| Sprint 10, Assignment 3.12 Please also update the doc name with correct numbers. | |
| --- | --- |
| Assignment type: JS Interactive | |
| Assignment name  Can remain the same as the assignment topic name, or…  can be phrased in the active/imperative voice (depending on the task), e.g. “Create your own X” | “Add Button Events & Input Conditionals” |
| Description  The essential tasks a student must perform in order to complete the assignment  ?+ a general, colorful introduction to the assignment? | No sign-up form is complete without a button. Add one to yours, and give it some style! This is a great chance to show off to your client what kind of CSS and Javascript skills you've got (and feel free to do more styling than we supply, here). Also, the client had a final note for the week: they'd like you to make sure the user is prompted if they mistakenly leave any field blank. |
| Link to full code file (Answer)  Push a full code file to our [GitHub repo](https://github.com/bitdegree-foundation/academy-assignments-code-files) for this sprint’s assignment and paste a link to that file here. Please write the file in a concise and clear format, according to the “[example-js-code-file](https://github.com/bitdegree-foundation/academy-assignments-code-files/blob/master/example-js-code-file.js)” found in our repo. | (Upload) |

| # | Step  Write each small step of the task. These assignments should be doable by students on their own (without BitDegree-instructor help). It can be any number of steps, but keep them pretty clear and separate (don’t combine 2 steps into 1 step). | 1x hint  Write brief text or partial code that will help the student figure out how to either move forward or completely accomplish the given step. | Output Expected code that our platform will be able to run & check against student submissions |
| --- | --- | --- | --- |
| 1 | (In your .html file …) Add a button at the bottom of your form with both an ID and a type of “submit”. Give it a value/the inner text of “Submit!” |  | …  <input *type*="submit" *id*="submit" *value*="Submit!" /><br />  </form> |
| 2 | (In your .js file …)  Store this button element in a variable named submitionButton, and add mouseover and away listeners that will log the user’s behavior |  | const submitionButton = document.querySelector("#submit");  submitionButton.addEventListener("mouseenter", function(){console.log("User considering SUBMITTING <button> ...");});  submitionButton.addEventListener("mouseleave", function(){console.log("User DIDN'T submit <button> ...");}); |
| 3 | Add additional arguments to your mouse on and mouse off listeners (for the submit button): make the button slightly bigger, change its background color, and text color when hovered-over (and toggled back to normal when not) | Hint: chain attributes of the button using "."  Hint: use .style.background, .style.color, and .style.font-size | const submitionButton = document.querySelector("#submit");  submitionButton.addEventListener("mouseenter", function(){  console.log("User considering SUBMITTING <button> ...");  submitionButton.style.color = "white";  submitionButton.style.background = "rgb(32,32,32)";  submitionButton.style.fontSize = "18px";  });  submitionButton.addEventListener("mouseleave", function(){  console.log("User DIDN'T submit <button> ...");  submitionButton.style.color = "black";  submitionButton.style.background = "rgb(232,232,232)";  submitionButton.style.fontSize = "13.5px";  }); |
| 4 | Add an onclick listener that will log that the button was pressed, and alert a message like “Welcome! We're delighted to have you as part of the family!!" when pushed | Hint: use alert(“”) | submitionButton.onclick = function(){  console.log(`Submission Button clicked!`);  alert(`Welcome! We're delighted to have you as part of the family!!`);  } |
| 5 | Add an event listener for the 'form' element that will, on "submit", prevent the default behavior of the browser; log when this happens. | Hint: use preventDefault  Hint: pass the event through the function and target it | form.addEventListener("submit", function(e){  e.preventDefault();  console.log("Prevented default browser behavior -");  }); |
| 6 | Add a conditional (to this listener) that will check whether any field has been left blank, and if so, log "User left a field blank" | Hint: you want to check whether the given field's "value" is the equivalent of an empty string  Hint: you can chain all the fields together in one conditional if you seperate the statements with 'or' characters (" || ") | form.addEventListener("submit", function(e){  e.preventDefault();  console.log("Prevented default browser behavior -");  if (  fName.value === "" ||  lName.value === "" ||  address1.value === "" ||  address2.value === "" ||  userCity.value === "" ||  userState.value === "" ||  userPhone.value === ""  ){  console.log("User left a field blank");  const errorMessage = document.createElement("p")  };  }); |
| 7 | Declare an error message in your conditional, and store in it a new "paragraph element" | Hint: use createElement  Hint: you're going to be appending this as html, so make the element a paragraph, h1 etc | form.addEventListener("submit", function(e){  e.preventDefault();  console.log("Prevented default browser behavior -");  if (  fName.value === "" ||  lName.value === "" ||  address1.value === "" ||  address2.value === "" ||  userCity.value === "" ||  userState.value === "" ||  userPhone.value === ""  ){  console.log("User left a field blank");  const errorMessage = document.createElement("p")  };  }); |
| 8 | Style your error message's color, font size, and text. Log and test your error message. | Hint: this is another time you can chain the property "style" with subsequent css properties  Hint: use .style.color, .style.fontweight, and .innerText | form.addEventListener("submit", function(e){  e.preventDefault();  console.log("Prevented default browser behavior -");  if (  fName.value === "" ||  lName.value === "" ||  address1.value === "" ||  address2.value === "" ||  userCity.value === "" ||  userState.value === "" ||  userPhone.value === ""  ){  console.log("User left a field blank");  const errorMessage = document.createElement("p")  errorMessage.style.color = "red";  errorMessage.style.fontWeight = 800;  errorMessage.innerText = "Sorry - no fields can be left blank!";  console.log(errorMessage);  };  }); |
| 9 | Append the errorMessage to the form element | Hint: target the "form" element and use the append method | form.addEventListener("submit", function(e){  e.preventDefault();  console.log("Prevented default browser behavior -");  if (  fName.value === "" ||  lName.value === "" ||  address1.value === "" ||  address2.value === "" ||  userCity.value === "" ||  userState.value === "" ||  userPhone.value === ""  ){  console.log("User left a field blank");  const errorMessage = document.createElement("p")  errorMessage.style.color = "red";  errorMessage.style.fontWeight = 800;  errorMessage.innerText = "Sorry - no fields can be left blank!";  console.log(errorMessage);  form.append(errorMessage);  };  }); |
| ... |  |  |  |