Web Programming

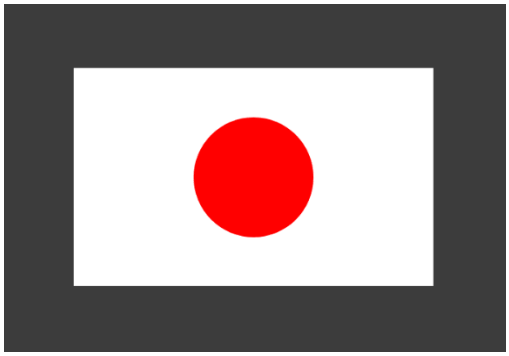
Tutorial 2

To begin this tutorial, please download `tut02-starter.zip` from the course website.

When you finish, zip all your deliveries to submit to this tutorial's submission box. The zip file's name should follow this format: `tclass_sid.zip` where `tclass` is your tutorial class name (e.g. `clc01`, `clc02`, etc.) and `sid` is your student's ID (e.g. `2101040015`).

Activity 1 – Japan Flag

(10 mins) Use HTML and CSS to create the following Japan flag on a page with dark background:



Try doing this exercise with and without Flexbox.

→ **Delivery**: `japan_flag` folder with possible HTML and CSS files.

Activity 2

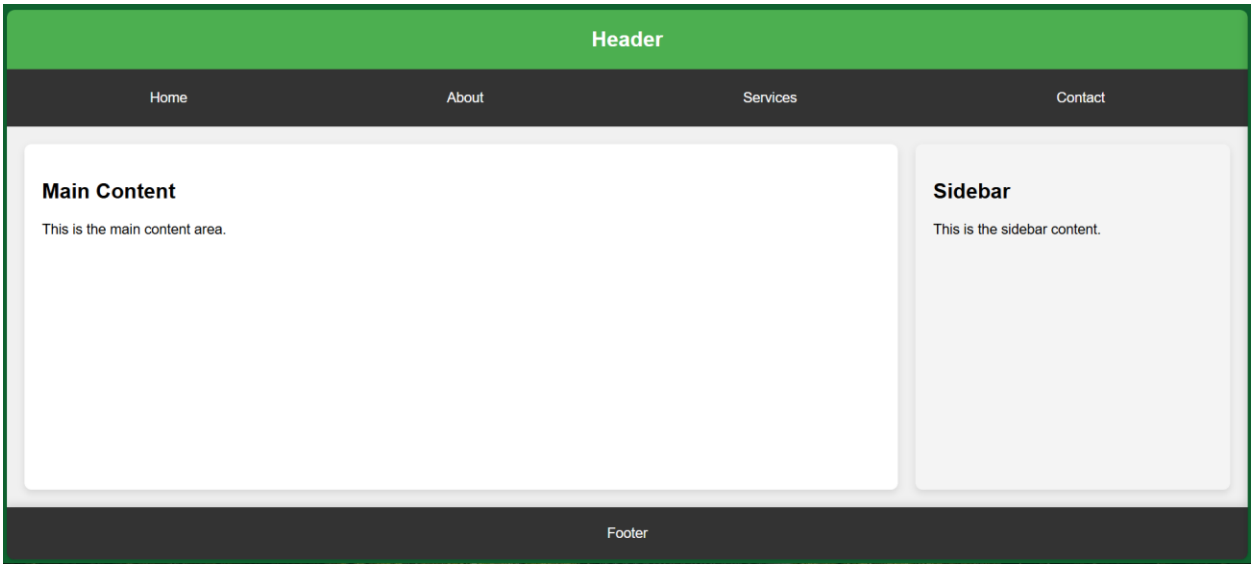
(20 mins) Use the given `card-numbers` folder as a starter pack. Use your understanding of CSS Flexbox to create the following bordered box which contains flash cards:

- The card container is represented using a div and has 4 images, each representing a number. This card should be `500px` wide and `200px` tall with a solid `#698733` border of `0.5em` width and a `border-radius` of `1em`.
- The four images in the card should take up `70%` of the height of the card, be centered vertically within, and space-evenly horizontally.
- *(Challenge)* Once you've finished this exercise to get the expected output, how can you modify your CSS to get the numbers in the order "3 2 1 0"? How about "2 3 1 0"? For the second ordering, try to use the order property.

→ **Delivery**: `card-numbers` folder with added `number.css`

Activity 3

Use the given `responsive-web-page` folder as a starter pack. Create a responsive web page using flex box that includes a header, navigation bar, main content area, sidebar, and footer. The layout should adjust for different screen sizes using media queries.



- Header: spans the full width of the page.
- Navigation bar: below the header, with horizontal navigation links.
- Main content area: includes a section for articles (content) and a sidebar for additional content.
- Footer: spans the full width of the page.
- Responsive design:

- On large screens (desktops), the sidebar should be next to the main content.
- On small screens (mobiles), the sidebar should be below the main content.

➔ **Delivery**: `responsive-web-page` folder with added `styles.css` file.

Activity 4 – Simple DOM interaction

In this activity, you'll add some simple interactivity to the given `pokeball.html`. The script `pokeball.js` has been linked in this page. In `pokeball.js`, write JS code to select all images on the page (`pokeball.html`) and set their `src` attribute into `mystery.gif`.

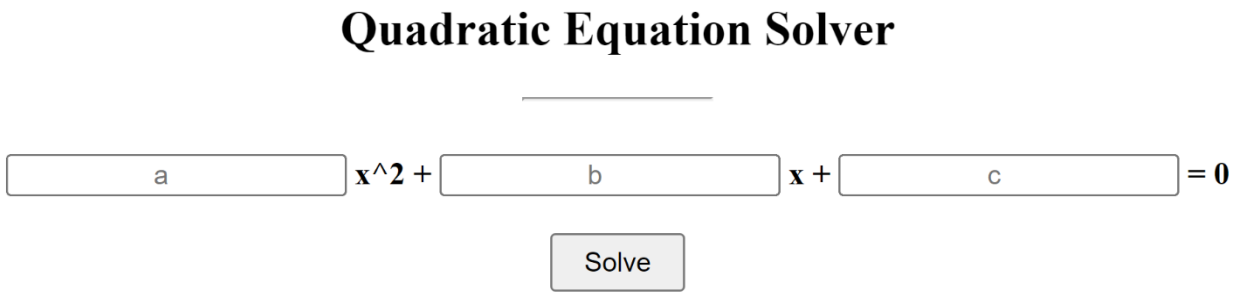
→ **Delivery**: modified `pokeball.js`

Activity 5 – Capitalize all paragraphs

Write JavaScript code in `capitalize.js` to capitalize text on all paragraphs of the webpage `capitalize.html`. DOM objects have a `textContent` property that you can access to get the text in the element, or set to change the text.

→ **Delivery**: modified `capitalize.js`

Activity 6 – Quadratic Equation Solver



Inside the `equations` directory there are `index.html` and `equations.js`, which is already embedded into `index.html`. Your job is to modify `equations.js` to add functionality to `index.html` (and leave `index.html` unchanged) so that a user can solve quadratic equations on it.

→ **Delivery**: modified `equations.js`