Accessing HTML Tags with CSS and JavaScript

Created on: 8 April 2013

Part 17 of the Arduino Ethernet Shield Web Server Tutorial

How to access HTML elements (tags) with CSS and JavaScript in order to apply CSS styles to the elements and manipulate the elements using JavaScript. Shows how to reference HTML tags by using ID and class names.

Running the Examples

Each of the examples below can be copied from this page and saved as an HTML file (e.g. index.htm). The file can then be loaded to the Arduino if desired by following the example in part 4 of this tutorial.

Alternatively each example can be opened from the computer in a web browser without using the Arduino.

Donate to Starting Electronics

Donate

Accessing HTML Elements

HTML elements need to be accessed so that CSS styles can be applied to them and so that we can get hold of these elements with JavaScript in order to change or manipulate them.

Accessing HTML Elements in CSS

There are three main methods of accessing or referring to HTML elements in CSS:

- By referring to the HTML element by its HTML tag name, e.g. **p** to refer to the paragraph HTML tag –
- By using an ID, e.g. Some text
- By using a class, e.g. Some text

A combination of the above methods can also be used to access an HTML element.

We will now look at an example of each of the above methods.

Reference by HTML Element Name

This method was already demonstrated in part 12 of this tutorial and was used in part 16 (the previous part).

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino SD Card Web Page</title>
        <style type="text/css">
            p {
                font-family: arial, verdana, sans-seri
                font-size: 12pt;
                color: #6B6BD7;
            }
        </style>
    </head>
    <body>
        <h1>Arduino SD Card Page with CSS</h1>
```

```
p {
    font-family: arial, verdana, sans-serif;
    font-size: 12pt;
    color: #6B6BD7;
}
```

Styles can be applied to other HTML elements in the same way by referring to their HTML tag names, e.g. h1, h2, h3, div, span, b, etc.

This method is used to set the default style for HTML elements on a page. It is possible to override these default styles by giving an HTML element an ID or class name.

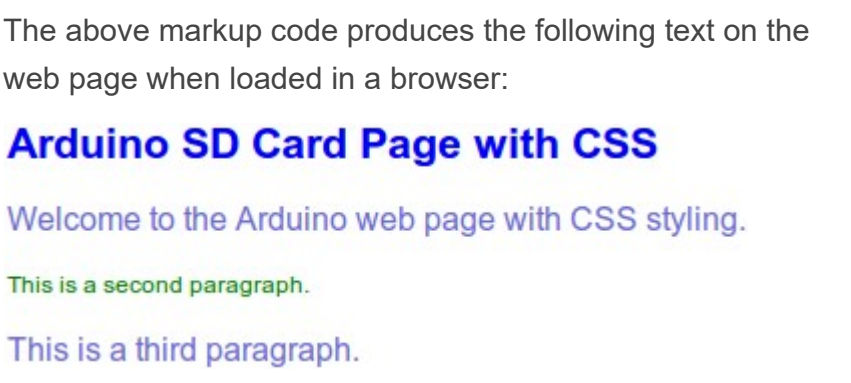
Reference by ID

An ID of a specific name can only be used once on a web page. The name of the ID is chosen by the person writing the HTML and CSS. An ID would normally be used for something like a menu that occurs only once per page. Using an ID also allows JavaScript to access the element using the unique ID.

This example code shows the use of an ID:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino SD Card Web Page</title>
        <style type="text/css">
            h1 {
                font-family: arial, verdana, sans-seri
                font-size: 16pt;
                color: blue;
            }
            p {
                font-family: arial, verdana, sans-seri
                font-size: 12pt;
                color: #6B6BD7;
```

The above markup code produces the following text on the web page when loaded in a browser:



As can be seen in the above markup, when an HTML element is referred to by its ID, the # character is used before its name in the CSS style:

```
#green_small {
    font-size: 9pt;
    color: green;
}
```

This style is then applied to the HTML element with that ID:

```
<p id="green_small">This is a second paragraph.</p>
```

Because it is an ID, it may not be used again on the web page – other IDs may be used, but each must have a unique name.

Also notice that the ID overrides the **p** default style that applies to all paragraphs. The other paragraphs that do not have an ID are then formatted with the default paragraph style.

Reference by Class

A class works the same way as an ID, except that it may be used more than once on a web page. A class uses a dot (.) in front of the class name in the CSS style to show that it is referring to a class and not an ID or HTML element.

The following example shows how to apply a CSS style to HTML elements that have class names.

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Arduino SD Card Web Page</title>
    <style type="text/css">
        h1 {
            font-family: arial, verdana, sans-se
            font-size: 16pt;
            color: blue;
        }
        p {
            font-family: arial, verdana, sans-se
            font-size: 12pt;
```

The above markup produces the following web page:

Arduino SD Card Page with CSS

Welcome to the Arduino web page with CSS styling.

This is a second paragraph.

This is a third paragraph.

Notice that the class style has been applied both to the HTML **span** tag as well as the HTML **p** (paragraph) tag. The style applied to the HTML elements that have class names overrides the default style.

The CSS style applied to the HTML elements with the class name **red_big** can be seen here using the dot to show that it is referring to a class:

```
.red_big {
    font-size: 14pt;
    color: red;
}
```

Mixing Access Methods

The same CSS style can be applied to more than one HTML element. HTML elements can also be referred to more specifically, e.g. every paragraph inside a div element. This example demonstrates:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino SD Card Web Page</title>
        <style type="text/css">
            h1, p {
                font-family: arial, verdana, sans-se
                color: cyan;
            }
            h1 {
                font-size: 16pt;
            }
            p {
                font-size: 12pt;
            }
```

When a CSS style has HTML tag names, id names or class names separated by a comma, then the specified style applies to each of the elements.

The following CSS style applies to both the h1 tag and the p tag and specifies the font family for both as well as the colour as cyan:

```
h1, p {
    font-family: arial, verdana, sans-serif;
    color: cyan;
}
```

In the above HTML with CSS listing, the sizes of the h1 tag and paragraph are then specified separately.

When the CSS style has HTML tag names, id names or class names that are not separated by a comma, then the style is applied to the elements as they occur inside each other. In the above HTML with CSS listing, a CSS style is applied to every paragarph that occurrs in an HTML div element. The style is shown here:

```
div p {
    font-style: italic;
    letter-spacing: 5px;
}
```

The above style makes the font italic and separates paragraph letters by 5 pixels in every paragraph that occurs inside an HTML div.

Accessing HTML Elements in JavaScript

HTML ID and class names allow JavaScript to get hold of these HTML tags to modify or manipulate them.

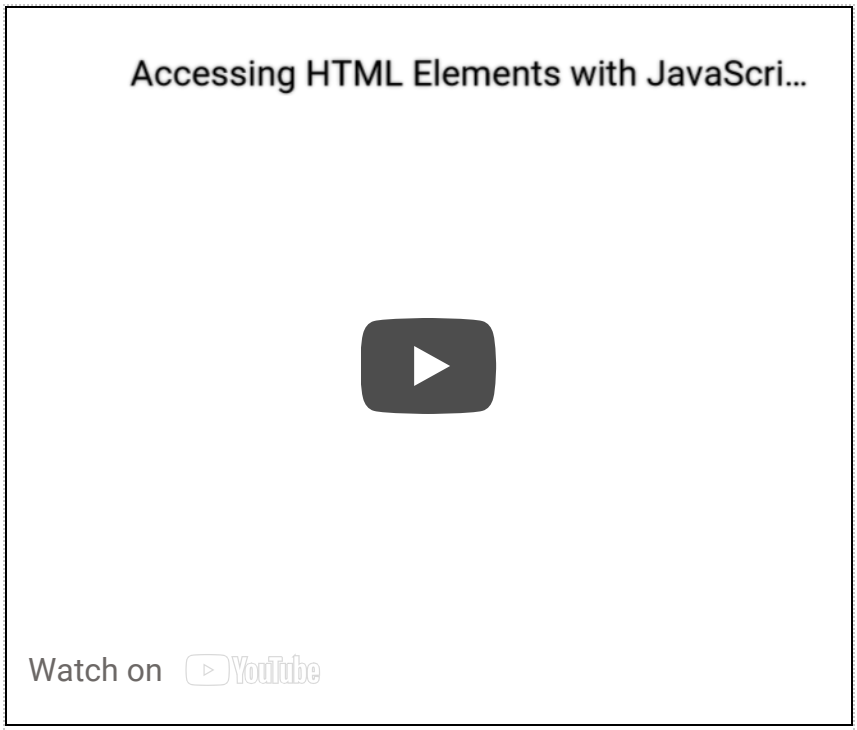
If you have been following each part of this tutorial, you will recognize these methods of accessing HTML elements using JavaScript.

Accessing HTML Elements with ID Names in JavaScript

The following example shows how to access an HTML element that has an ID name from JavaScript:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino SD Card Web Page</title>
        <script>
            var btn_count = 0;
            function ButtonCount()
            {
                btn_count++;
                document.getElementById("btn_clicks").
            }
        </script>
    </head>
    <body>
        <h1>Arduino SD Card Page with JavaScript</h1>
        <button type="button" onclick="ButtonCount()">
        <p>The button has been clicked <span id="btn_c
    </body>
</html>
```

This video shows the above web page with JavaScript code running in a web browser:



Can't see the video? View on YouTube →

The JavaScript function ButtonCount() is run every time that the button is clicked. The function adds 1 to the variable **btn_count** every time that the function is run.

The JavaScript gets hold of the HTML span tag with the ID **btn_clicks** by using the following code:

```
document.getElementById("btn_clicks").innerHTML = btn_
```

Which accesses the span in this line of HTML:

```
<p>The button has been clicked <span id="btn_clicks">0
```

The current value of **btn_count** is then inserted into the HTML span.

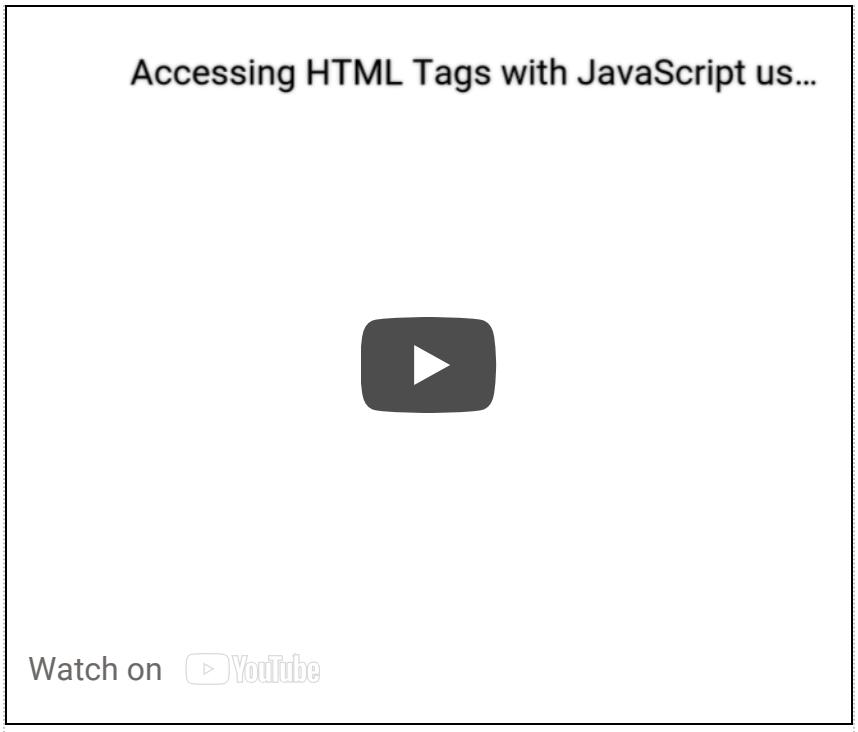
Remember that only one element on the page can use the ID with the name **btn_clicks**.

Accessing HTML Elements with Class Names in JavaScript

The following code accesses HTML tags that have the same class name:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino SD Card Web Page</title>
        <script>
            var btn_count = 0;
            var btn_count_0 = 0;
            var num_classes = 0;
            function ButtonCount()
            {
                btn_count++;
                document.getElementsByClassName("btn
                document.getElementsByClassName("btn
                btn_count_0++;
                // get the number of btn_clicks clas
                document.getElementById("btn_classes
                                        document.getElem
            }
        </script>
    </head>
    <body>
```

This video shows the above HTML page with JavaScript code operating:



Can't see the video? View on YouTube →

Two HTML span tags have the same class name (btn_clicks) and are accessed in the JavaScript code as shown here:

```
document.getElementsByClassName("btn_clicks")[0].inner
document.getElementsByClassName("btn_clicks")[1].inner
```

The number of HTML tags on the page that use this class name can be obtained using this code:

```
document.getElementsByClassName("btn_clicks").length;
```

Each tag with this class name can then be accessed in a loop if needed.

The HTML tags with the same class name are accessed on the page in the order that they occur from top to bottom.

← Go back to Part 16 Go to Part 18 →

Comments

Login

There are no comments posted yet. Be the first one!

Post a new comment

Enter text right here!

Comment as a Guest, or login:

Name | Email | Website (optional)

Displayed next to your comments. | *Not displayed publicly.* | *If you have a website, link to it here.*

Subscribe to None

Submit Comment

© 2012 – 2021, Starting Electronics