

Assignment 1: Personality Quiz

IMPORTANT: *This assignment must be done individually.*

Read Section A to understand the programming requirements, Section B to understand the programming tasks that you need to carry out and Section C to know what you need to submit as the result.

A. Description

In this assignment, you will create a personality quiz called “**Adopt A Dog And We’ll Reveal A Deep Truth About You.**”

1. Overall appearance and behavior

The following video shows the look and behavior of the quiz we are asking you to implement.

<https://drive.google.com/file/d/19grmmyMi2g0bEQDcw9upoHUpyasfrHnB/view>

One detail that is hard to see in the video:

- After the quiz is completed (i.e. after all three questions have been answered), it should *not* be possible to change your answer anymore.

Section measurements

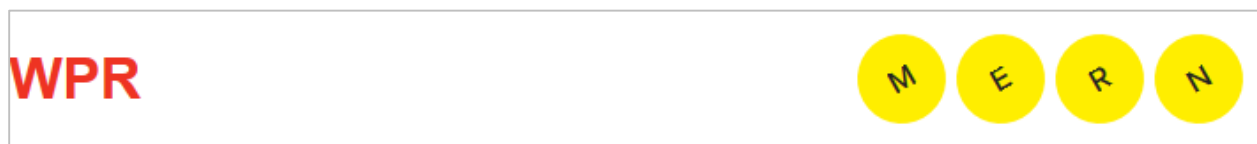


Font faces, size, and colors:

Common

- The font face is `'Proxima Nova'`, and the fallback fonts are, in order, `'Helvetica'`, `'Arial'`, `sans-serif`.
- The font size is `18px`.
- The font color is `#222222`

Header



- It has a `#ccc` border that is `1px` thick.

Course name

- The font face is 'Helvetica' and the fallback font is sans-serif.
- The font size is 32px.
- The font weight is bold.
- The font color is #ee3322.

MERN Stack (Yellow circles)

- The background color is #ffee00
- They have rounded corners with radius 100%.
- The font weight is bold.
- The text is vertical and center aligned to the circle.
- They are rotated 30 degrees to the left. **Hint:** see css properties transform [w3schools](https://www.w3schools.com/css/default.asp)

Page title

Adopt A Dog And We'll Reveal A
Deep Truth About You

- The font size is 42px
- The font weight is bold.

Author

By Cong Nguyen

- The font size is 14px
- **Note:** You have to replace “Cong Nguyen” by *your name* as the author of your assignment.

Question name

Pick a pup:

- The back ground color is semi-transparent, which is rgba(255, 255, 255, 0.9)
- The font face is 'Pangolin', and the fallback fonts are 'Trebuchet MS', cursive
- The font size is 60px
- The font color is black

- Space between the border and the content of the element is 10px.
- The question name is vertical and center aligned to the question image.

Images

- Question image
 - The header is a background image (images/background.gif).
 - Its height is 425px.

2. Choice grid

You need to finish styling the answer choice grids.

Choice grid



- There is 20px of space between each row
- Each flex item has a width of 32.5%, and each item is spread out with as much space between the other as there can be in the flex container.

- **Hint:** You may consider to use a **multi-row flexbox** to achieve this layout.
 - We didn't talk about multi-row flexbox layout in class, but you can allow flex items to wrap onto the next line (i.e. the second row) by setting `flex-wrap: wrap;` onto the flex container.

Choice: unanswered

- Container
 - Background color is `#f4f4f4`
 - Border is 1px thick and color is `#dcdcdc`
 - Width of the element **including** the border is `32.5%`
 - **Hint:** The `box-sizing` CSS property might be helpful.
 - Space between the border and the content of the element is `10px`.
- Choice image (i.e. dog, house, or toy image)
 - **Hints:** We did not tell you the width or height of the image intentionally, as we want you figure out how to get it to behave like this. There are many ways to get the images to “shrink” inside its container
 - The beverage image of `tut02` also had similar behavior. Recall what dimensions we gave you for the beverage image.
- Checkbox
 - The unchecked image is `images/unchecked.png`
 - The height and width is of the image is `20px`
 - **Note:** Do not use an `<input type="checkbox" />` for this. Just display the image.

3. Mobile layout

You need to also modify the CSS and HTML if necessary to implement support a mobile view. The video below shows an example of how it should look and behave:

<https://drive.google.com/open?id=15zypGtz9KB3NKvG6i3CtltOYTBHtsX4>

- If the page is viewed on mobile:
 - The viewport should be set to zoom-level `100%`, and the width should be the device width
- If the device screen size is less than `700px` wide:
 - The width of the page content should be `95%` instead of `700px`
- If the device screen size is less than `500px` wide:

- Each answer choice should be 49% in width instead of 32.5%
- The yellow circles in the page header should not appear

4. Quiz behavior

This web page is a 3-question personality quiz. Write the code necessary to implement the quiz behavior as detailed below.

Even though you will mostly be writing JavaScript for this part, you may need to also update the HTML or CSS in order to implement the behavior as described.

Dataset attributes

Note that you should take advantage of the [dataset attributes](#) added to the HTML elements in `index.html`:

- `data-choice-id`: (eg. blep, happy...) Maps to which quiz result the choice should “count” for, as defined in `constants.js`. See details below for scoring and what to do if there are ties.
- `data-question-id`: Maps to the question number: one, two, and three.

You can access these attributes in JavaScript by using:

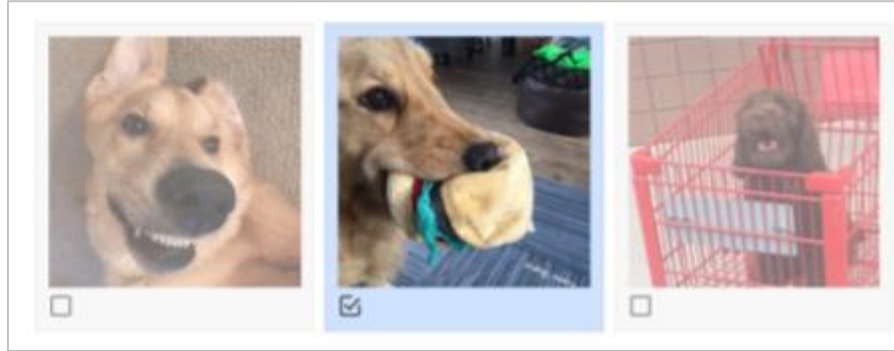
- `element.dataset.choiceId`
- `element.dataset.questionId`

And you can select these attributes via CSS selectors like the following:

- `[data-choice-id='blep']`
- `[data-question-id='two']`

Clicking an answer

When the user clicks an answer choice, the answer choices should update in the following way:



- For the selected item:
 - The checkbox image should change from unchecked to checked
 - Background color is `#cfe3ff`
 - The checked image is `images/checked.png`
- For the unchosen items:
 - They should be made semi-transparent by setting their `opacity: 0.6;`
 - Note that only the unchosen items *for this question* should change style.

Changing an answer

If the user has not completed the quiz (i.e. there is at least one unanswered question), they should be able to change their answer to a question by clicking a different answer.

After the user has answered every question, the answer choices should lock and it should no longer be possible to change an answer, until the user clicks “Restart quiz” or refreshes the page.

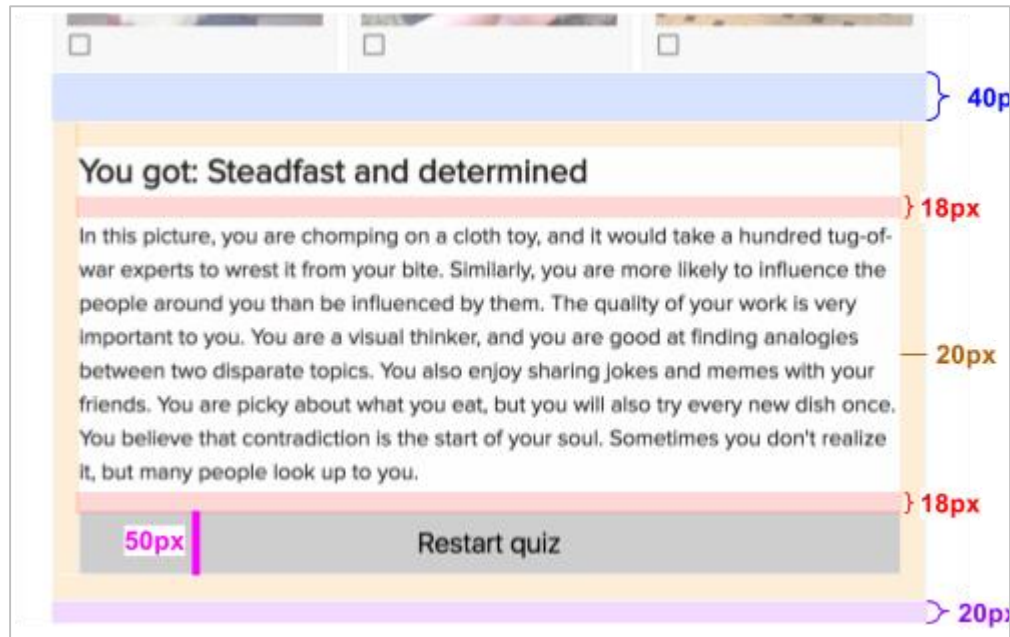
Completing the quiz

You should check every time after the user clicked any answer. If the user has answered all three questions, the quiz is complete.

- It should not be possible to select another answer after you’ve completed the quiz.
 - i.e. you click another dog after the quiz is complete, nothing on the page should change.
- The page should remain in this “completed” state until the user refreshes the page or clicks “Restart quiz”.

The personality results should appear at the bottom of the page, with the personality data that corresponds to the `dog` (eg. blep, happy...), as defined by `constants.js`.

Here is the description of how the personality results should look:



- Result button
 - Background color is `#cecece`
 - On hover, the color should change to `#e0e0e0`



Scoring the quiz

The `data-choice-id` for each answer choice maps to the key name for each possible result in `RESULTS_MAP`, stored in `constants.js`. You can access `RESULTS_MAP` in `script.js` because `constants.js` is included before `script.js` in `index.html`.

When the quiz is complete, you can score the quiz by tallying the `data-choice-ids` from each answer. For example, if a user chooses `blepdog.jpg`, `sleepybed.jpg`, and `bleptoy.jpg`, you should show the title and contents from `RESULTS_MAP['blep']`.

If there is a tie, i.e. if someone chooses all unique `data-choice-ids`, the answer to the first question should win. For example, if a user chooses `burgerdog.jpg`, `nerdbed.jpg`, and `shydog.jpg`, you should show the title and contents from `RESULTS_MAP['burger']`.

Resetting the quiz

If the user clicks the “Restart quiz” button, the page should reset to its original state.

- The answer choices should return to their original appearance before the quiz was complete
- The personality results should disappear
- The answer choices should be selectable again, and clicking on a different set of answers should result in showing an updated personality type.
- The page should look and behave the same as if you had refreshed the page (but you should not actually refresh the page)
- You should also make the page scroll to the top of the “Pick a pup” element.
 - You can call `element.scrollIntoView();` to do this. See [mdn](#) for more details.

`constants.js`

- This file contains the definition of the object `RESULTS_MAP`, which maps the dog type to its result title and description.
- Because this variable is declared in the global scope, you can access this variable in `script.js`.
- You should **not** have to modify this file to solve the assignment.

`script.js`

- This is the file in which we expect you to implement the quiz behavior.
- You should define and attach event listeners in this file.

Image to personality mapping

The `RESULTS_MAP` is defined in `constants.js` and the images are saved in the `images/` directory.

- `blep{dog,bed,toy}.jpg` corresponds to `blep` in `RESULTS_MAP`
- `happy{dog,bed,toy}.jpg` corresponds to `happy` in `RESULTS_MAP`
- `sleeping{dog,bed,toy}.jpg` corresponds to `sleeping` in `RESULTS_MAP`
- `dopey{dog,bed,toy}.jpg` corresponds to `dopey` in `RESULTS_MAP`
- `burger{dog,bed,toy}.jpg` corresponds to `burger` in `RESULTS_MAP`
- `cart{dog,bed,toy}.jpg` corresponds to `cart` in `RESULTS_MAP`

- `nerd{dog,bed,toy}.jpg` corresponds to `nerd` in RESULTS_MAP
- `shy{dog,bed,toy}.jpg` corresponds to `shy` in RESULTS_MAP
- `sleepy{dog,bed,toy}.jpg` corresponds to `sleepy` in RESULTS_MAP

NOTE: To make this easier to test, the items for each question are in the order displayed above, i.e. the first answer of every question maps to `blep`, the second answer of every question maps to `happy`, etc. **You should not rely on this ordering for your assignment solution.** Instead, use the dataset attributes as described at the top of this section.

B. Task requirements

1. Download the *starter_pack* for folder structure suggested to you. We expect you will have to make modifications to the following files to complete the assignment:
 - `index.html`: Write your HTML here (<head> section has been *partially* completed for you)
 - `style.css`: Write your CSS here (pay attention about the mobile layout)
 - `script.js`: Write your JavaScript here.
2. HTML: Copy and paste the text of *index-content.txt* into your *index.html* file, then add the HTML tags necessary to style the page. Later you will likely need to make some modifications for the mobile layout and the JavaScript.

NOTE: To make this faster and easier to test later, the items for each question are in the order of `blep{dog,bed,toy}.jpg`, `happy{dog,bed,toy}.jpg` ... (as mentioned in section *Image to personality mapping*)

3. CSS: in the `style.css` file,
 - Add appropriate CSS to style your page as described in section **B.2**
 - Add appropriate viewport & CSS media queries to support the mobile view. (see section **B.3**)
4. JavaScript: in the `script.js` file, to accomplish quiz behavior as described in section **B.4**
 - Add data attributes.
 - Handle user action: clicking an answer
 - Handle user action: changing an answer

- Check if user complete the quiz, score them and display result
- Handle user action: resetting the quiz

5. Weekly plan:

You have to schedule yourselves a weekly plan to complete your tasks, an example is given below.

Week	HTML	CSS	JS
1	- header - quiz title, author - question 1: + question name + choice grid & choice - copy x2 question 1 & replace with appropriate content for question 2 & 3		
2	- result view	- mobile layout - hide result view	- add data attributes - handle user click an answer / change answer
3			- check if quiz completed - score with RESULT_MAP & display - reset the quiz
	- test your application (mobile layout, correct result corresponding to clicked items, reset quiz)		

C. Submission

You must submit a single zip file containing the application to the portal by the due date. The zip file name must be of the form `a1_Sid.zip`, where *Sid* is your student identifier (the remaining bits of the file name must not be changed!). For example, if your student id is 1701040001 then your zip file must be named `a1_1701040001.zip`.

IMPORTANT: failure to name the file as shown will result in no marks being given!

NO PLAGIARISM: if plagiarism is detected, 0 mark will be given!