

# Web Development From Scratch

# What is this course about?

- Learn basic HTML5, CSS and Javascript from scratch
- No pre-requisite necessary
- Won't teach you everything there is about Web Dev, but enough so you can explore on your own after the course

# 1. HTML5

## What is HTML5?

- HTML5 is the latest version of **HyperText Markup Language**, or HTML
- It is *the language of the Web*
- It describes *the structure and content* of web pages
- It is used together with CSS (**Cascading StyleSheet**) and Javascript to make web pages rich and interactive

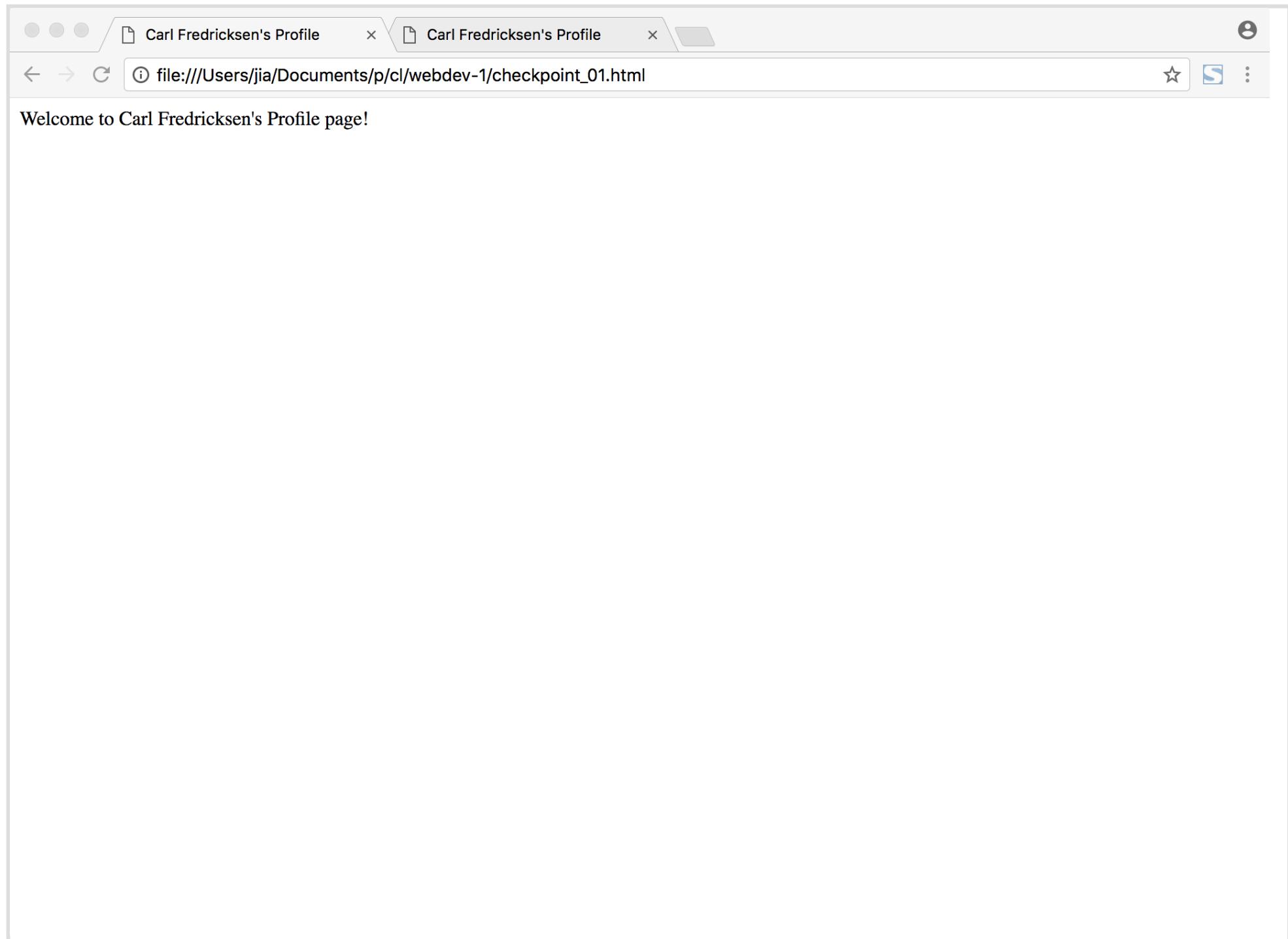
## By the end of this chapter

- Make a self profile page

# Starting very simply

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Carl Fredricksen's Profile</title>
5      </head>
6      <body>
7          Welcome to Carl Fredricksen's Profile page!
8      </body>
9  </html>
```

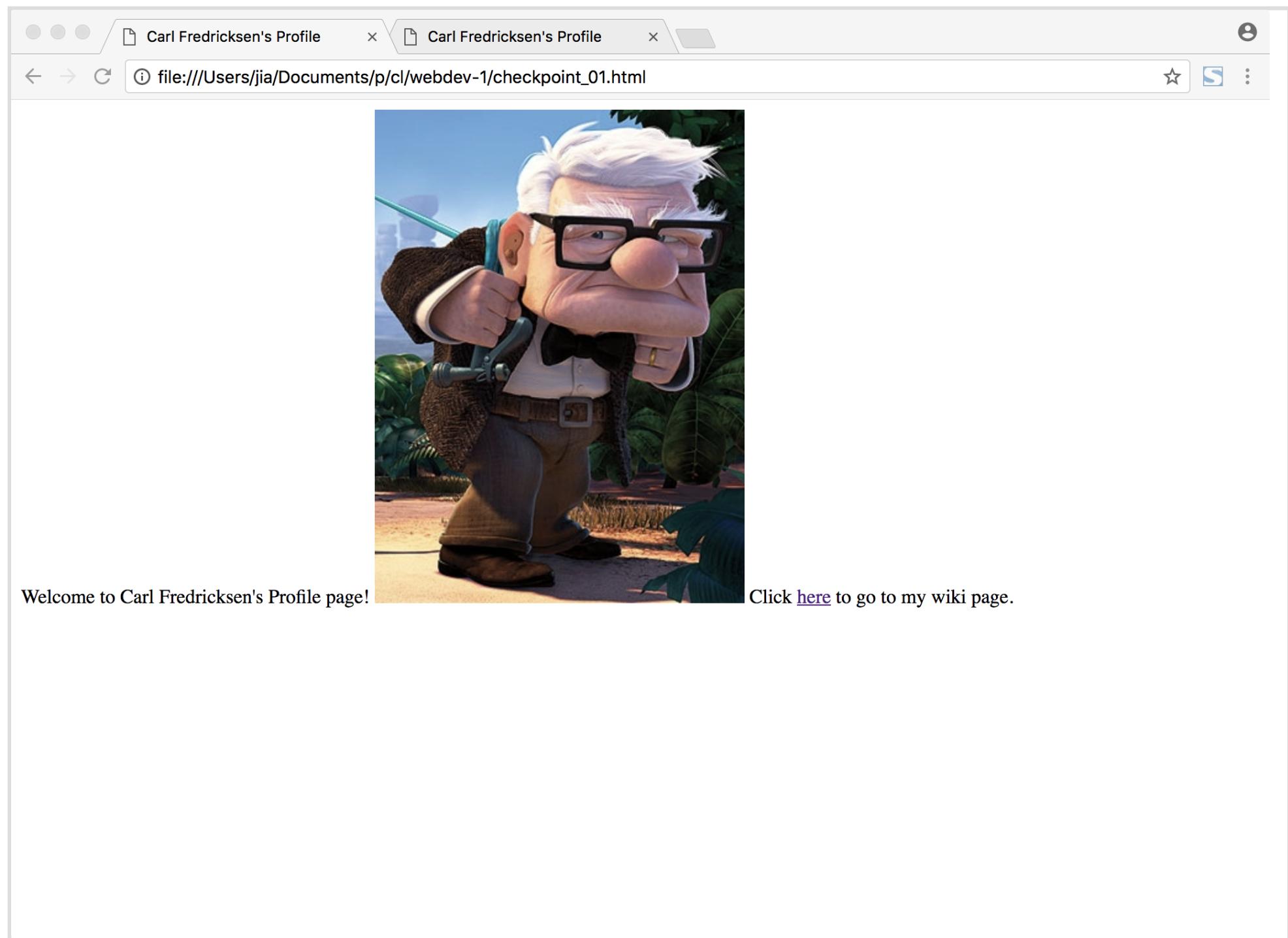
- HTML5 is made up of tags that usually come in nested pairs
- ... like `<X> <Y> ... </Y> </X>`
- The top-most tag is `<html>`, which has two child tags: `<head>` and `<body>`
- `<title>` is a child tag of `<head>`
- Content of the web page goes inside the `<body>` tag



# Images and links

```
1 <body>
2   Welcome to Carl Fredricksen's Profile page!
3   <img src='img/mugshot.jpg'>
4   Click <a href='http://pixar.wikia.com/wiki/Carl_Fredricksen'>here</a>
5   to go to my wiki page.
6 </body>
```

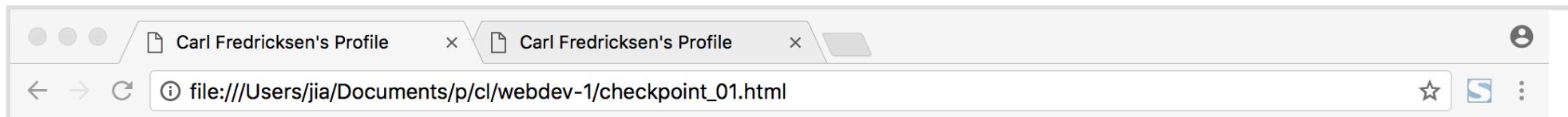
- Tags have attributes (or properties), specified like `name= 'value'`
- Did you notice?
  - `<img>` has no closing tag
  - Browser ignores line breaks. How do we change that?



# Headings, paragraphs and text format

```
1 <body>
2   <h1>Welcome to Carl Fredricksen's Profile page!</h1>
3   <p><img src='img/mugshot.jpg'></p>
4   <h3>Links</h3>
5   <p>Click <a href='http://pixar.wikia.com/wiki/Carl_Fredricksen'>here</a>
6     to go to my wiki page.
7   </p>
8   <h3>Hobbies</h3>
9   <p>I <strong>love</strong> travelling,
10    <em>especially</em> by balloons.
11   </p>
12 </body>
```

- 5 levels of headings: from `<h1>` to `<h5>`
- `<p>` organizes text into paragraphs
- `<em>` (emphasis) is for *italics*, `<strong>` is for **bold**



## Links

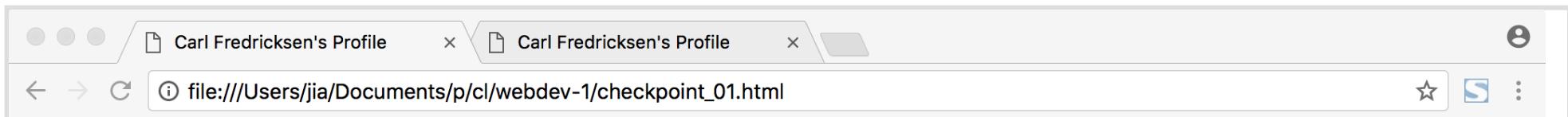
Click [here](#) to go to my wiki page.

## Hobbies

I love travelling, *especially* by balloons.

# Tables

```
1 <table>
2   <tr> <td>Name</td> <td>Carl Fredricksen</td> </tr>
3   <tr> <td>Age</td> <td>78</td> </tr>
4   <tr> <td>Occupation</td> <td>Retiree</td> </tr>
5   <tr> <td>Email</td>
6     <td><a href='mailto:carl@paradisefalls.org'>
7       carl@paradisefalls.org
8     </a>
9   </td>
10  </tr>
11 </table>
```



# Welcome to Carl Fredricksen's Profile page!

## Basic information

Name      Carl Fredricksen

Age      78

Occupation Retiree

Email      [carl@paradisefalls.org](mailto:carl@paradisefalls.org)

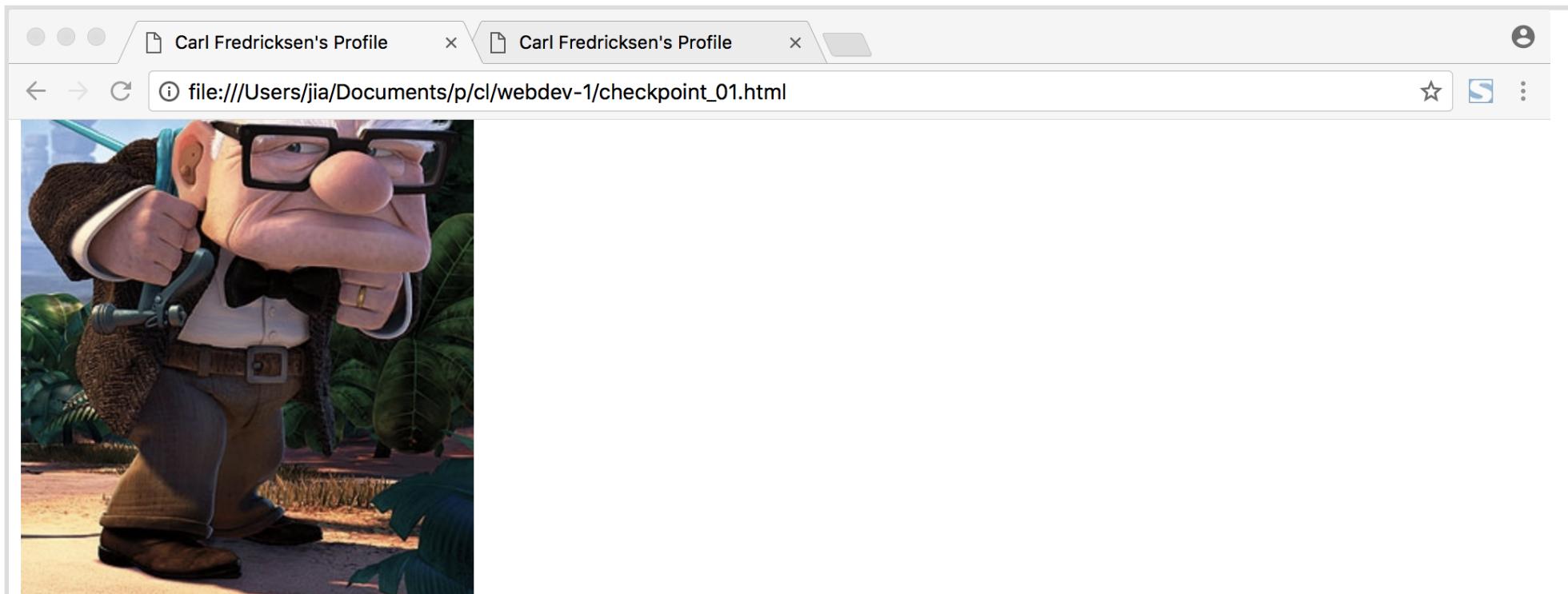


## Links

# Lists, ordered and unordered

```
1      <h3>Friends</h3>
2      <ul>
3          <li>Russell</li>
4          <li>Dug</li>
5          <li>Kevin</li>
6      </ul>
7      <h3>Countries travelled</h3>
8      <ol>
9          <li>Venezuela</li>
10     </ol>
```

- **<ul>** is **unordered list** - bullets are not numbered
- **<ol>** is **ordered list** - bullets are numbered
- **<li>** is **list item**, can be used in either **<ol>** or **<ul>**
- A list can be nested inside another one



## Links

Click [here](#) to go to my wiki page.

## Hobbies

I love travelling, *especially* by balloons.

## Friends

- Russell
- Dug
- Kevin

## Countries travelled

1. Venezuela

## 2. CSS

# What is CSS?

- **Cascading StyleSheet**
- It describes *the format, style and layout* of web pages

## By the end of this chapter

- Format the self profile page like this

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name	Carl Fredricksen
Age	78
Occupation	Retiree
Email	<a href="mailto:craig@paradisefalls.org">craig@paradisefalls.org</a>



## Links

Click [here](#) to go to my wiki page.

## Hobbies

I **love** travelling, especially by balloons.

## Friends

- Russell
- Dug

# Directory structure

Use the following structure to keep things neat and allow for future expansion

```
my_profile_dir
|
|--- css
|   |
|   |--- main.css
|
|--- img
|   |
|   |--- mugshot.jpg
|
|--- index.html
```

# Embedding CSS in web page

```
1 <head>
2   <title>Carl Fredricksen's Profile</title>
3   <link rel='stylesheet' type='text/css' href='css/main.css'>
4 </head>
```

# Introducing the <div> HTML element

```
1      <body>
2          <div id='container'>
3              <div id='header'>
4                  </div>
5                  <div id='overview'>
6                      </div>
7                      <div id='mugshot'>
8                          </div>
9                          <div id='detail'>
10                         </div>
11                         <div id='footer'>
12                             </div>
13                             </div>
14             </body>
```

- Used like containers to group content to sections
- Works together with CSS to format web pages
- Give them unique names using the `id` attribute

# Moving content inside <div>s

```
1 <div id='container'>
2   <div id='header'>
3     <h1>Welcome to Carl Fredricksen's Profile page</h1>
4   </div>
5   <div id='overview'>
6     <table>
7       <tr><td>Name</td> <td>Carl Fredricksen</td></tr>
8       <tr><td>Age </td> <td>78</td></tr>
9       <tr><td>Occupation</td> <td>Retiree</td></tr>
10      <tr><td>Email</td>
11        <td><a href='mailto:carl@paradisefalls.org'>
12          carl@paradisefalls.org</a></td>
13      </tr>
14    </table>
15  </div>
16  ...
17 </div>
```

# Changing colors

```
1   body
2   {
3       color: #333333;
4       background-color: #a7a09a;
5   }
```

- `body` is called the *selector*
- `{ . . . }` contains properties and values being set
- `#333333` and `#a7a09a` are RGB color codes

The screenshot shows a web browser window with two tabs both titled "Carl Fredricksen's Profile". The address bar indicates the page is at "file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html". The main content area features a large heading "Welcome to Carl Fredricksen's Profile page". Below it, there is a table of personal information:

Name	Carl Fredricksen
Age	78
Occupation	Retiree
Email	<a href="mailto:carl@paradisefalls.org">carl@paradisefalls.org</a>

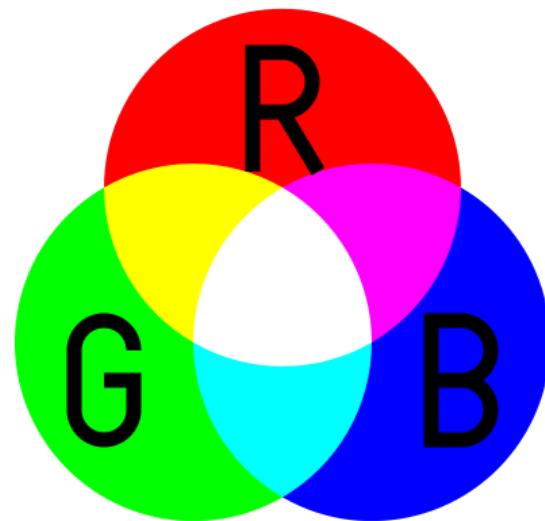
Next to the table is a large, detailed image of Carl Fredricksen from the Pixar movie "Up". He is shown from the waist up, wearing his signature white shirt, brown vest, and glasses, with his hands clasped near his chin.

**Links**

Click [here](#) to go to my wiki page.

**Hobbies**

# The RGB color code



- Uses a mixture of Red, Green and Blue components to represent colors
- A number from 0 to 255 represents the 'intensity' of the component, written in base-16, i.e. from 00 to ff. *Quiz: how many colors can RGB represent?*
- For example: #ff0000 is red, #ffff00 is yellow
- The RGB calculator: [https://www.w3schools.com/colors/colors\\_rgb.asp](https://www.w3schools.com/colors/colors_rgb.asp)

# The *selector* - 1

- "Selects" the HTML element(s) whose properties are to be modified
- 3 basic kinds:
  - by element type, e.g. body { ... }
  - by element id, e.g. #header { ... }
  - by element class, e.g. .someclass { ... } (will see it in chapter 3)
- Mix-and-match
  - example 1: div#header { ... }
  - example 2: div h1 { ... }
  - example 3: #header h1, #header h2 { ... }

*Can you guess what each of selectors in the above examples do?*

# The *selector* - 2

- example 1: `div#header { ... }`
  - The `<div id='header'>` element is selected
- example 2: `div h1 { ... }`
  - All `<h1>` elements that are *inside* any `<div>` element are selected
- example 3: `#header h1, #footer h2 { ... }`
  - All `<h1>` elements that are *inside* any HTML elements with `id='header'`, plus all `<h2>` elements that are *inside* any HTML elements with `id='footer'` are selected
- Selector syntax is complex
  - For complete reference, visit  
[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

# What are the properties I can change?

- This site <https://www.w3schools.com/cssref/default.asp> is a good reference

# Before we go further ...

Let's give each <div> a different background color. This will help our eyes when we re-arrange them.

The screenshot shows a web browser window with two tabs both titled "Carl Fredricksen's Profile". The URL in the address bar is "file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html". The main content area displays a profile for Carl Fredricksen from the Pixar movie "Up".  
**Profile Information:**  
Name: Carl Fredricksen  
Age: 78  
Occupation: Retiree  
Email: [carl@paradisefalls.org](mailto:carl@paradisefalls.org)  
**Image:** A large, detailed 3D rendering of Carl Fredricksen, an elderly man with white hair and glasses, wearing a bow tie and holding a pipe.  
**Links:**  
Click [here](#) to go to my wiki page.  
**Hobbies:**  
I love travelling, especially by balloons.  
**Friends:**

- Russell
- Dug
- Kevin

**Countries travelled:**

- Venezuela

Created on Feb 20, 2018

# Changing fonts

```
1      #header h1
2      {
3          font-family: 'Impact';
4      }
5      #overview
6      {
7          ...
8          font-family: 'Helvetica';
9          font-size: 20pt;
10     }
11     #detail
12     {
13         ...
14         font-family: 'Helvetica';
15         font-size: 16pt;
16     }
17     #footer
18     {
19         font-family: 'Courier New';
20         font-size: 12pt;
21     }
```

The screenshot shows a web browser window with two tabs open, both titled "Carl Fredricksen's Profile". The active tab displays the URL "file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html". The page content is a profile for Carl Fredricksen, featuring a large image of him from the Pixar movie "Up". The profile includes the following information:

Name	Carl Fredricksen
Age	78
Occupation	Retiree
Email	<a href="mailto:carl@paradisefalls.org">carl@paradisefalls.org</a>

**Links**  
Click [here](#) to go to my wiki page.

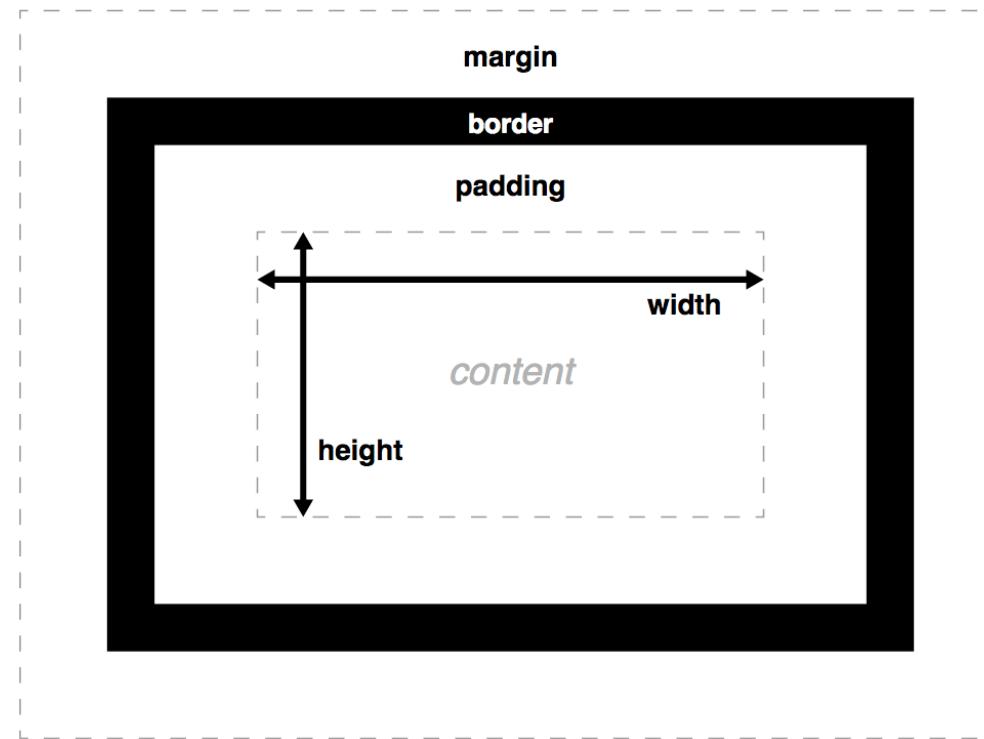
**Hobbies**  
I love travelling, especially by balloons.

**Friends**

- Russell
- Dug
- Kevin

**Countries travelled**

# The Box Model



# Centering the page

```
1      #container
2  {
3      max-width: 750px;
4      margin-top:    0;
5      margin-bottom: 0;
6      margin-left:   auto;
7      margin-right:  auto;
8      background-color: #99c;
9 }
```

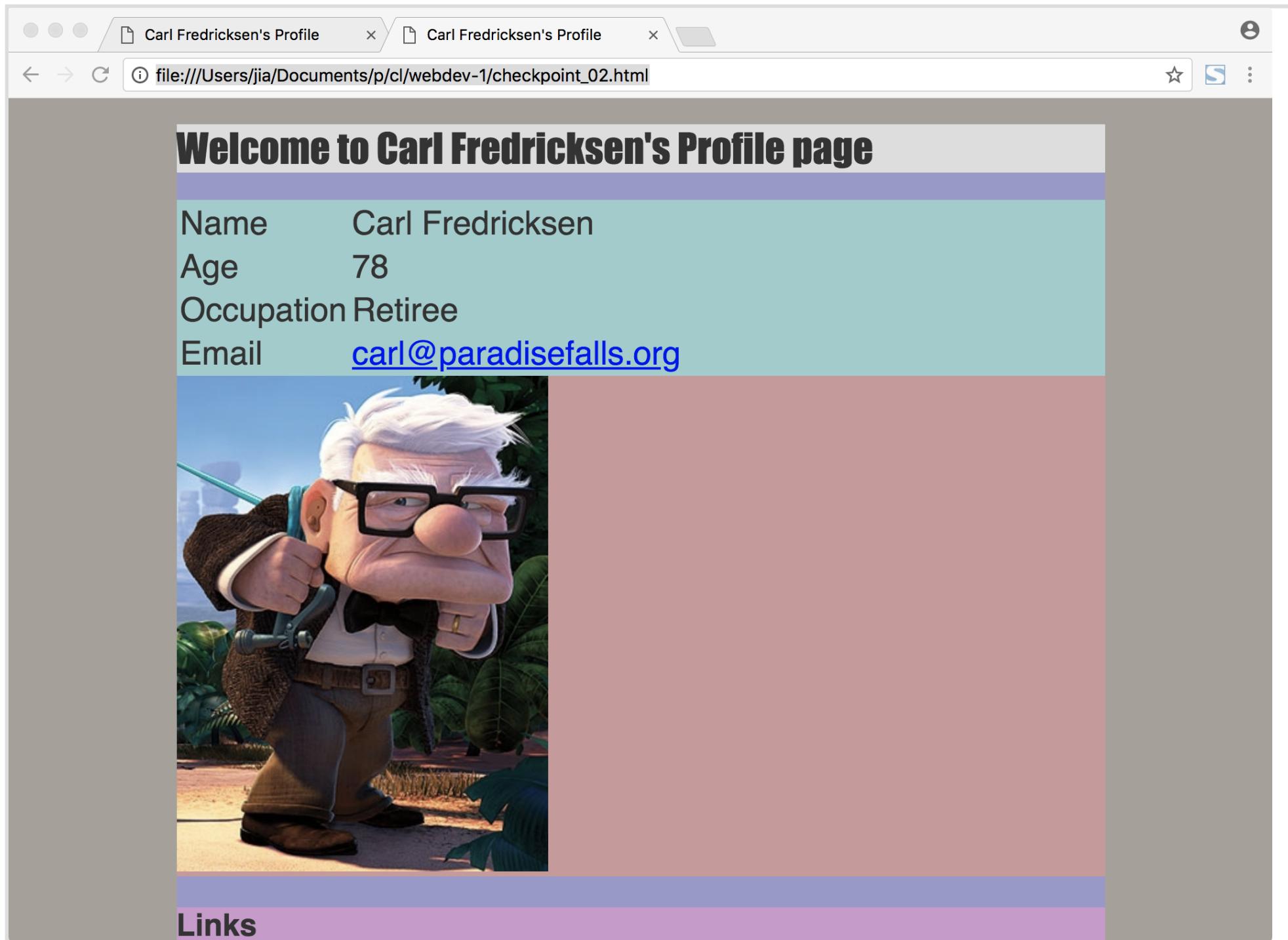
- **max-width**
  - when browser is larger, element's width is set to 750px; when browser is smaller, the element will be automatically resized
- **auto for margin**
  - Browser will automatically determine the margin size using any space left.
  - This has a centering effect when used for left and right margins.

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name    Carl Fredricksen  
Age    78  
Occupation Retiree  
Email    [craig@paradisefalls.org](mailto:craig@paradisefalls.org)



Links

# Putting <div>s side-by-side

```
1  #overview
2  {
3      float: left;
4      width: 550px;
5  }
6  #mugshot
7  {
8      float: right;
9      width: 200px;
10 }
11 #mugshot img
12 {
13     display: block; /* what does this line do??? ;-) */
14     max-width: 100%;
15     max-height: 100%;
16 }
```

- `float` allows elements to be side by side, instead of one after another
- Remove `display: block` and see what happens. [Explanation](#)

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name    Carl Fredricksen  
Age    78  
Occupation Retiree  
Email    [craig@paradisefalls.org](mailto:craig@paradisefalls.org)

## Links

Click [here](#) to go to my wiki page.

## Hobbies

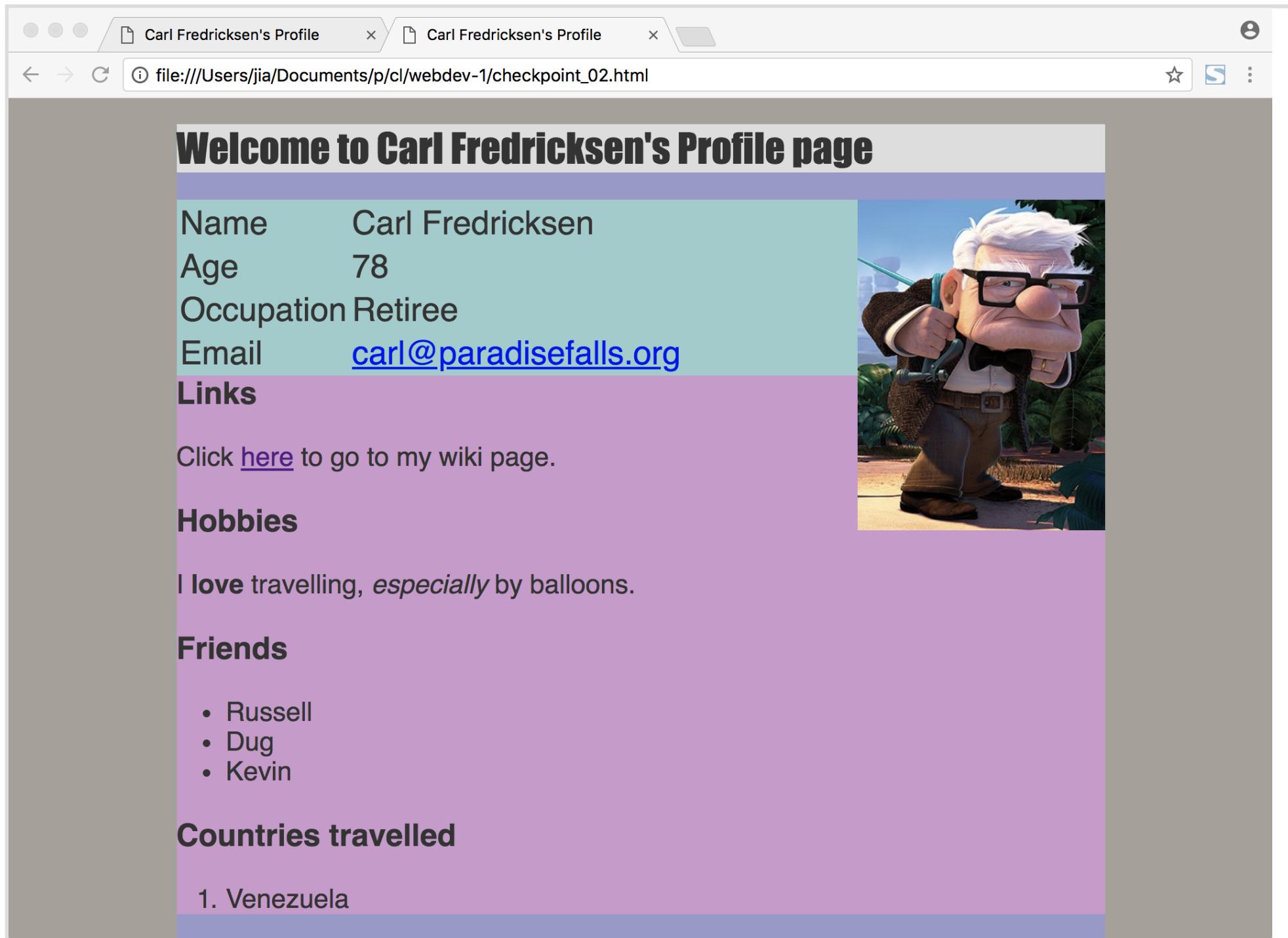
I love travelling, especially by balloons.

## Friends

- Russell
- Dug
- Kevin

## Countries travelled

1. Venezuela



# Staying clear of floated elements

```
1      #detail  
2      {  
3          ...  
4          clear: both;  
5      }
```

- Did you notice the 'detail' section went under the mugshot?
  - This happens to elements next to floated ones
  - Solution: `clear: both;`

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name	Carl Fredricksen
Age	78
Occupation	Retiree
Email	<a href="mailto:craig@paradisefalls.org">craig@paradisefalls.org</a>



## Links

Click [here](#) to go to my wiki page.

## Hobbies

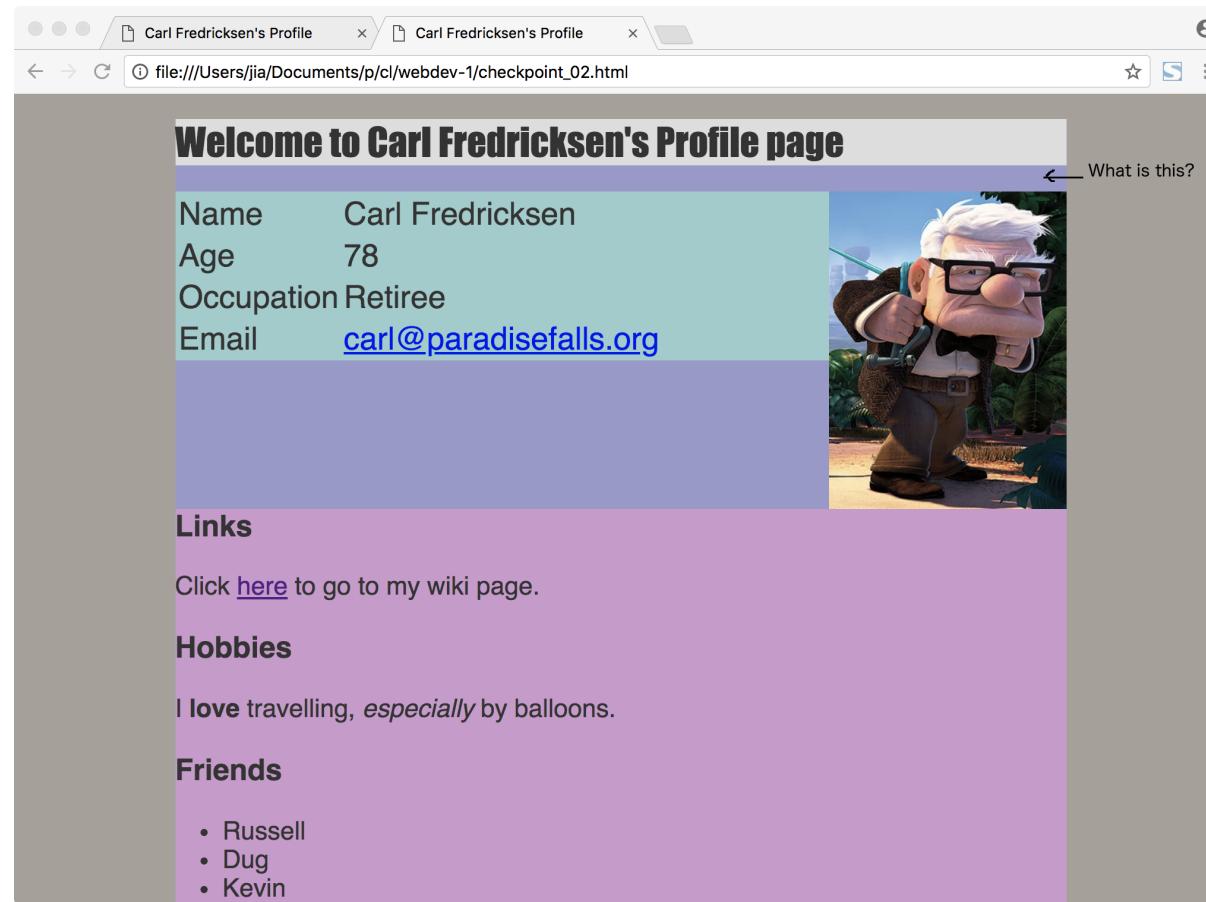
I **love** travelling, especially by balloons.

## Friends

- Russell
- Dug
- Kevin

# Using Chrome to troubleshoot

- What is this gap? Let's use Chrome's Developer Tool to find out



Carl Fredricksen's Profile    Carl Fredricksen's Profile

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

**Carl Fredricksen**

Age 78  
Occupation Retiree  
Email [carn@paradisefalls.org](mailto:carn@paradisefalls.org)

The developer tools sidebar shows the following:

- Elements** tab: Shows the DOM structure of the page.
- Styles** tab: Shows the CSS rules applied to the selected element (an **h1** tag).
- Computed** tab: Shows the final computed values for the selected element.
- Event Listeners** tab: Shows any event listeners attached to the selected element.

**Selected Element:** `<h1>Welcome to Carl Fredricksen's Profile page</h1>`

**Computed Styles (body #container #header h1):**

```

:root {
    --color-primary: #333333;
    --color-secondary: #a7a09a;
}

body {
    margin: 0;
    color: var(--color-primary);
    background-color: var(--color-secondary);
}

#header h1 {
    font-size: 1.5em;
    font-weight: bold;
    color: black;
}

```

**Element Box Model Diagram:**

```

margin: 21.440px
border: -px
padding: -px
width: 750px
height: 39px

```

# Margin collapsing

- "The top and bottom margins of blocks are sometimes combined (collapsed) into a single margin whose size is the largest of the individual margins (or just one of them, if they are equal), a behavior known as margin collapsing."  
- [MDN](#)
- Occurs when parent has no margin or padding to separate itself from its first or last child
- The collapsed margin ends up outside the parent

# Margin collapsing - the fix

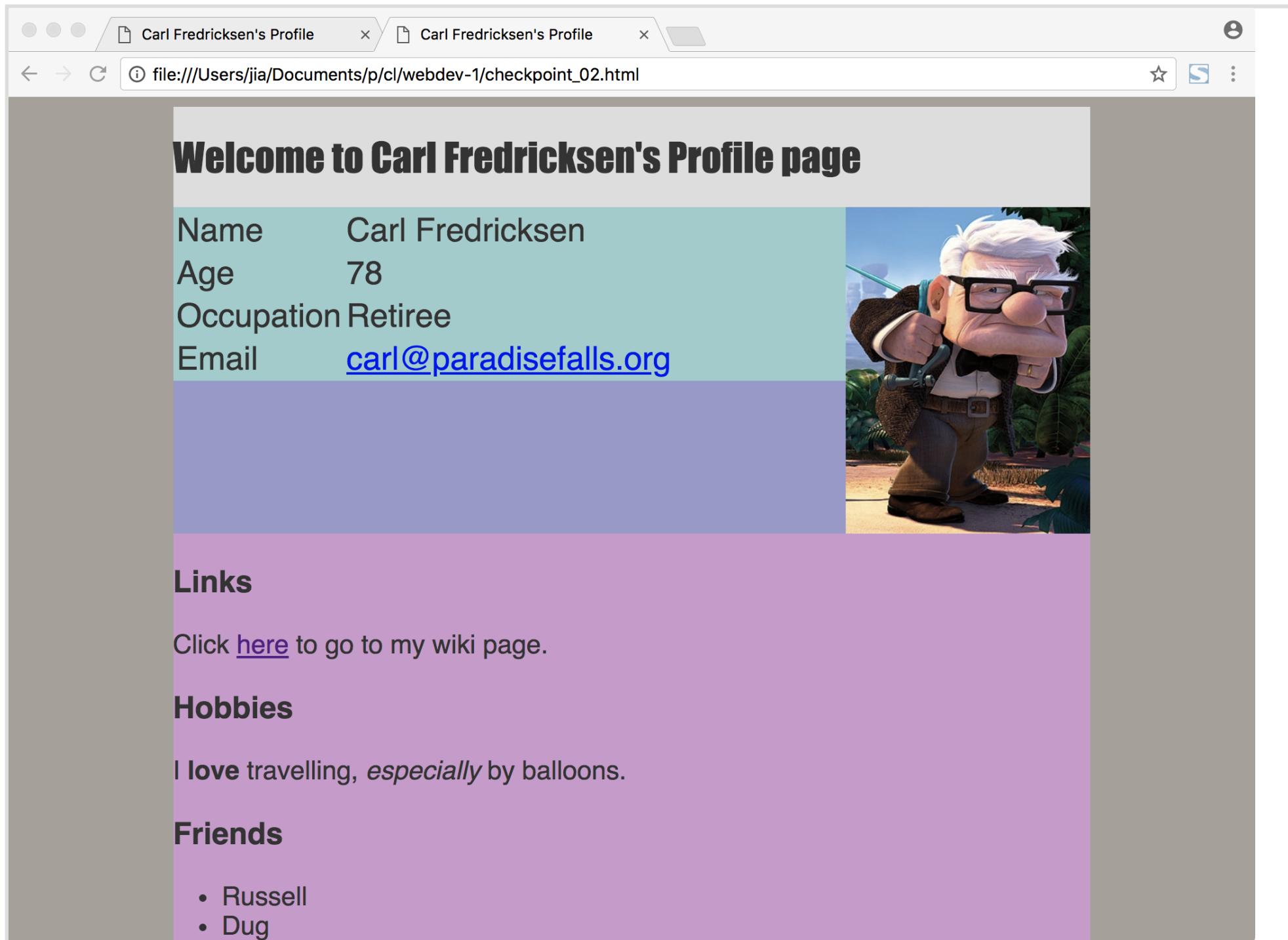
```
1      div
2      {
3          padding-top: 0.1px;
4          padding-bottom: 0.1px;
5      }
```

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name      Carl Fredricksen  
Age        78  
Occupation Retiree  
Email      [carl@paradisefalls.org](mailto:carl@paradisefalls.org)



**Links**

Click [here](#) to go to my wiki page.

**Hobbies**

I love travelling, especially by balloons.

**Friends**

- Russell
- Dug

# Final touches - 1

Add paddings to make things look less crammed

```
1      #overview
2  {
3      padding-top:    5px;
4      padding-bottom: 5px;
5      padding-left:   15px;
6      padding-right:  15px;
7  }
8  #detail
9  {
10     padding: 5px 15px;
11 }
12 #footer
13 {
14     padding: 5px 15px;
15 }
```

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name	Carl Fredricksen
Age	78
Occupation	Retiree
Email	<a href="mailto:carl@paradisefalls.org">carl@paradisefalls.org</a>

**Links**

Click [here](#) to go to my wiki page.

**Hobbies**

# Finak touches - 1 *what's wrong?*

Need to subtract left and right padding from width

```
1 #overview
2 {
3     width: 520px;          /* was 550px */
4     padding-top: 5px;
5     padding-bottom: 5px;
6     padding-left: 15px;
7     padding-right: 15px;
8 }
```

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name	Carl Fredricksen
Age	78
Occupation	Retiree
Email	<a href="mailto:carl@paradisefalls.org">carl@paradisefalls.org</a>

**Links**

Click [here](#) to go to my wiki page.

**Hobbies**

I **love** travelling, especially by balloons.

**Friends**

- Russell
- Dug

# Final touches - 2

```
1      body
2      {
3          margin: 0;
4      }
5      #header h1
6      {
7          text-align: center;
8      }
9      td#fieldname
10     {
11         width: 40%;
12     }
13     #overview
14     {
15         background-color: #9cc;
16     }
```

Carl Fredricksen's Profile    x    Carl Fredricksen's Profile    x

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_02.html

# Welcome to Carl Fredricksen's Profile page

Name	Carl Fredricksen
Age	78
Occupation	Retiree
Email	<a href="mailto:craig@paradisefalls.org">craig@paradisefalls.org</a>



## Links

Click [here](#) to go to my wiki page.

## Hobbies

I **love** travelling, especially by balloons.

## Friends

- Russell
- Dug

# 3. Javascript

# What is Javascript, and why?

- The scripting language for the web, often used in browsers (called front-end scripting).
- We can use Javascript to do many things
  - add/change/remove content
  - validate user input
  - show/hide content with CSS
  - create a game using HTML5 Canvas

# Directory structure

Use the following structure to keep things neat and allow for future expansion

```
1  <html>
2    ...
3    <body>
4    ...
5    </body>
6    <script src='js/main.js'></script>
7  </html>
```

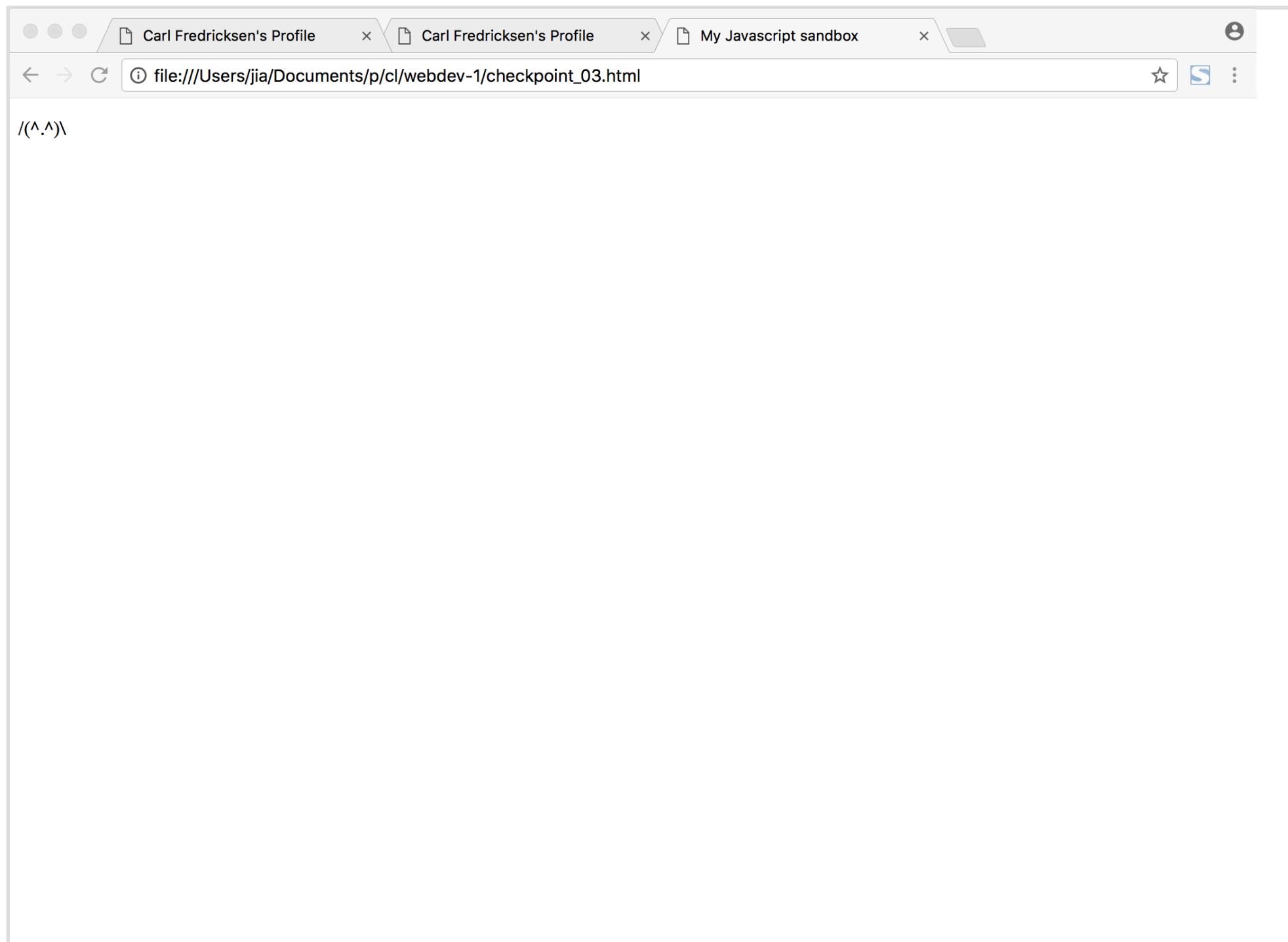
# Puppy face /(^.^)\

## index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>My Javascript sandbox</title>
5      </head>
6      <body>
7          <div id='myDiv'></div>
8      </body>
9      <script src='js/main.js'></script>
10 
```

## css/main.js

```
1 var myDiv = document.getElementById('myDiv');
2 var myP = document.createElement('p');
3 myP.innerText = '/(^.^)\\';
4 myDiv.appendChild(myP);
```



# Puppy face / (^.^)\ what's going on?

1. In HTML, create an empty <div> and give it an unique ID 'myDiv'
2. In Javascript, retrieve (aka getting a handle of) this <div> using  
`document.getElementById()`
3. Create a new <p> element using `document.createElement()`
4. Set this <p>'s `innerText` property to puppy face
5. Add this <p> to myDiv as a child using `myDiv.appendChild()`

## Javascript syntax

- Use `var` to declare a variable
- Use `=` to assign value to an variable or its property
- Use single quote '`'` (or double quote "`"") to enclose a string of characters`

# Lots of puppies

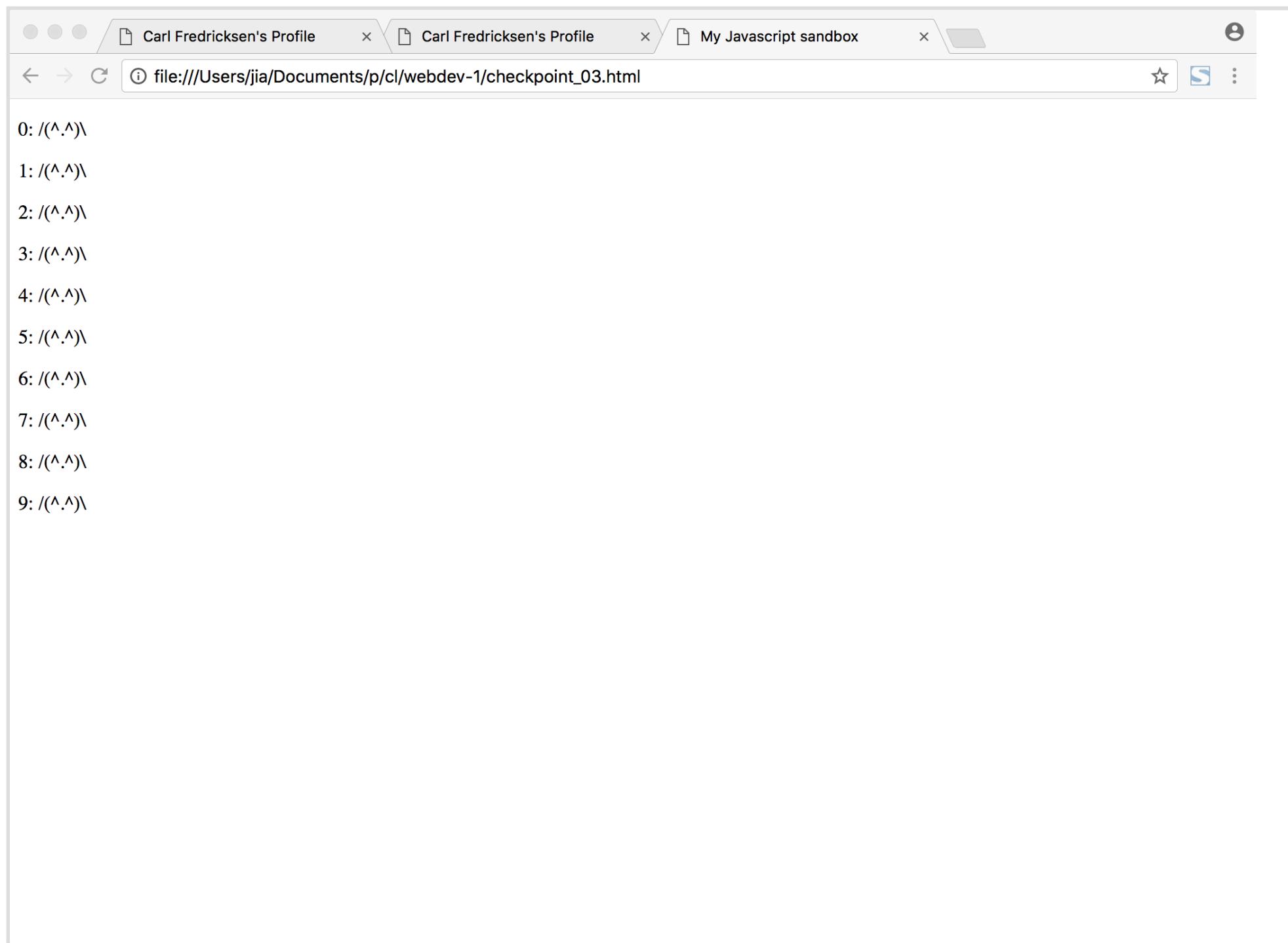
```
1 var myDiv = document.getElementById('myDiv');
2 for (var i = 0; i < 10; i++)
3 {
4     var myP = document.createElement('p');
5     myP.innerText = i + ': ' + '/(^.^)\\"';
6     myDiv.appendChild(myP);
7 }
```

## Javascript syntax

- The for-loop:

```
for ( [Initial Expression]; [Condition]; [Step Expression] )
{ ACTION }
```

- `i++` is a shorthand for `i = i + 1`
- `+` does addition when both operands are numbers; when any of them is a string, it does string *concatenation*

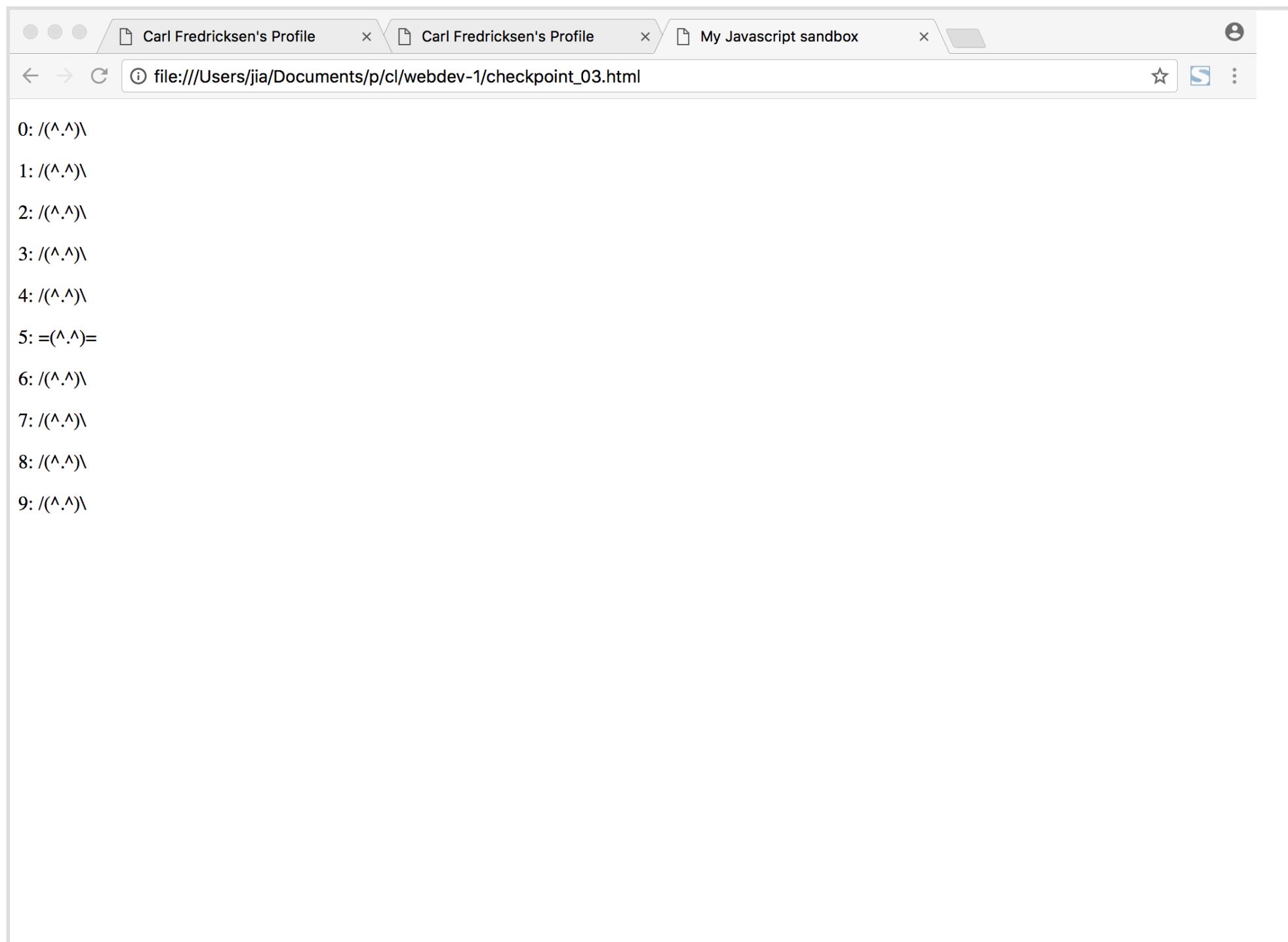


# Kitty in puppies

```
1  var myDiv = document.getElementById('myDiv');
2  for (var i = 0; i < 10; i++)
3  {
4      var myP = document.createElement('p');
5      if (i === 5)
6      {
7          myP.innerText = i + ': ' + '=^(^.^)=';
8      }
9      else
10     {
11         myP.innerText = i + ': ' + '/(^.^)\\"';
12     }
13     myDiv.appendChild(myP);
14 }
```

## Javascript syntax

- The if-else-branches: `if ( [Condition] ) { Action A } else { Action B }`
- `i++` is a shorthand for `i = i + 1`



# Introducing booleans

- A boolean is either `true` or `false`
  - Real world example: light switch (on or off), yes/no question (yes or no)
- [Condition] used in for-loop or if-else-branches must evaluate to a boolean
- How to write a boolean expression?
  - use comparison operators

Operator	Meaning
<code>a === b</code>	Is a equal b?
<code>a != b</code>	Is a NOT equal b?
<code>a &gt; b</code>	Is a greater than b?
<code>a &gt;= b</code>	Is a greater than or equal to b?
<code>a &lt; b</code>	Is a less than b?
<code>a &lt;= b</code>	Is a less than or equal to b?
◦ <code>true</code> and <code>false</code> themselves are booleans	

# Boolean operators

- Boolean operators work on booleans just like + or – work on numbers

Name	Operator	Meaning
AND	<code>a &amp;&amp; b</code>	true if <b>both</b> a <b>and</b> b are true, false otherwise
OR	<code>a    b</code>	true if <b>either</b> a <b>or</b> b is true, false otherwise
NOT	<code>! a</code>	true if a is false, false if a is true

- Quiz: Evaluate the following boolean expressions, given a=5 and b=7
  - `a > 6 && b > 6`
  - `a != 7 || b === 7`
  - `(a + b) <= 12`
  - `!((a - b) > 0)`
  - `a < 8 && b < 8 && (a + b) <= 12`

# Creating functions

Why? To reuse code, and to reduce code complexity.

A function has 3 parts: a name, a list of parameters and a body.

```
1  function drawPuppiesAndKitty(where, howMany)
2  {
3      var myDiv = document.getElementById(where);
4
5      for (var i = 0; i < howMany; i++)
6      {
7          var myP = document.createElement('p');
8          if (i === 5)
9          {
10              myP.innerText = i + ': ' + '=(^.^)=';
11          }
12          else
13          {
14              myP.innerText = i + ': ' + '/(^.^)\\";
15          }
16          myDiv.appendChild(myP);
17      }
18  }
```

# It's a Pawdemonium!

## HTML

```
1 ...  
2 <body>  
3   <div id='myDiv'><p>Here are some furry friends</p></div>  
4   <div id='myDiv2'><p>Another bunch of furry friends</p></div>  
5 </body>  
6 ...
```

## Javascript

```
1 function drawPuppiesAndKitty(where, howMany)  
2 {  
3   ...  
4 }  
5  
6 drawPuppiesAndKitty('myDiv', 7);  
7 drawPuppiesAndKitty('myDiv2', 6);
```

The screenshot shows a web browser window with three tabs open. The tabs are titled "Carl Fredricksen's Profile", "Carl Fredricksen's Profile", and "My Javascript sandbox". The URL bar shows the file path "file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_03.html". The browser interface includes standard controls like back, forward, and search, along with a star icon for bookmarks and a settings icon.

Here are some furry friends

```
0: /(^.^)\n1: /(^.^)\n2: /(^.^)\n3: /(^.^)\n4: /(^.^)\n5: =(^.^)=\n6: /(^.^)\n
```

Another bunch of furry friends

```
0: /(^.^)\n1: /(^.^)\n2: /(^.^)\n3: /(^.^)\n4: /(^.^)\n5: =(^.^)=\n
```

# Introducing arrays

Think of an array as a row of chairs with numbers (starting from 0) on them



Let's create an array of pets:

```
1 var pets = ['P', 'K', 'P', 'P', 'K', 'K']; // P for Puppy, K for Kitty
```

You can *access* each pet by specifying its *index*, like `var secondPet = pets[1];`

Can you write a function to draw an array of pets? *Hint: use a for-loop*

# Puppy or kitty, it's your call - 1

Create a new <div> for this exercise

```
1 <div id='myDiv3'><p>An array of pets</p></div>
```

We will declare and call our new function this way

```
1 function drawPets(where, arrayOfPets)
2 {
3     ...
4 }
5
6 drawPets('myDiv3', pets);
```

# Puppy or kitty, it's your call - 2

```
1  function drawPets(where, arrayOfPets)
2  {
3      var myDiv = document.getElementById(where);
4      for (var i = 0; i < arrayOfPets.length; i++)
5      {
6          var pet = arrayOfPets[i];
7          var myP = document.createElement('p');
8          switch (pet)
9          {
10             case 'P':
11                 myP.innerText = i + ': ' + '/(^.^)\\"';
12                 break;
13             case 'K':
14                 myP.innerText = i + ': ' + '=(^.^)=';
15                 break;
16             default:
17                 myP.innerText = i + ': ' + '???';
18                 break;
19             }
20             myDiv.appendChild(myP);
21         }
22     }
```

The screenshot shows a web browser window with three tabs open. The tabs are titled 'Carl Fredricksen's Profile', 'Carl Fredricksen's Profile', and 'My Javascript sandbox'. The URL bar indicates the page is 'file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_03.html'. The main content area displays an array of objects with the following properties:

- 0: /(^.^)\
- 1: =(^.^)=
- 2: /(^.^)\
- 3: /(^.^)\
- 4: =(^.^)=
- 5: =(^.^)=

# A puppy, kitty, or something else???

`switch` is handy when there are more than 2 branches

```
1      switch (VARIABLE)
2  {
3      case VALUE_1:
4          ACTION_1;
5          break;
6      case VALUE_2:
7          ACTION_2;
8          break;
9      ...
10     default:
11         ACTION_DEFAULT;
12         break;
13 }
```

Try drawing this array of pets: [ 'P' , 'K' , 'P' , 'P' , 'K' , 'F' ]

# Hide-and-Seek!

Allow user to play hide-and-seek with a chosen pet. You need the following:

- a text box for user to enter the pet's number
- a 'Hide-and-Seek` button

## HTML

```
1 ...  
2 <div>  
3   <label>Which pet?  
4     <input id='petNumber' type='text' placeholder='Enter pet number'>  
5   </label>  
6   <button onclick='hideAndSeek("myDiv3")'>Hide-and-Seek</button>  
7 </div>  
8 ...
```

The screenshot shows a web browser window with three tabs open:

- Carl Fredricksen's Profile
- Carl Fredricksen's Profile
- My Javascript sandbox

The address bar displays the URL: file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_03.html

The main content area contains the following text:

```
Array of pets
0: /(^.^)\\
1: =(^.^)=
2: /(^.^)\\
3: /(^.^)\\
4: =(^.^)=
5: /(^.^)\\\
```

Below this text is a user interface for searching:

Which pet?

# Introducing events and event handlers

- Analogy: You are home watching TV. The door bell rings, and you run downstairs to answer the door.
  - Event: door bell rings
  - Event handler: you answer the door
- For the complete list of events that you can respond to, see [this page](#)
- Event handlers are written in Javascript

# Naming the pets

```
1 ...  
2     for (var i = 0; i < arrayOfPets.length; i ++)  
3     {  
4         var pet = arrayOfPets[i];  
5         var myP = document.createElement('p');  
6         myP.id = where + '_' + i;  
7         switch (pet)  
8         {  
9             case 'P':  
10                ...
```

Now, take a look in Chrome Developer Tools to inspect the generated content

Carl Fredricksen's Profile    Carl Fredricksen's Profile    My Javascript sandbox

file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_03.html

**p#myDiv3\_0 | 764 x 18**

Array of pets

0: /(^.^)\n1: =(^.^)=\n2: /(^.^)\n3: /(^.^)\n4: =(^.^)=\n5: /(^.^)\n

Which pet?  Enter pet number

Elements

```
<!DOCTYPE html>
<html>
  <head>...</head>
  ...<body> == $0
    <div id="myDiv3">
      <p>Array of pets</p>
      <p id="myDiv3_0">0: /(^.^)\n</p>
      <p id="myDiv3_1">1: =(^.^)=</p>
      <p id="myDiv3_2">2: /(^.^)\n</p>
      <p id="myDiv3_3">3: /(^.^)\n</p>
      <p id="myDiv3_4">4: =(^.^)=</p>
      <p id="myDiv3_5">5: /(^.^)\n</p>
    </div>
    <div>...</div>
    <script src="js/main.js">
    </script>
  </body>
</html>
```

html body

Styles Computed Event Listeners >

Filter :hover .cls +

```
element.style { }
body { user agent stylesheet
  display: block;
  margin: 8px;
}
```

margin 8  
border -  
padding -  
764 x 257  
-

# The CSS Invisibility Cloak

We will create a *class* in CSS called `hidden`. Any HTML element belong to this class will become invisible.

In CSS, the selector for a *class* is denoted by a `.` in front of its name.

CSS

```
1      .hidden
2      {
3          visibility: hidden;
4      }
```

HTML

```
1 <head>
2     <title>My Javascript sandbox</title>
3     <link rel="stylesheet" type="text/css" href="css/checkpoint_03.css">
4 </head>
```

# Now you see me, now you don't

```
1  function hideAndSeek(where)
2  {
3      var petNumberInput = document.getElementById('petNumber');
4      var petID = where + "_" + petNumberInput.value;
5      var petP = document.getElementById(petID);
6
7      if (petP.className != 'hidden')
8      {
9          petP.className = 'hidden';
10     }
11    else
12    {
13        petP.className = '';
14    }
15 }
```

The screenshot shows a web browser window with three tabs open:

- Carl Fredricksen's Profile
- Carl Fredricksen's Profile
- My Javascript sandbox

The address bar displays the URL: file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_03.html

The main content area contains the following text:

An array of pets

0: /(^.^)\|

1: =(^.^)=

2: /(^.^)\|

4: =(^.^)=

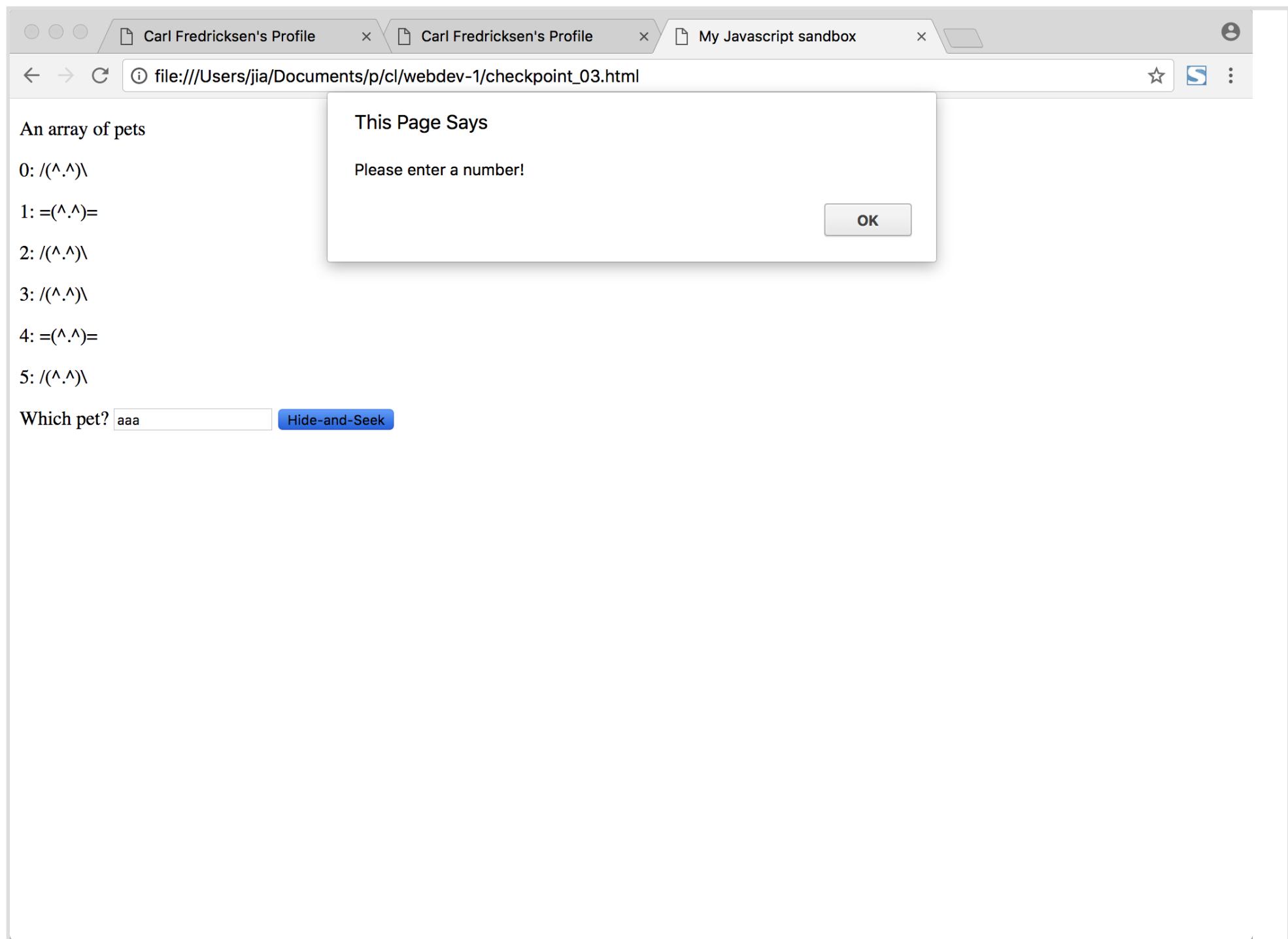
5: /(^.^)\|

Which pet?

# Validating input - 1

Check if user has entered a number

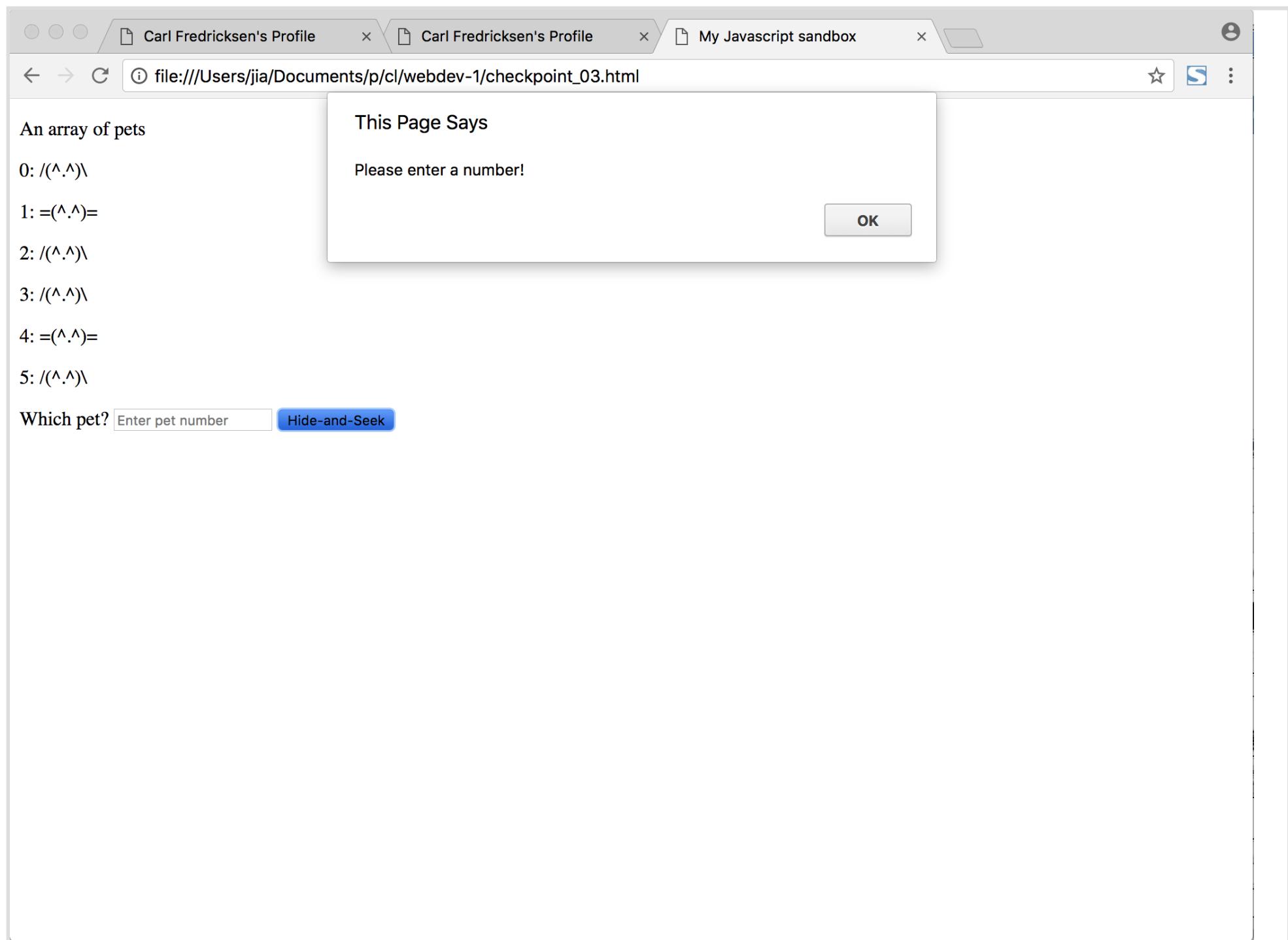
```
1  function hideAndSeek(where)
2  {
3      var petNumberInput = document.getElementById('petNumber');
4      if (isNaN(petNumberInput.value))
5      {
6          alert('Please enter a number!')
7          return;
8      }
9      ...
10 }
```



# Validating input - 2

Try pressing the button without entering anything. Can you fix it?

```
1 function hideAndSeek(where)
2 {
3     var petNumberInput = document.getElementById('petNumber');
4     if (isNaN(petNumberInput.value) || petNumberInput.value === '')
5     {
6         alert('Please enter a number!')
7         return;
8     }
9     ...
10 }
```



# Validating input - 3

Try entering a pet number that doesn't exist. How do you fix it?

```
1  function hideAndSeek(where)
2  {
3      ...
4      var petID = where + "_" + petNumberInput.value;
5      var petP = document.getElementById(petID);
6
7      if (petP === null)
8      {
9          alert("Pet number doesn't exist!");
10     }
11     ...
12 }
```

The screenshot shows a web browser window with three tabs: "Carl Fredricksen's Profile", "Carl Fredricksen's Profile", and "My Javascript sandbox". The active tab is "My Javascript sandbox" which displays the URL "file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_03.html". The page content includes a list of numbered items and a search input field.

An array of pets

0: /(^.^)\|

1: =(^.^)=

2: /(^.^)\|

3: /(^.^)\|

4: =(^.^)=

5: /(^.^)\|

Which pet?  Hide-and-Seek

This Page Says

Pet number doesn't exist!

OK

Detailed description: This is a screenshot of a web browser window. The title bar says 'Web Development From Scratch'. There are three tabs open: 'Carl Fredricksen's Profile' (active), 'Carl Fredricksen's Profile', and 'My Javascript sandbox'. The 'My Javascript sandbox' tab is active and shows the URL 'file:///Users/jia/Documents/p/cl/webdev-1/checkpoint\_03.html'. The page content includes a list of numbered items and a search input field. A modal dialog box is overlaid on the page, containing the text 'This Page Says' and 'Pet number doesn't exist!', with an 'OK' button.

# We are almost at the end

## What did we learn in this course?

- HTML
  - h1, p, em, strong, img, a, ul, ol, table, div, input, button
- CSS
  - Embedding in HTML
  - Selector syntax
  - RGB color code
  - Box Model
  - Using float and clear to create multi-column layout
  - Changing fonts, width
  - Using visibility to hide/unhide content

# We are almost at the end

## What did we learn in this course?

- Javascript
  - Embedding in HTML
  - Accessing HTML elements by ID
  - Creating HTML elements and attach to page
  - for-loop, if-else, switch
  - Booleans, comparison operators and boolean operators
  - Functions, creating and calling
  - Arrays, creating and accessing elements
  - Events and event handlers
  - Changing CSS class programmatically
  - Validating user input

# What next?

Javascript/HTML5 Game Development

Web Development, Intermediate Level

# The END

