1. Introduction

Use JS and the DOM the create, delete, and manipulate DOM elements.

Project repository: <https://github.com/udacity/course-JS-and-the-DOM>

This is just the Udacity Homepage, not the full website.

1. Update Existing Page Content

innerHTML and textContent and innerText

Both textContent and innerText allow the manipulation of text.

document.getElementsByClassName("card")[0]; gets the first care element.

const nanodegreeCard = document.querySelector('.card');

This means that every element has an .innerHTML property. This property, as it's rightly named, represents the markup of the element's content. We can use this property to:

* get an element's (and all of its descendants!) HTML content
* set an element's HTML content

.innerHTML is a property

If you look at this in the console, it looks like a string. Technically, however, what it returns is called a DOMString.

Passing any text that looks like HTML to the .textContent property will still be displayed as text. It will not be displayed as HTML when the element is rendered.

textContent ignores CSS formats, where innerText displays text visually to user.

1. Add New Page Content

* createElement()
* createTextNode()
* appendChild()
* insertAdjacentHTML()

The .createElement() method is a method on the document object.

<https://developer.mozilla.org/en-US/docs/Web/API/Document/createElement>

document.CreateElement(‘p’)

// create a brand new <span> element

const newSpan = document.createElement('span');

// select the first (main) heading of the page

const mainHeading = document.querySelector('h1');

// add the the <span> element as the last child element of the main heading

mainHeading.appendChild(newSpan);

.appendChild() Needs An Element!

The .appendChild() method will move an element from its current position to the new position.

<https://developer.mozilla.org/en-US/docs/Web/API/Element/insertAdjacentHTML>

1. Remove Page Content

* .style.<prop>
* .cssText()
* .setAttribute()
* .className
* .classList

1. Style Page Content

Only the width styling will be in the element's style attribute. The .style.cssText will overwrite anything that's already in the .style attribute (which removes the color styling). The textDecoration rule is misspelled and should be text-decoration, so it gets dropped.

1. Lesson summary

Update, add, remove, and style content.