|  |
| --- |
| **Set 21: Creating DOM Elements** |

|  |
| --- |
| **Skill 21.01: Write code to add an element to an HTML page**  **Skill 21.02: Write code to remove or hide an HTML element**  **Skill 21.03: Reference media files**  **Skill 21.03: Write code to add an image**  **Skill 21.04: Write code to add a hyperlink**  **Skill 21.05: Write code to modify an attribute** |

|  |
| --- |
| **Skill 21.01: Write code to add an element to an HTML page** |

**Skill 21.01 Concepts**

In our previous lesson about the DOM we learned how to select and modify existing elements on our webpage. But, just as the DOM allows scripts to modify existing elements, it also allows for the creation of new ones. The *.createElement(tagName)* method creates a new element based on the specified tag name. However, it does not append it to the document. It creates an empty element with no inner HTML.

To create an element and add it to the web page, you must assign it to an element that already exists on the DOM. We call this process appending. The *.append* method will add the element as the last child node.

The following code (1) creates a new paragraph element and assigns the element to a variable called *myParagraph*, (2) adds text to the new element’s *innerHTML*, and (3) appends it to the body of the document:

|  |
| --- |
| var myParagraph = document.createElement('p');  myParagraph.innerHTML = "The text inside paragraph";  document.body.append(myParagraph); |

In our previous lessons we have learned how to control elements on the page by accessing their id’s. In the example above, the paragraph we created does not have an *id* associated with it. To specify an id we can use the id property as follows,

|  |
| --- |
| myParagraph.id = "p1"; |

Now that we have associated an *id* to our new element, we can modify it using the same techniques we learned before.

In the below example, we change the text to read, “Hello DOM!”

|  |
| --- |
| document.getElementById("p1").innerHTML = "Hello DOM!"; |

**[Skill 21.01 Exercise 1](https://hpluska.github.io/APCompSciPrinciples/ticketOutTheDoor/set22/Set22TicketOutTheDoorAPCompSciPrinciples.pdf)**

|  |
| --- |
| **Skill 21.02: Write code to remove or hide an HTML element** |

**Skill 21.02 Concepts**

In addition to modifying or creating an element from scratch, the DOM also allows for the removal of an element. The *.remove* method removes a specified element.

|  |
| --- |
| var e = document.getElementById("someElement");  e.remove(); |

If you want to hide an element, the *.hidden* property allows you to hide it by assigning it as true or false. Hiding an element does not remove an element however. The element is still on the page, just now viewable by the user.

|  |
| --- |
| var e = document.getElementById("someElement");  e.hidden = true; |

To show a hidden element, simply set the hidden property to false,

|  |
| --- |
| var e = document.getElementById("someElement");  e.hidden = false; |

**[Skill 21.02 Exercise 1](https://hpluska.github.io/APCompSciPrinciples/ticketOutTheDoor/set22/Set22TicketOutTheDoorAPCompSciPrinciples.pdf)**

|  |
| --- |
| **Skill 21.03: Reference media files** |

**Skill 21.03 Concepts**

The images and videos you include with your page are separate files and will need to be referenced correctly to be displayed.

Images and videos are referenced using src attribute - short for source. This tells the tag the location of the image or video to load.

Consider the following file structure. Media is the name of the folder. And, inside the folder we have a file called *Index.html* and an image called *Frog.jpg*. The following code could be used to reference the *Frog.jpg* image from the *Index.html* page.

|  |  |
| --- | --- |
| **Media** | src = “Frog.jpg” |
| Index.html  Frog.jpg |

Now consider an example where the *Frog.jpg* we are trying to reference is stored in a directory that is different than *Index.html*. In the file structure below, we have created a directory called *Images* inside the *Media* folder and placed the *Frog.jpg* image inside of it. The following code could be used to reference the *Frog.jpg* image from the *Index.html* page.

|  |  |
| --- | --- |
| **Media** | src = “Images/Frog.jpg” |
| Index.html   |  | | --- | | **Images** | | Frog.jpg | |

Now consider the situation below. *Media* and *Images* are both separate directories in the *MyWebsite* directory. Inside the *Media* directory we have an *Index.html* page and inside the *Images* directory we have our image *Frog.jpg* we want to reference. To do this, we must first “backout” of the *Media* directory, then enter the Images directory. The “. .” syntax is used to backout of a directory.

|  |  |  |
| --- | --- | --- |
| **MyWebsite** | | src = “../Images/Frog.jpg” |
| **Media** | **Images** |
| Index.html | Frog.jpg |

**[Skill 21.03 Exercise 1](https://hpluska.github.io/APCompSciPrinciples/ticketOutTheDoor/set22/Set22TicketOutTheDoorAPCompSciPrinciples.pdf)**

|  |
| --- |
| **Skill 21.03: Write code to add an image** |

**Skill 21.03 Concepts**

|  |
| --- |
| **Skill 20.03: Write code to modify an attribute** |

**Skill 20.03 Concepts**

Attributes are values that contain additional information about HTML elements. They usually come in name/value pairs, and may be essential depending on the element. Consider the *img* tag below. The image tag below requires the *src* attribute to indicate the location of the image.

|  |
| --- |
| <img id= “myImage” src = “path/to/my/image.jpg”> |

The *a* tag below is another example. In this example, the *href* attribute indicates the path to the link and the target attribute indicates where the link should open – in this case, “BLANK” indicates the page should open in a new page.

|  |
| --- |
| <a href = “path/to/my/page.html” target = “\_BLANK”> |

The code below could be used to modify the *img* tag above,

|  |
| --- |
| var image = document.getElementById('myImage');  image.getAttribute('src'); // returns path/to/my/image.jpg  image.removeAttribute('src'); // removes the src attribute  image.setAttribute('src', 'diff.jpg'); // changes the src attribute |

The attribute for any element can be accessed by using the dot notation. Below is another example of how we can access and reassign the path of the image element above.

|  |
| --- |
| var image = document.getElementById('myImage');  image.src = ‘diff.jpg’; |

**[Skill 20.03 Exercises 1](https://hpluska.github.io/APCompSciPrinciples/ticketOutTheDoor/set22/Set22TicketOutTheDoorAPCompSciPrinciples.pdf)**