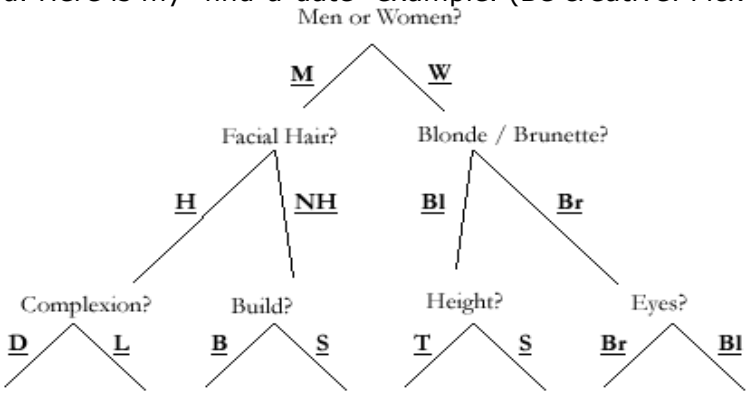
Project 1: Interactive Form Elements

Overview

For this project, you will make a web page that contains a form using *select* pull-down elements. Once the user selects the desired option, your script will dynamically create another pull-down element whose contents will depend upon the visitor's first choice…

Overall Requirements

- Dynamically creating form elements depending upon the answer of the last question
  - The form elements for the questionnaire MUST be <select><option> elements
- At least 2 choices for each selection
- At least a depth of 3 different questions (if you make yours so that it can create 'x' number of dropdowns, then the depth can be 2 for some and 4 or 5 for others!)
- Topic is up to you. Here is my "find-a-date" example. (Be creative! Pick an interest.)



Finally, Print out choices to the screen!

- Once the user has made their selections, create, at least, a new node to print out their final choices to the screen (e.g.: You want a man with facial hair of slight build! That is minimal – do more to do better!)
- Give the user the option to start over at any time, and restart once done!
- A form on some page getting the users information, with validation.
- Use local Storage. Hold user information and use the stored information.
  - You could have the user enter their name, place it in local storage and some time later retrieve it and format and present with the results of their selections.
- Looks!!! Expect professional looking, quality graphics, usability, etc.
- All displayed questions, select choices, and extras are to be stored in data structure(s) stored in a *.js file. No displayed information should appear in the JavaScript code.
  - In another words, if you swap out your data *.js file for another similarly structured data *.js file, your program will still work, display questions, choices, and other similar displays to the user. Create a separate file for your data, json, XML, etc.
- Name your main page index.html
- Last requirement is:
  Do NOT post this on the internet such as people.rit.edu or any other internet location.
  If your program will only work being placed on the internet, the program is wrong.

Suggestion of How to Start

Start (don't finish) with the first option being coded in index.html. When the user makes a choice, another option menu dynamically appears that contains choices that are directly related to the answer of the first choice. Once the user makes a second choice, the third, or more, option menu will appear with new choices that are dependent upon the second choice (and, of course, on the first choice…). Once the user has made at least 3 different choices, you will create a new text node and dynamically display their choices out to the screen.

To Finish

After getting the first hardcoded select to work – you need to make the first option/select choice NOT be hard coded in html. Make the first select work dynamically in JavaScript.
At the end, you should have one function to create all of the dropdowns, the content of the select/option should be dependent upon what information is sent to the function.
Much will become clearer after the DOM lecture.

Programming requirements

- Dynamic creation of at least 3 sets of <select><option>value… nodes (tags)
- Constructor method that creates the next select menu from the previous choice
- Use of the following functions, but not limited to:
  - createElement, setAttribute, appendChild, getElementById, getElementsByTagName, createTextNode, nodeValue to name a few…
- Do NOT use visibility to show/hide the selections – everything is dynamically created!
- Some use of DHTML (menu system, sliding elements, animation preferred!) Comment!
- No layout with tables! Everything done with CSS
- No use of innerHTML, innerText, etc. ! (I might allow it for the practical, but not for the projects)
- Extensive comments in the code to explain functionality. More than function headers
- Once 3+ select options are showing, changing the 2nd or 1st will *remove all those selections below the changed one*, except for the next item allowing a new selection!!

Grading

An excellent project (worthy of an A) will have everything I'm asking for here, *plus something extra* - evidence that you intend to be exceptional. Perhaps it explores an area we didn't cover in detail in class, or would merit me recommending you to a client who needed similar work done on a multimedia project.

To do better…

- Good, reusable code – data from an AJAX call or even a local xml document
- Scalability (is your code generic enough to have more than 3 choices if the data changes? Did you prove scalability by providing 2 data set files that both work?
- Extend Functionality
- High Level of Technical Development
- Clearly document in your code the *exceptional* parts for exceptional grading level
- To ensure you get the grade you work so hard to receive, include a **README.txt** that lists the work and extras you added. Include **DHTML item**, what is validated, etc.
- Surprise me…

- **DUE DATE: September 26, week 6 at midnight, see drop box in myCourses for date. Zip the entire project (file structure intact!) in a folder with your last name.zip.**

Client Programming – Project 1 – Grade sheet

Basic requirements (½ to 1 letter grade per item)

- ☐ Dynamically creating form elements depending upon the answer of the previous question, no innerHTML, innerText, outerText, textContent, text, etc. (Project not accepted/graded)
  - New selection shows from making a previous select. No need to click a button.
- ☐ Minimal HTML. No <select> in HTML, etc.
- ☐ Data separation from code. ANY data appearing in code -1 or more letter grades!
- ☐ Separate file for data, formatted. Allows for changing data files and program still works
- ☐ At least 2 choices for each selection
- ☐ At least a depth of 3 different questions (if you make yours so that it can create x number of dropdowns, then the depth can be 2 for some and 4 or 5 for others!)
- ☐ Once the user has made all their selections, create a new node to print out their final choices to the screen (Ex: You want a man with facial hair of slight build! – or could use information to open a new window)
- ☐ After final node/screen, ability to start new choices, by changing first select vs. button
- ☐ Give the user the option to start over at any time by simply alter any answer
- ☐ Graphics / Look. Everything done with your own CSS. No layout with tables or bootstrap
- ☐ Use of DHTML in JavaScript. Animations preferred! CSS animate does not count.
- ☐ Do NOT use visibility to show/hide the selections – I want them dynamically created!
- ☐ A form on some page getting the users information (Validation)
- ☐ The use of localStorage

- ☐ Include comments in the code to clearly explain functionality. More than functional level
- ☐ Can delete cookies & localStorage to restart the program. (use a button, menu, etc.)
- ☐ No use of innerHTML, innerText, etc.
- ☐ Loss of one letter grade per day late

Major grade problems:

- ☐ No separation of code in files: JS, CSS, HTML, data. (-½ to -3 letter grades)

Other considerations:

- ☐ Put some thought into the choices, not a random set of choices such as 'write your own story' that has no story
- ☐ Put your name at the beginning of all files: JS, CSS, HTML, etc. Change <title> to your name/project name
- ☐ README.txt – Text only file, not RTF. What you added to move your grade above a "B", include exactly what DHTML was used.