Net-centric introduction to computing

CV

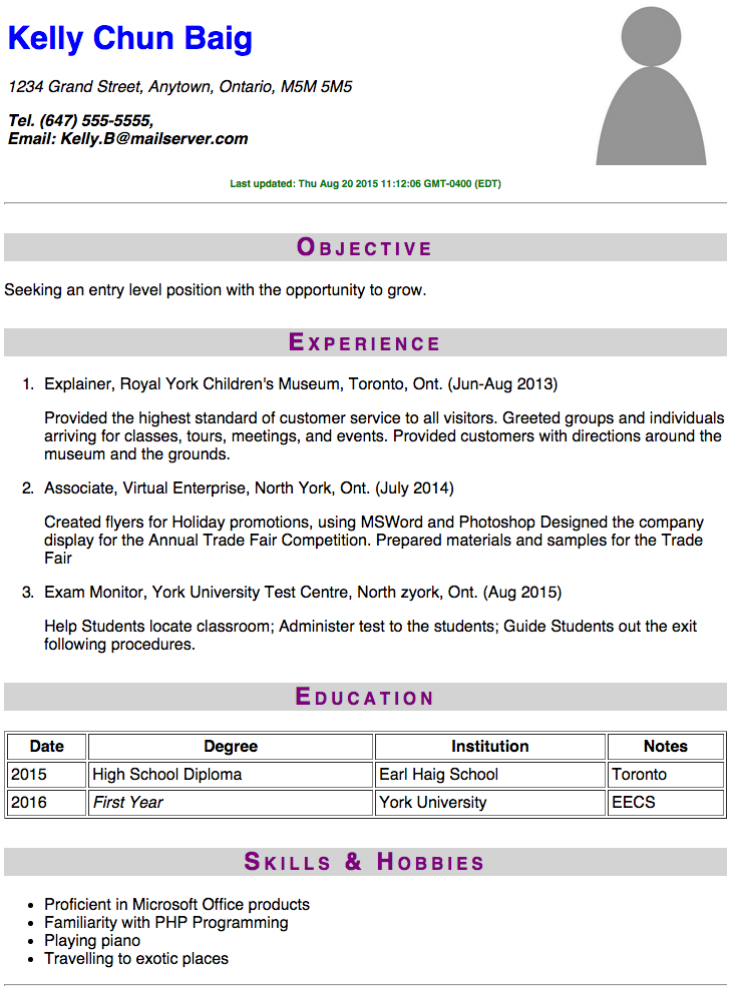
m Jenkin + h Roumani

Table of Contents

1. Introduction

A ***CV*** (an acronym for **C**urriculum **V**itae) is a document that describes your skills, experience, education, achievements, and other qualifications. It is often used, together with a cover letter, when you apply for a job so that the employer can quickly screen applicants and select those eligible for an interview. A ***resume*** is a short version of the CV (typically one or two pages) but we will use the two terms in this lab as synonymous.

In this lab, you are going to create a CV for yourself and store it online. Unlike a static paper CV, this Web CV is very easy to change (content and formatting wise) and can be made interactive (i.e. has behaviour that responds to events). The final CV that you will create will look like this:



Kelly Chun Baig

1234 Grand Street, Anytown, Ontario, M5M 5M5

Tel. (647) 555-5555,
Email: Kelly.B@mailserver.com

Last updated: Thu Aug 20 2015 11:12:06 GMT-0400 (EDT)

OBJECTIVE

Seeking an entry level position with the opportunity to grow.

EXPERIENCE

1. Explainer, Royal York Children's Museum, Toronto, Ont. (Jun-Aug 2013)

   Provided the highest standard of customer service to all visitors. Greeted groups and individuals arriving for classes, tours, meetings, and events. Provided customers with directions around the museum and the grounds.

2. Associate, Virtual Enterprise, North York, Ont. (July 2014)

   Created flyers for Holiday promotions, using MSWord and Photoshop Designed the company display for the Annual Trade Fair Competition. Prepared materials and samples for the Trade Fair

3. Exam Monitor, York University Test Centre, North zyork, Ont. (Aug 2015)

   Help Students locate classroom; Administer test to the students; Guide Students out the exit following procedures.

EDUCATION

Date	Degree	Institution	Notes
2015	High School Diploma	Earl Haig School	Toronto
2016	*First Year*	York University	EECS

SKILLS & HOBBIES

- Proficient in Microsoft Office products
- Familiarity with PHP Programming
- Playing piano
- Travelling to exotic places

In order to create this CV, you will need the following:

- Raw Data that pertains to you.
- HTML tags to present the raw data.
- CSS style rules to format the presentation.
- JavaScript code to add behaviour (interactivity).

The first item is for you to prepare. The remaining three are web technologies that you will learn. The Background section of this document sheds some light on the needed tools.

As with all labs in this course, each lab is laid out as an ePub. (This is the document that you are reading now.) It is expected that you will have read the ePub prior to attending the lab. In order to ensure this, each lab has an associated set of 'pre lab' assignments that **must** be completed prior to attending the lab. Your lab TA will not allow you to participate in the lab if you cannot demonstrate that you have completed the pre-lab assignment prior to attending the lab itself. Nor will you receive credit for the lab if you do not complete the pre lab assignment prior to your lab. Exercises in the lab are to be documented in your ePortfolio lab book.

2. Background

You may want to skim through the background material in your first review of this lab. Afterwards, go to the actual exercises associated with the lab and then refer back to the material presented here as needed.

2.1 HTML

The presentation layer of your raw data is done using HTML tags. For example, the experience section of the CV uses markup similar to the one shown below:

```
<h2>Experience</h2>
  <ol>
    <li>Explainer, Royal York Children's Museum</li>
    <li>Associate, Virtual Enterprise</li>
  </ol>
```

As you can see, this section is tagged as a second-level header using the <*h2*> tag. It presents the various jobs that you have held using an **o**rdered **l**ist of items, or the <*ol*> tag. Each list item is shown using the <*li*> tag. Download the sample *preCV* file from the ***JR*** site and familiarize yourself with the used tags.

2.2 CSS

In order to format the CV the way we want, we apply style rules. For example, if we want the content of the first-level header to be rendered in red, we supply this style rule:

```
h1
{
  color: red;
}
```

The *PreCV* file that you downloaded earlier contains several style rules at its top sandwiched in a style tag. Examine these rules and compare them with how the file

is rendered in the browser.

2.3 JavaScript & DOM

JavaScript is a programming language that is executed by the browser that renders the HTML document. As such, it is capable of monitoring how the user interacts with the document (e.g. by clicking buttons or swiping down) and can modify the document's content, presentation, and formatting accordingly.

A simple example of such interactivity exists in our project because we need to add a line to the CV to show the current date and time (when the document is viewed). To achieve this, JavaScript uses the DOM API to navigate to a particular tag in the document where the change is to be made. In our project, we need to change the content of the <*span*> tag whose ID is "now". As such, we use the following statement:

```
var now = document.getElementById("now");
```

The variable "now" represents the HTML <*span*> tag in the world of JavaScript and we can use it to modify the text sandwiched in the tag. For example, if you want the text: *"This is my Resume!"* to appear in it, you would add a second line of code like this:

```
now.innerHTML = "This is my Resume!";
```

2.4 Separating the Concerns

Putting HTML, CSS, and JavaScript in one document is not a good idea for a variety of reasons. It mixes levels of abstractions, impedes teamwork and reusability. Because of this, the professional approach separates these into three files. In the context of our CV project, the three files would be named as follows:

- ***index.html***
- ***CV.css***
- ***CV.js***

The HTML file will *always* be named "index.html" because this facilitates its transfer to the Android tablet. The other two are typically named after the project name.

To separate a *CV.html* file into three files, do this:

1. Cut all the style rules (sandwiched in the <*style*> tag in *CV.html*) and paste them in *CV.css*. Discard the opening and closing <*style*> tags.
2. Cut all the code lines (sandwiched in the <*script*> tag in *CV.html*) and paste them in *CV.js*. Discard the opening and closing <*script*> tags.
3. Replace the <*body*> tag in index.html with <*body onload="start();"*>.
4. Insert the following lines in the <*head*> tag on *index.html* (e.g. after the *title* tag):

```
<script type="text/javascript" src="CV.js"></script>
<link rel="stylesheet" href="CV.css" charset="utf-8"/>
```

After these changes, the top of *index.html* would look like this:

```
<html>
  <head>
    <title>EECS1012 CV</title>
    <script type="text/javascript" src="CV.js"></script>
    <link rel="stylesheet" href="CV.css" charset="utf-8"/>
  </head>
  <body onload="start();">
```

3. Exercises

Each lab in this course must be properly documented using the e-Portfolio site provided. Rich media content (e.g., videos) should be converted for publication using YouTube, Vimeo or some similar online video system. You should be aware that all material you publish on these sites will be visible to a large number of people, including people outside of the class and the university. Inappropriate publication can result in penalties beyond those associated with your academic record.

Lab exercises are broken down into three sections, ***pre-lab***, ***in-lab*** and ***advanced***. pre-lab exercises must be completed ***before*** entering the lab. As you might imagine, in-lab exercises are those that it is anticipated that you will complete in your normally scheduled lab. advanced lab exercises are those that go beyond what is normally expected in the course.

3A. Pre-lab

A Moodle quiz associated with the pre-lab must be completed before attending your lab session. **The lab TA will *not* allow you to participate in the lab without passing this quiz prior to the lab.** Furthermore, you will not receive a grade for the lab without completing the pre-lab quiz prior to your lab.

3B. In-lab

1. You will need a **laptop** and an Android **tablet** for this lab:

    1.a) You should ideally bring your own laptop (with VBox installed on it) because this makes everything easier. If not, you can borrow a laptop from the lab TA and use it during your lab session, but in this case, you must arrange to have your work saved somewhere (on a USB memory stick or in the cloud) before returning the laptop (see ***LabTools***).

1.b) For the tablet, borrow one from the lab TA and use it during the lab session. If you prefer to use your own Android device then make sure it is set up to accept USB debugging and apps from unknown sources (see ***LabTools***).

2. Start VBox and launch ***Atom*** from the *Applications*, *Programming* sub-menu.

3. Atom starts with several welcoming tabs open. Make sure you close all of them. To close a tab, click its title first (in order to give it the focus) and then click the ✖ symbol to the right of the title.

4. In Atom's file menu, select *Open Folder* and navigate to the *preCV* folder by clicking *user* in the left pane and then double-clicking *server*. Click the *preCV* folder and then OK.

5. The left pane of Atom will display all files in the opened folder. Click the *index.html* file. Its content will be shown in a tab. This should be the only opened tab.

6. The file contains a template that you can use to create your own CV. Spend a few minutes to examine its content. It starts with some style rules (in CSS syntax) and this is followed by the main body (in HTML syntax). The file ends with some behavioural code (in JavaScript syntax).

7. To better understand each piece of the content and how it contributes to the rendered CV, activate the preview by clicking **Ctrl-Shift-H**. Make sure the tab has the focus before pressing this key combination (i.e. click anywhere in the tab's content first). You can also activate the preview by selecting the *Preview HTML* submenu from the *Packages* menu.

8. The preview appears in a separate tab and it shows the CV as if it has been opened by a browser. The presence of these two tabs next to each other makes it easy to learn, and experiment with, web technologies because you can make changes in one and see the result in the other.

9. The name in the sample resume appears at the top in red. Change it to your own name and make it blue. You do this by modifying the colour in the *h1* style rule that appears near the top of the file and by modifying the context of the *<h1>* tag that appears further down in the file.

10. Modify the entire file so that:

    10.a) Its data become yours, including the picture. If you prefer to anonymize the address and the contact information then that is fine but do include information about you that would be relevant to a prospective employer.

    10.b) The CV should look exactly like the one shown in the *Introduction* section of this handbook. This includes colours, fonts, styles, layout, and spacing.

    10.c) The behavioural aspect of the CV is captured by the "Last updated" line that appears in green at the top. Add the needed code to achieve this.

11. Once your CV is complete, view it from your Android device. To do that, launch the *Html2Apk* app in your tablet; select *Load* from its menu; and enter the URL: *http://ip:8000/zip/preCV* (where ip is the IP address of your laptop). See the *Laptop* section in this document to learn about finding the IP, and see the *Tablet* section for details on the needed setup.

12. Back in Atom, select *Add Project Folder* from the file menu and navigate to the *CV* folder; i.e. double-click *user* then *server* and then CV. You will now see both the preCV and CV folders in the left pane and can access the files in each.

13. Transform your single-file preCV project to the 3-file, professional structure by transferring the content that you created in index.html in preCV to the three files in CV: index.html, CV.css, and CV.js. See *Separating the Concerns* section in this handbook to understand what goes where.

14. Verify the new CV project by viewing the CV in three different ways:

    14.a) View it in Atom by giving the focus to its index.html tab and activating the HTML preview.

    14.b) View it in the laptop browser by launching *FireFox* from the *Applications*, *Internet* menu and visiting the URL:
    `file:///home/user/server/CV/index.html`

    14.c) View it from your Android device.

15. Create a video that showcases your work and capture it in your e-Portfolio. The video should explain what you learned in this lab about HTML, CSS, and JavaScript in the context of creating a CV.

C. Advanced

1. Take a second picture of yourself and store it in the CV folder. You can do this, for example, by using your phone to take the picture and then transferring it to the laptop through the cloud.
2. Modify the CV's behaviour so that if one clicks on the picture then it will be replaced with the second picture. Here are the needed steps assuming the two picture files are *pic1.jpg* and *pic2.jpg* (replace with the actual filenames):

   2.a) Add an attribute to the *img* tag of the picture:

   ```
   <img onclick="swap(); " …
   ```

   This attribute instructs the browser to invoke the *swap* function whenever the image is clicked.

   2.b) Add the swap function to your *CV.js* file:

   ```
   function swap()
   {
      var picture = document.getElement …
      picture.src = "pic2.jpg";
   }
   ```

   2.c) Complete the above code and test your project.
3. Modify the code again so that one can toggle between the two pictures by clicking on the picture. To that end, your code needs to first determine which picture is currently displayed. If it is the first, replace it with the second, and if it is the second then replace it with the first.

   Hint: Your code must use an if statement to determine if a string x contains a string y. Here is a rough sketch:

   ```
   if (x.indexOf(y) != -1)
   {
      // x does contain y
   }
   else
   {
   ```

```
        // x does not contain y
    }
```

4. Amend your e-Portfolio and include the new feature of your project and how it was implemented.

4. Further Reading

- Tutorials on web technologies.
- A Javascript tutorial.

5. Credits


WEB COMPUTING
EXTECH
D6
VR2
Copyright © 2015 by:
m Jenkin + h Roumani