you did in Step 3. When you get an element by its ID, you can set any of its attributes by using `setAttribute()`. Since we want to set `src` attribute of the element with ID "flowchart" to `dosaDesign.jpg`, we need to add the following code:

```
document.getElementById("flowchart").setAttribute("src","images/dosa/dosaDesign.jpg");
```

Because we do not want to display the image yet, we need to set the display property of its style to none.

Similar to above, assign `dosa1.jpg` to `src` of the element that its ID is "js" and set its display to none. (You may want to download another image for dosa from internet (save it as `dosa2.jpg`) and assign it to the element that its ID is "another" and set its display to none.)

Note that the images are still not displayed. We will display them in the next exercise.

Exercise 3: (CREATE: **myLearningKit_Ex3.html** and **myLearningKit_Ex3.js)**
Copy **myLearningKit_Ex2.html** to a new file named **myLearningKit_Ex3.html**. Copy **myLearningKit_Ex2.js** to a new file named **myLearningKit_Ex3.js**.

The idea is to modify the new files such that when we check/uncheck the checkboxes by clicking on them, the images display/disappear, respectively.

1) First, we need to add the `onclick` event to checkbox "check1" in **myLearningKit_Ex3.HTML** . This is similar to what you did in Step 1 of Exercise 2 for buttons. When this checkbox is clicked, we want to call a function that we name it `checkUncheck1()`.

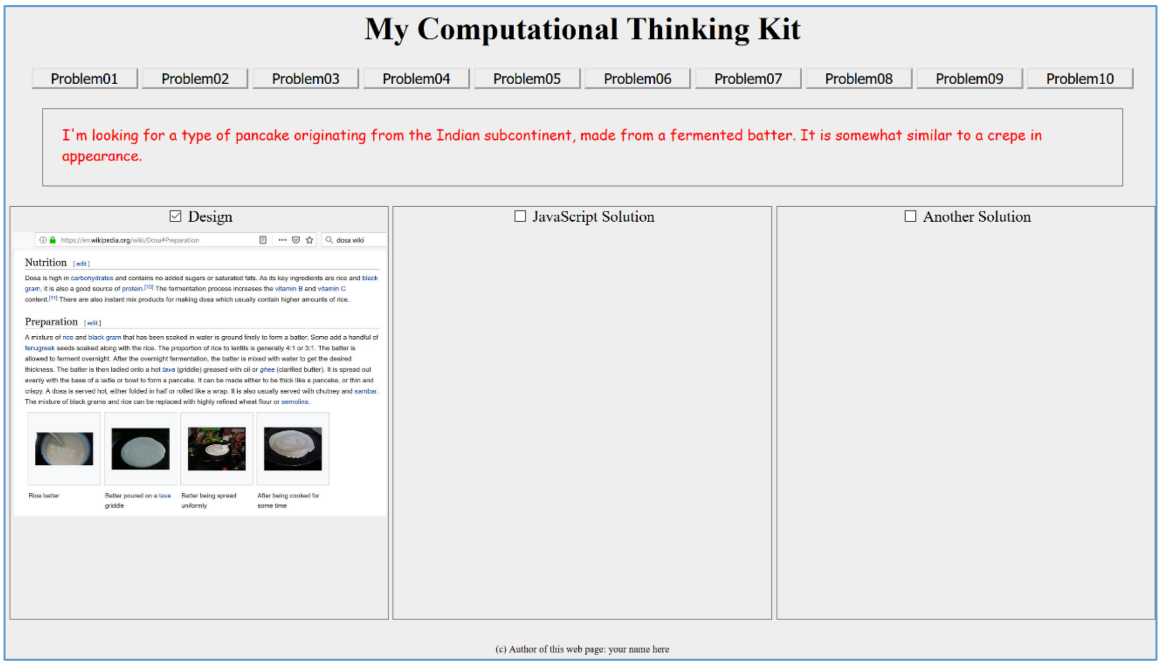
```
onclick="checkUncheck1()"
```

2) Then, we want to provide the function is **myLearningKit_Ex3.js**. Remove the two lines that we ask you to remove in the `js` file to uncomment `checkUncheck1()`. Then add each of the following two commands where appropriate.

```
document.getElementById("flowchart").style.display="inline";
document.getElementById("flowchart").style.display="none";
```

The `if` statement there determines if the image is displayed or not; if not, we display it. If yes, we disappear it. In other words, by clicking on the Design checkbox, the image's display toggles.

The following picture shows when the Design checkbox is checked.



I'm looking for a type of pancake originating from the Indian subcontinent, made from a fermented batter. It is somewhat similar to a crepe in appearance.

☑ Design

Nutrition [edit]

Dosa is high in carbohydrates and contains no added sugars or saturated fats. As its key ingredients are rice and black gram, it is also a good source of protein.[10] The fermentation process increases the vitamin B and vitamin C content.[11] There are also instant mix products for making dosa which usually contain higher amounts of rice.

Preparation [edit]

A mixture of rice and black gram that has been soaked in water is ground finely to form a batter. Some add a handful of fenugreek seeds soaked along with the rice. The proportion of rice to lentils is generally 4:1 or 5:1. The batter is allowed to ferment overnight. After the overnight fermentation, the batter is mixed with water to get the desired thickness. The batter is then ladled onto a hot *tava* (griddle) greased with oil or ghee (clarified butter). It is spread out evenly with the base of a ladle or bowl to form a pancake. It can be made either to be thick like a pancake, or thin and crispy. A dosa is served hot, either folded in half or rolled like a wrap. It is also usually served with chutney and sambar. The mixture of black grams and rice can be replaced with highly refined wheat flour or semolina.

Rice batter
Batter poured on a tava griddle
Batter being spread uniformly
After being cooked for some time

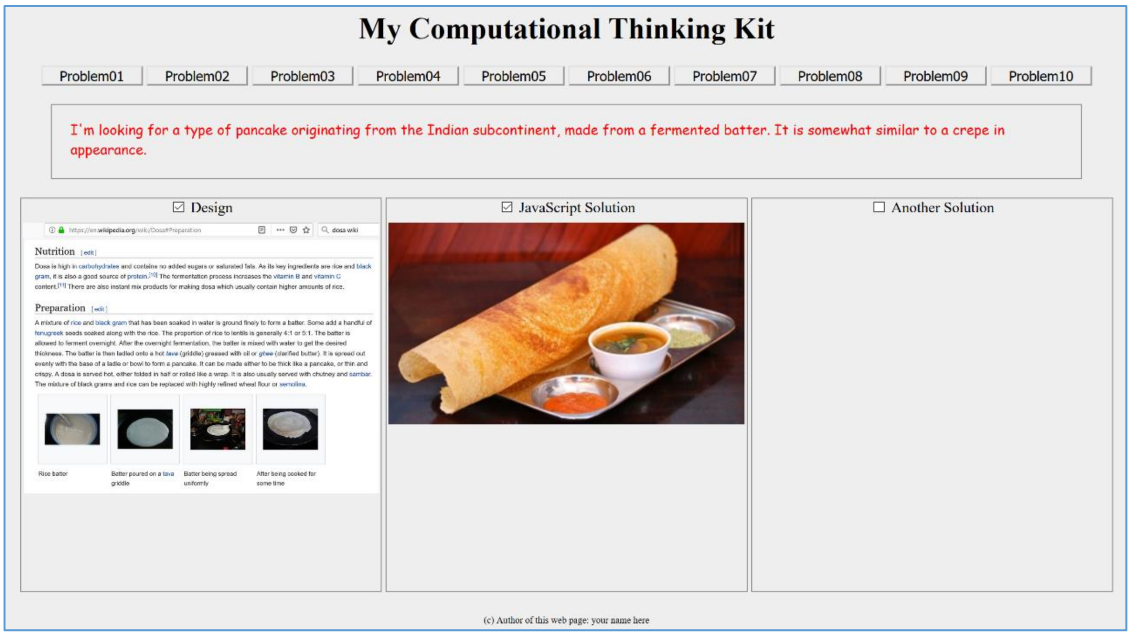
☐ JavaScript Solution

☐ Another Solution

(c) Author of this web page: your name here

3) Mimic steps 2 and 3 above for toggling the image that is displayed in the next checkbox.

The following picture shows when the next checkbox is checked too.



I'm looking for a type of pancake originating from the Indian subcontinent, made from a fermented batter. It is somewhat similar to a crepe in appearance.

Nutrition [edit]

Dosa is high in carbohydrates and contains no added sugars or saturated fats. As its key ingredients are rice and black gram, it is also a good source of protein.[10] The fermentation process increases the vitamin B and vitamin C content.[11] There are also instant mix products for making dosa which usually contain higher amounts of rice.

Preparation [edit]

A mixture of rice and black gram that has been soaked in water is ground finely to form a batter. Some add a handful of fenugreek seeds soaked along with the rice. The proportion of rice to lentils is generally 4:1 or 5:1. The batter is allowed to ferment overnight. After the overnight fermentation, the batter is mixed with water to get the desired thickness. The batter is then ladled onto a hot tava (griddle) greased with oil or ghee (clarified butter). It is spread out evenly with the base of a ladle or bowl to form a pancake. It can be made either to be thick like a pancake, or thin and crispy. A dosa is served hot, either folded in half or rolled like a wrap. It is also usually served with chutney and sambar. The mixture of black grams and rice can be replaced with highly refined wheat flour or semolina.

Rice batter

Batter poured on a tava griddle

Batter being spread uniformly

After being cooked for some time

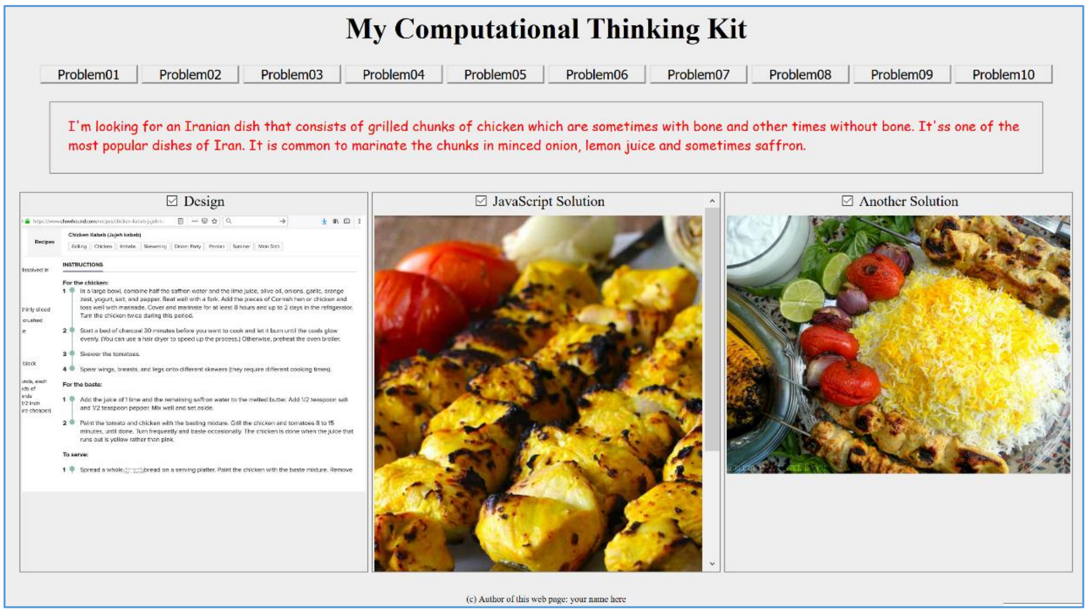
(c) Author of this web page: your name here

Exercise 4: (CREATE: **myLearningKit_Ex4.html** and **myLearningKit_Ex4.js)**
Copy **myLearningKit_Ex3.html** to a new file named **myLearningKit_Ex4.html**. Copy **myLearningKit_Ex3.js** to a new file named **myLearningKit_Ex4.js**.

Mimic what you did in Exercise 2, to assign problem description and corresponding images when button `Problem02` is clicked. Add function `p02Func()` to the `js` file. Its body is similar to that of `p01Func()`. The `innerHTML`, of the object we get, will have the following paragraph.

> *I'm looking for an Iranian dish that consists of grilled chunks of chicken which are sometimes with bone and other times without bone. It'ss one of the most popular dishes of Iran. It is common to marinate the chunks in minced onion, lemon juice and sometimes saffron.*

The images that you assign are `jujehDesign.jpg`, `jujeh1.jpg`, and `jujeh2.jpg`. Note that during this exercise, you will work on `check3` as well (if you did not do it during Step 4 of Exercise 2). The following picture illustrates when `Problem02` is clicked and all checkboxes are checked.



I'm looking for an Iranian dish that consists of grilled chunks of chicken which are sometimes with bone and other times without bone. It'ss one of the most popular dishes of Iran. It is common to marinate the chunks in minced onion, lemon juice and sometimes saffron.

Chicken Kabab (Jujeh kabab)

INSTRUCTIONS

For the chicken:

1 In a large bowl, combine half the saffron water and the lime juice, olive oil, onions, garlic, orange zest, yogurt, salt, and pepper. Beat well with a fork. Add the pieces of Cornish hen or chicken and toss well with marinade. Cover and marinate for at least 8 hours and up to 2 days in the refrigerator. Turn the chicken twice during this period.

2 Start a bed of charcoal 30 minutes before you want to cook and let it burn until the coals glow evenly. (You can use a hair dryer to speed up the process.) Otherwise, preheat the oven broiler.

3 Skewer the tomatoes.

4 Spear wings, breasts, and legs onto different skewers (they require different cooking times).

For the baste:

1 Add the juice of 1 lime and the remaining saffron water to the melted butter. Add 1/2 teaspoon salt and 1/2 teaspoon pepper. Mix well and set aside.

2 Paint the tomato and chicken with the basting mixture. Grill the chicken and tomatoes 8 to 15 minutes, until done. Turn frequently and baste occasionally. The chicken is done when the juice that runs out is yellow rather than pink.

To serve:

1 Spread a whole [illegible] bread on a serving platter. Paint the chicken with the baste mixture. Remove

(c) Author of this web page: your name here

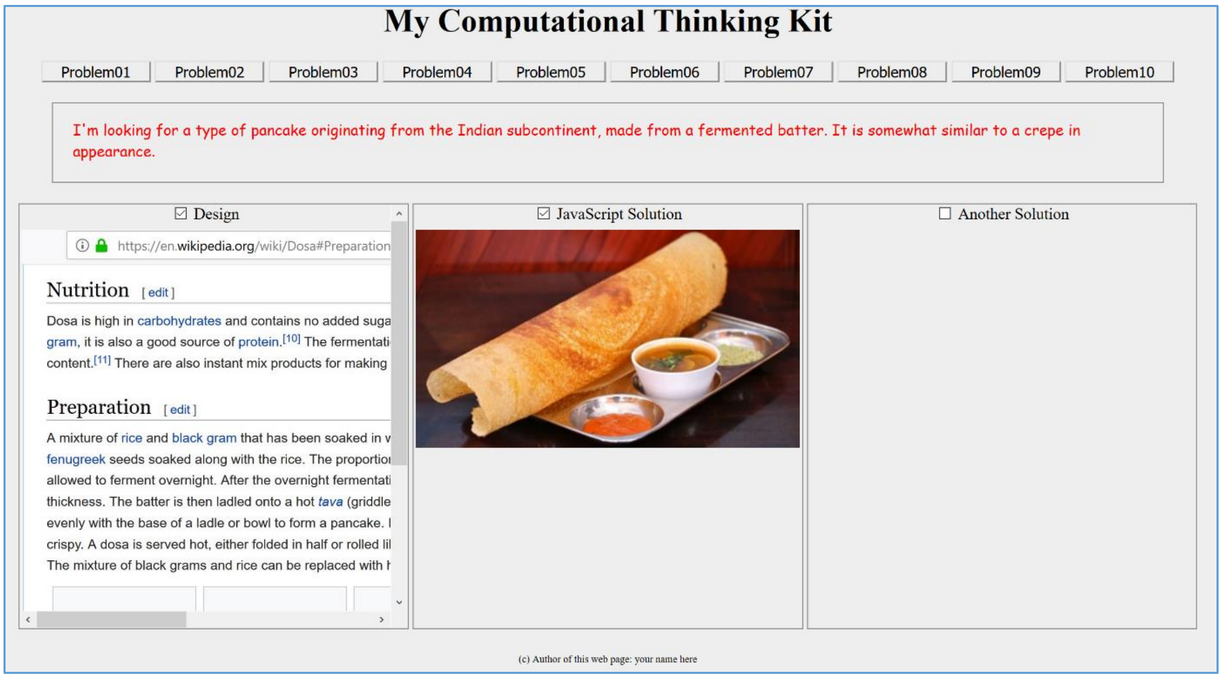
Exercise 5: (CREATE: **myLearningKit_Ex5.html** and **myLearningKit_Ex5.js)**
Copy **myLearningKit_Ex4.html** to a new file named **myLearningKit_Ex5.html**. Copy **myLearningKit_Ex4.js** to a new file named **myLearningKit_Ex5.js**.

In this exercise, we want to add two more functions to our `JavaScript` code. One we call it `zoomIn`() and the other `zoomOut`(). The objective is that when the image of the first checkbox is double clicked, we want to make its width 200% (do this in the body of `zoomIn`()) and when is single clicked, we want to have its original width of 100% (we do this in the body of `zoomOut`()).

Here, it's list of what you need to do:

1) In your HTML file, add `onclick` and `ondblclick` attributes to the image that its ID is "flowchart"
   a. on an `ondblclick` event, function `zoomIn`() is called
   b. on an `onclick` event, finction `zoomOut`() is called
2) In your `js` file, add `zoomIn`() and `zoomOut`() functions
   a. write `document.getElementById("flowchart").style.width="200%";` in the body of `zoomIn`()
   b. write a similar line in the body of `zoomOut`() to change the width to its original size

The following figure illustrates when `Problem01` is clicked and the Design image is double clicked.



My Computational Thinking Kit

I'm looking for a type of pancake originating from the Indian subcontinent, made from a fermented batter. It is somewhat similar to a crepe in appearance.

Nutrition [edit]

Dosa is high in carbohydrates and contains no added suga
gram, it is also a good source of protein.[10] The fermentati
content.[11] There are also instant mix products for making

Preparation [edit]

A mixture of rice and black gram that has been soaked in v
fenugreek seeds soaked along with the rice. The proportio
allowed to ferment overnight. After the overnight fermentati
thickness. The batter is then ladled onto a hot *tava* (griddle
evenly with the base of a ladle or bowl to form a pancake. I
crispy. A dosa is served hot, either folded in half or rolled li
The mixture of black grams and rice can be replaced with h

(c) Author of this web page: your name here

Exercise 6 (further practice, for bonus):

You may want to continue this project and add at least two more problems (aka food definition) and corresponding design (aka receipe) and implementations (aka result pictures) for 10% bonus point.

Note. In order for the project to work properly all problems should have 3 images, one for design, one for `js` solution, and the 3rd for "another solution".