



First Nations
Technology Council

Focus Web Developer

Week 2 – Lesson 2

Lesson Topics

- HTML 5 Practice Continued

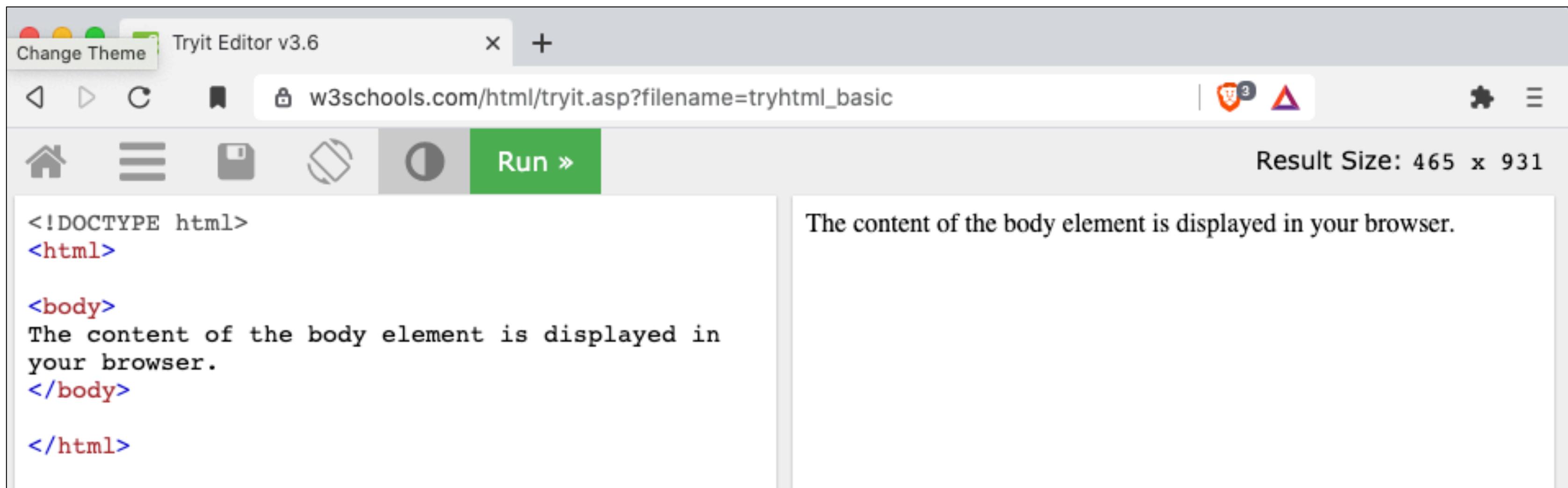


Quick tip!

- Bookmark this URL:

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic

- use this to test little bits of code!



The screenshot shows the Tryit Editor interface. The title bar reads "Tryit Editor v3.6". The toolbar includes icons for "Change Theme", "Run", and "Stop". The address bar shows the URL "w3schools.com/html/tryit.asp?filename=tryhtml_basic". The main editor area contains the following HTML code:

```
<!DOCTYPE html>
<html>

<body>
The content of the body element is displayed in
your browser.
</body>

</html>
```

The preview pane on the right displays the output: "The content of the body element is displayed in your browser." with a result size of 465 x 931.

Activity 1 Introduction

Create the following webpage using **only** HTML

This is an example of a <form> in HTML

Create an Account

Email:

Password:

Age:

Tell us a little about yourself:

I agree to the [terms of service](#)

Dropdown menu

checkbox

Activity 1 Introduction

In addition to the <tags> in your boilerplate, here are some of the <tags> to create this webpage:

- <h1> - Defines a hyperlink
- <form> - Defines an HTML form for user input
 - <input> - Defines input control
 - <label> - Defines a label for an <input> element
 - <select> - Defines a dropdown list
 - <option> - Defines an option of a dropdown list
 - <textarea> - Defines a multiline input control (text area)

Note: this page has no <header> or <footer> tags

For a full list of HTML <tags> aka "elements" go here:

<https://www.w3schools.com/TAGs/>

HTML <form> element

When the <form> element is used all these other elements can be **nested** within it

- <input> - Defines input control
- <label> - Defines a label for an <input> element
- <select> - Defines a dropdown list
- <option> - Defines an option of a dropdown list
- <textarea> - Defines a multiline input control (text area)

HTML <input> element

<input> is used to take input from the user. Use the **type** and **name** attributes:

Example:

```
<input type = "text" name = "firstName">
```

- The input is stored **type** “text” (aka a string) by the computer
- The input is stored with a **name** of “firstName” by the computer
 - This is like creating a variable named “firstName” that becomes whatever the user enters in the field

Note: <input> is a self closing tag and **does not need a closing tag!**

~~<input> blah blah blah </input>~~

HTML <label> element

The <label> element creates labels for other elements.

- no attributes are needed (here) for <label>

Example of how to use <label> and <input>

HTML Code

```
<label>First Name:</label>
<input type = "text" name = "firstName">
```

What you see in the web browser

First Name:

HTML <select> element

The <select> element is used to define a drop-down. Use the attribute **name**.

Example:

```
<select name = “drop_down_list”>
```

- this defines (or names) this drop-down list as “drop_down_list” for the computer to reference later – this will not be displayed

HTML <option> element

The <option> element makes options in a drop-down list that was defined with the <select>. Use the **value** attribute:

Example of <option>:

```
<option value = “option1”> this is option 1< /option>
```

- The value is what the computer stores this option as (not what is displayed)

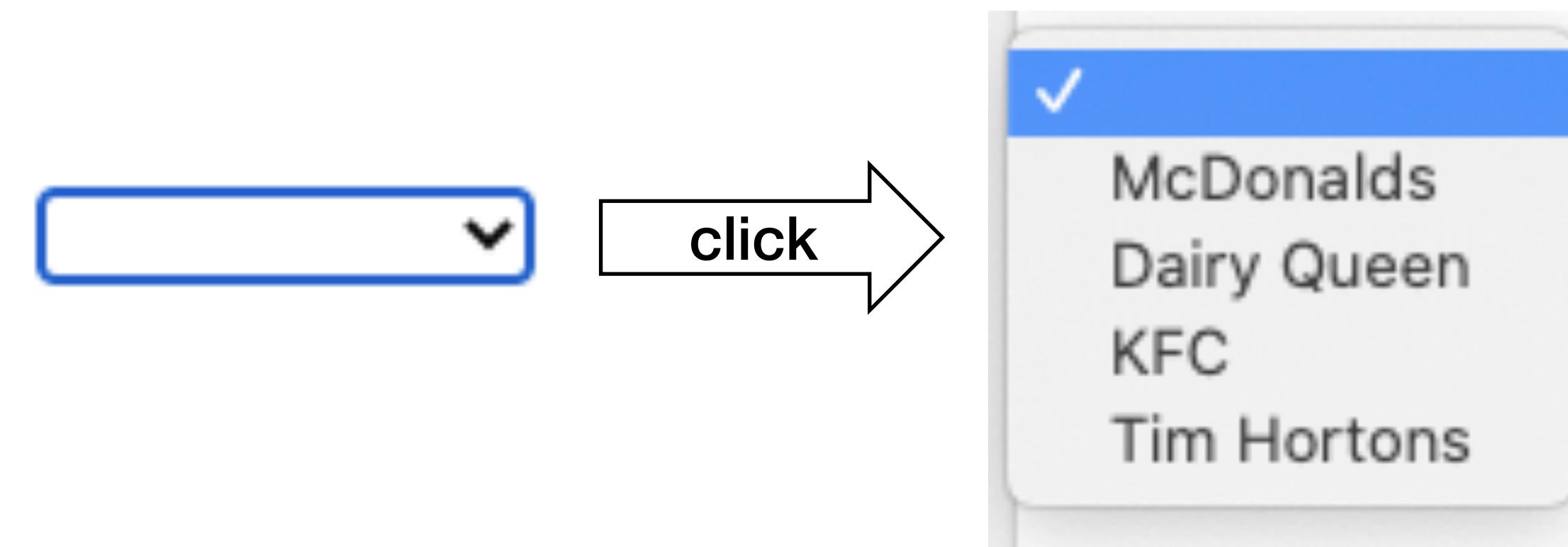
HTML <select> and <option> elements

Example of using <select> and <option> together:

HTML Code

```
<select name="age">
<option value=""></option>
<option value="option1">McDonalds</option>
<option value="option2">Dairy Queen</option>
<option value="option3">KFC</option>
<option value="option4">Tim Hortons</option>
</select>
```

What you see in the web browser



Why do you think value = "" is used in the first option tag?

Why is there no value in the menu before it is clicked?

HTML <textarea> element

The <textarea> element defines a multiline entry field for the user. Use the attributes **name**, **rows** and **cols**.

Example:

```
<textarea name=“userAnswer” rows = “10” columns = “5”>Some text</textarea>
```

- **name** - this is what the computer stores the input as
- **rows** - the height of the box
- **columns** – the width of the text box

HTML <textarea> element

Example of using <textarea>:

HTML Code

```
<textarea name="about" rows="2" cols="40">what do I put here?</textarea>
```

What you see in the web browser

what do I put here?

Activity 1 Instructions

Create the following webpage using **only** HTML

Create an Account

Email:

Password:

Age: A dropdown menu is displayed above the input field, titled "Select Age" with a checkmark icon. It contains the following options:

- Under 18
- 18 to 30
- 31 to 45
- 46+

Tell us a little about yourself:

I agree to the [terms of service](#)

Requirements:

1. Use only HTML to make identical format
2. Save the file in your Activity 1 folder and name it “index.html”
3. Use `<title>Activity 1</title>` in the head
4. Use `<label>` to create labels for all input/menu fields
5. Use the `<tags>` covered in the previous slides
6. Terms of Service opens in a new tab

Note:

- You will need to figure out how to make the checkbox and the Sign Up button at the bottom

Solution:

- you can find the full solution on the next pages. Please **DO NOT** check this site until you have completely done this on your own

Activity 1 Solution

```
index.html

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5
6     <title>Create an Account</title>
7   </head>
8
9   <body>
10
11   <h1>Create an Account</h1>
12
13   <form>
14
15     <p>
16       <label>Email:</label>
17       <br>
18       <input type="text" name="email">
19     </p>
20
21     <p>
22       <label>Password:</label>
23       <br>
24       <input type="password" name="password">
25     </p>
26
27     <p>
28       <label>Age:</label>
```

```
29       <br>
30       <select name="age">
31         <option value="">Select Age</option>
32         <option value="group1">Under 18</option>
33         <option value="group2">18 to 30</option>
34         <option value="group3">31 to 45</option>
35         <option value="group4">46+</option>
36       </select>
37     </p>
38
39     <p>
40       <label>Tell us a little about yourself:</label>
41       <br>
42       <textarea name="about" rows="5" cols="50"></textarea>
43     </p>
44
45     <p>
46       <input type="checkbox" name="accepts_tos" value="yes"> I agree to the
47       <a href="https://www.delidded.com/html-css-practice-test/" target="_blank">terms of service</a>
48     </p>
49
50     <p>
51       <input type="submit" value="Sign up">
52     </p>
53   </form>
54 </body>
55 </html>
```

Activity 1 Challenges

Once you have checked your code with your partner, try the challenge.

- Duplicate your index.html file from Activity 1 into a new folder named “Activity 1 Challenge”

Challenges:

1. have example text in the Email, Password and “Tell us a little about yourself”
2. put a black double border around all of the input fields (you will end up with a bunch of boxes, not just one)
3. change the background of each of the boxes above to different colours
4. Put the checkbox on the right side of the “I agree with the terms of service”
5. Center the “Sign Up” button

CSS

- CSS is used to “style” your sheets
- best practice is to name your CSS file “styles.css”
- quick comment Ctrl + /

```
styles.css
1 /* this is my stylesheet */
2 h1{
3     color: red;
4 }
5 p{
6     font-size: 16px;
7 }
8 
```

CSS

The general format of CSS uses a **selector** followed by **properties** which have **values**

Selector – tells CSS what part of the HTML code the properties apply to

properties – exactly what will change

value – what the property is changed to

A screenshot of a dark-themed code editor showing a snippet of CSS. The code is as follows:

```
1 | section {  
2 |   border: 1px solid gainsboro;  
3 |   background-color: white;  
4 |   margin: 30px;  
5 |   padding: 30px;  
6 | }
```

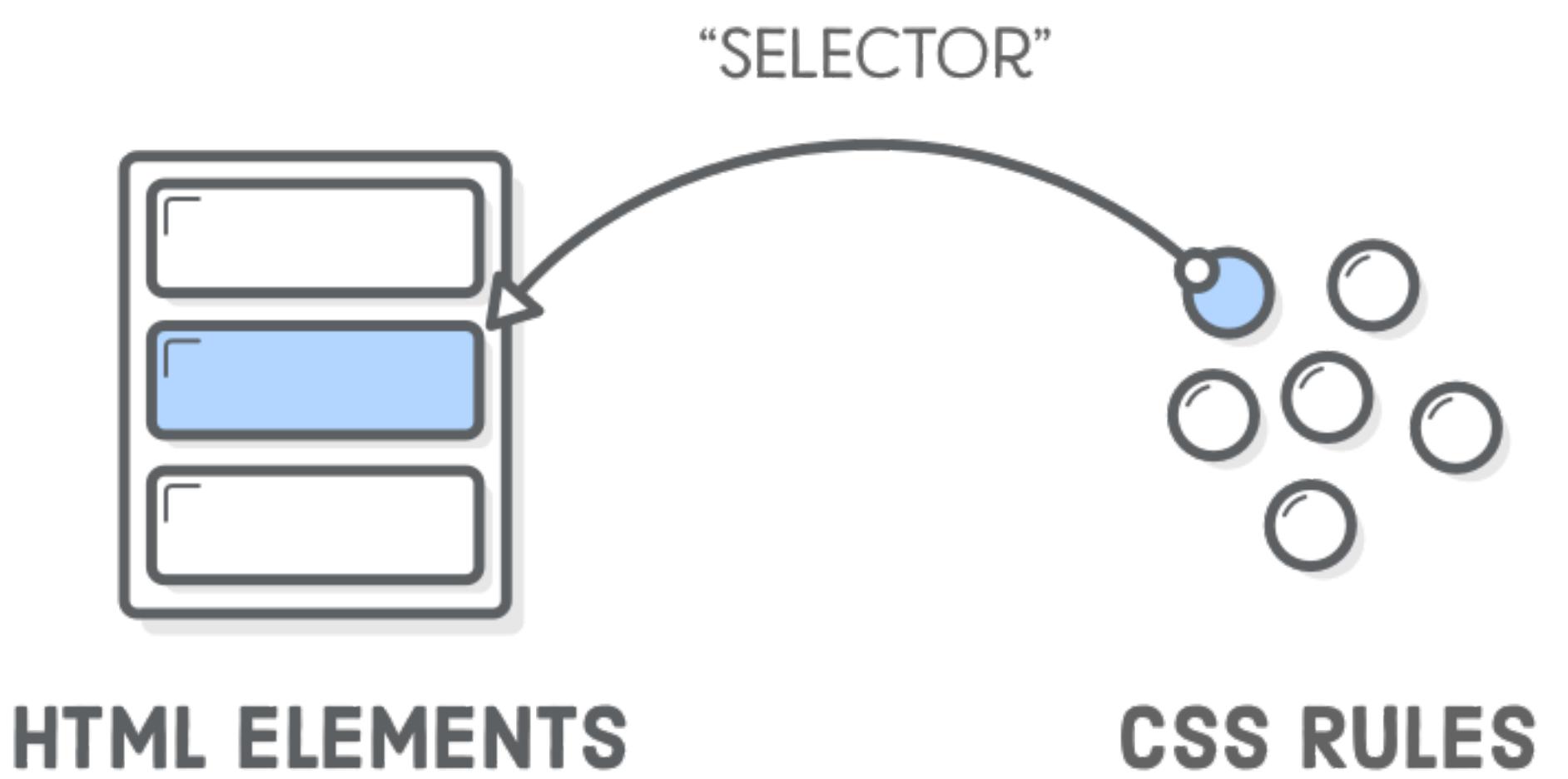
Annotations with cyan arrows and boxes explain the components:

- An arrow points from the word "section" to a cyan box labeled "selector".
- An arrow points from the word "border" to a cyan box labeled "property".
- An arrow points from the value "1px solid gainsboro" to a cyan box labeled "value".
- An arrow points from the word "background-color" to a cyan box labeled "property".
- An arrow points from the value "white" to a cyan box labeled "value".
- An arrow points from the word "margin" to a cyan box labeled "property".
- An arrow points from the value "30px" to a cyan box labeled "value".
- An arrow points from the word "padding" to a cyan box labeled "property".
- An arrow points from the value "30px" to a cyan box labeled "value".

Selector Types

There are **three** ways you can select HTML elements to apply style to:

1. Element type
2. Class
3. Identifier



Class Selectors

Classes are way of **grouping** items that should or behave look the same way.

They are a **powerful** tool that give you complete style control over the elements on your page.



Class Selectors

Classes allow you to link your HTML to your CSS with custom values

This removes the limitation of styling your pages using only the built-in element types.

```
1 <div class="about">  
2   <h2>About</h2>  
3   <p>This is our website.</p>  
4 </div>
```

```
1 .about {  
2   background-color: red;  
3 }  
4  
5 .about h2 {  
6   font-size: 24px;  
7 }
```

The `class` Attribute

A class attribute can be added to any HTML element.

Its value is defined by you.

An HTML element can be more than one class, and each is separated by a space.

```
1 <img class="photo" />
2
3 <h2 class="title">Title</h2>
4
5 <div class="post post-featured">
6   <h2>Post title</h2>
7   <p>Post description</p>
8 </div>
```

The **class** Selector

Class declarations are written using a period `.` followed by the class name

Declarations for a class will apply to every element that has that class attribute in the HTML

Otherwise they work exactly the same as **type** selectors

```
1 .blog-title {  
2   font-size: 36px;  
3 }  
4  
5 .blog-post h2 {  
6   font-size: 24px;  
7   color: red;  
8 }
```

Class Specificity

Classes are **more specific** than regular type selectors.

The `<h1>` element will be **blue** in this example, even though the `h1` rule comes **after** the `.title` rule.

```
1 | <h1 class="title">Big title</h1>
```

```
1 | .title {  
2 |   color: blue;  
3 | }  
4 |  
5 | h1 {  
6 |   color: red;  
7 | }
```



Class Naming

CSS has naming conventions just like in HTML.

When choosing a **class name**, choose something descriptive, functional and concise.

Spaces are **not** allowed.

These are all different!

```
1 | .class-name { }  
2 | .class_name { }  
3 | .className { }  
4 | .ClassName { }
```

Class names are **case sensitive**.

Class Naming

Use this for now!

Dashed case

. class-name

Camel case

. className

Snake case

. class_name

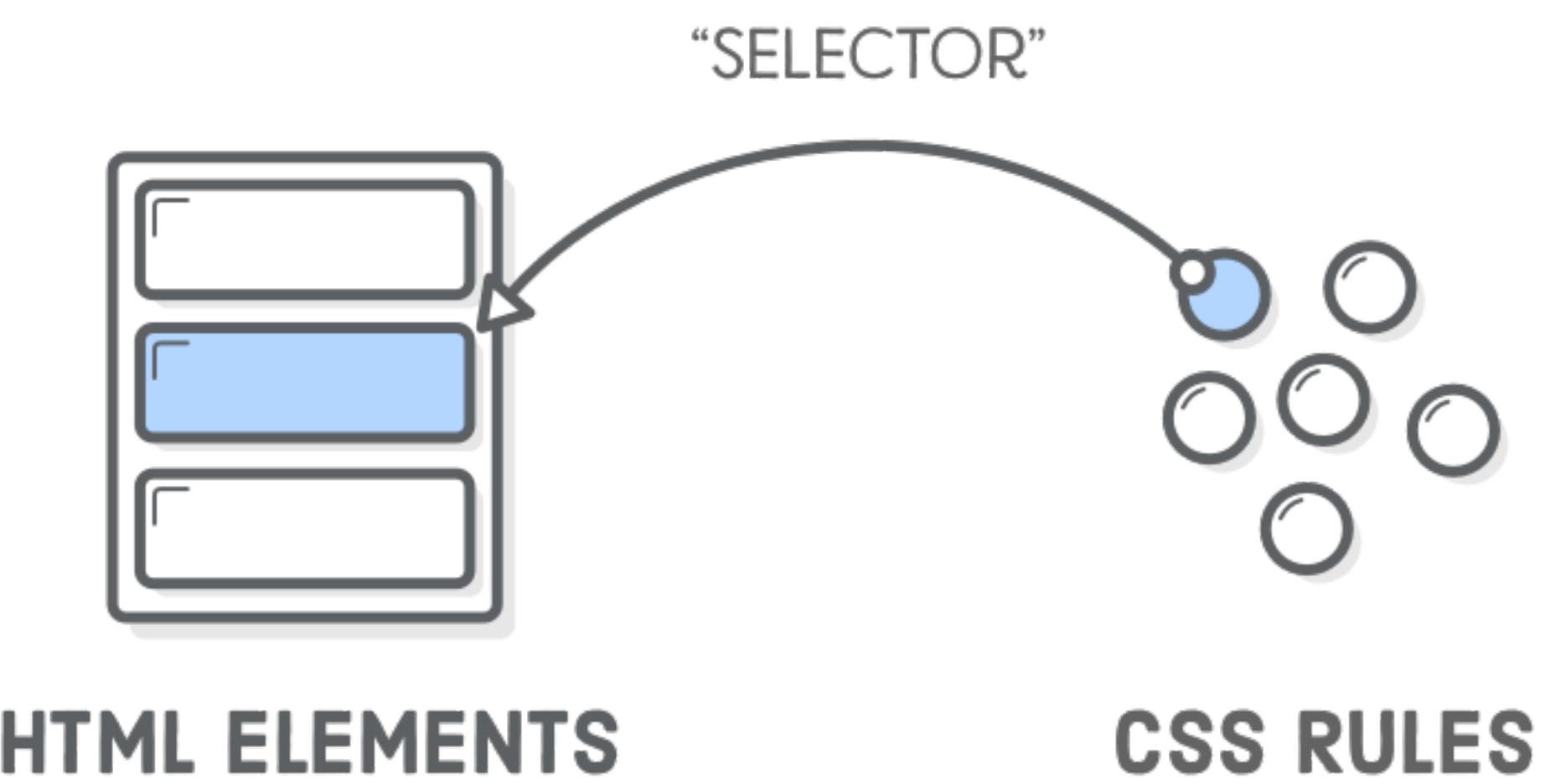
Pascal Case

. ClassName

Selector Types

There are **three** ways you can select HTML elements to apply style to

1. Element type
2. Class
3. Identifier



ID Selectors

Identifiers can style elements in the same way as classes

IDs are denoted using the hash symbol **#** plus the name (just like internal link)

The **id** attributes can only be used **once per page**

```
1 <div id="about">  
2   <h2>About</h2>  
3   <p>This is our website.</p>  
4 </div>
```

```
1 #about {  
2   background-color: red;  
3   color: white;  
4 }
```

Best Practices

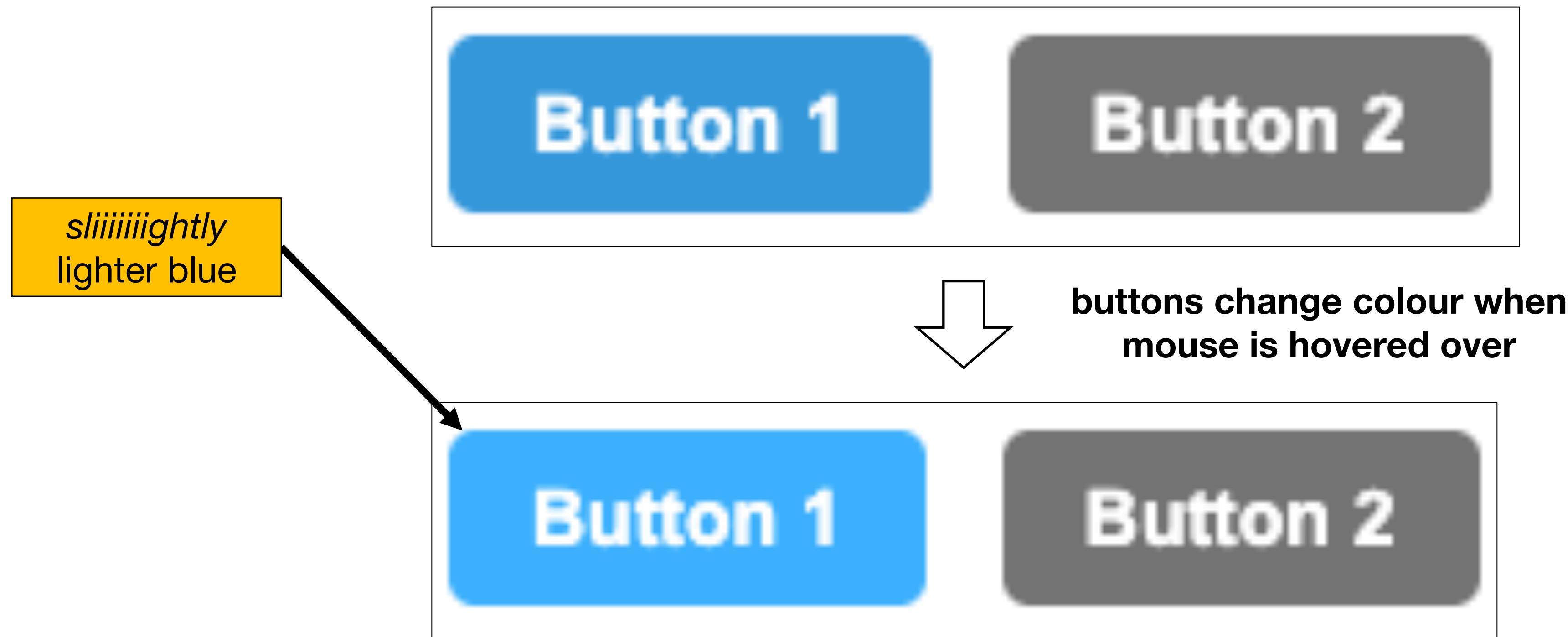
ID selectors are finicky and prone to creating bugs.

Type and **class** selectors are much more reliable, so try to use these where possible.



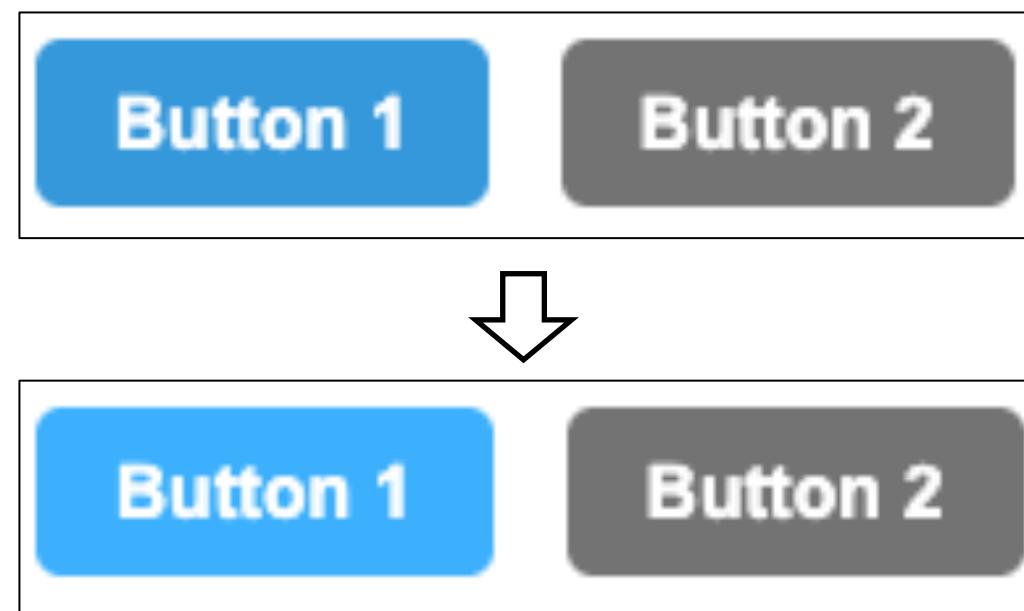
Activity 2 Introduction

Create the following webpage using HTML and CSS



Activity 2 Instructions

Create the following webpage using HTML and CSS



when the mouse is hovered over the buttons they should change colour

Requirements:

1. Use HTML and CSS to make these buttons change colour when you hover over them. You can choose the colours and text style.
2. Save the files in your Activity 2 folder
3. Use <title>Activity 2</title> in the head
4. Use classes only to do this, do not use id's

Note:

- make sure there is space between the buttons and they have rounded corners

Solution:

- you can find the full solution on the next pages. Please **DO NOT** check this site until you have completely done this on your own

Activity 2 Solution

```
index.html | styles.css
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <link rel = "stylesheet" href="styles.css"> |
6   <title>Activity 2</title>
7 </head>
8
9 <body>
10 <input type="button" class="btn" value="Button 1">
11 <input type="button" class="btn gray" value="Button 2">
12 </body>
13
14 </html>
```

```
index.html | styles.css
```

```
1 .btn {
2   /* Add some space to the right of each button (so they don't touch) */
3   margin-right: 10px;
4   /* Add some space around the text inside the button */
5   padding: 8px 15px;
6
7   color: white;
8   background-color: #3498db;
9
10  font-weight: bold;
11  font-family: Arial, sans-serif;
12  font-size: 14px;
13
14  border: none;
15  /* rounded corners */
16  border-radius: 5px;
17 }
18
19 /* Style differently when mouse is hovering over button */
20 .btn:hover {
21   background-color: #3cb0fd;
22   /* change the mouse cursor */
23   cursor: pointer;
24 }
25
26 .btn.gray {
27   background-color: #737373;
28 }
29
30 .btn.gray:hover {
31   background-color: #8c8c8c;
32 }
```



Activity 2 Challenge

Once you have checked your code with your partner, try the challenge.
Duplicate your index.html file from Activity 2 into a new folder named “Activity 2 Challenge”

Challenges:

1. animate the buttons to have arrows as shown below
2. See what else you can animate the button to do!



Button 1

Gathering Time



Gathering Time is a time for check-in.



First Nations
Technology Council

Class Wrap-up

How was your day?

How's your website coming along?

Are you excited to add some colour?

