| Sprint 10, Assignment 5.5 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” | “Change given code & prevent it from the bubbling” |
| Description  The essential tasks a student must perform in order to complete the assignment  ?+ a general, colorful introduction to the assignment? | After reading some articles on event propagation, it seems like a good idea to test it out. Create some html that represents a series of nested elements (say, colored boxes named grand-parent, parent etc) and then add a click listener that will trigger whenever anything is clicked. Then, create a second listener that will change the background color starting at a nested element (so, anything other than the most grand-parent one) and see if you can "turn it off" for specific children. Become a bubble master! ... it should really pay off at parties! |
| 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 |
| --- | --- | --- | --- |
| 0 | Starter Code / STEP 0 |  | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style></style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 1 | In the body of your HTML, add a div with the class of "grand-parent" and an inline paragraph tag (with inner text of grand-parent) | Hint: <div class="grand-parent"><p>grand-parent</p></div> | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style></style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body>  <div *class*="grand-parent">  <p>grand-parent</p>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 2 | Create a second set of div and inline paragraph tags \*nested\* within the grand-parent div. Give them a class and inner text (respectively) of "parent" | Hint: ...<p>grand-parent</p><div class="parent"><p>parent</p></div></div> | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style></style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body>  <div *class*="grand-parent">  <p>grand-parent</p>  <div *class*="parent">  <p>parent</p>  </div>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 3 | Repeat step 3 and create a third nested set of divs and paragraph tags with class and inner text of "child" |  | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style></style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body>  <div *class*="grand-parent">  <p>grand-parent</p>  <div *class*="parent">  <p>parent</p>  <div *class*="child">  <p>child</p>  </div>  </div>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 4 | Create three sets of divs - all nested under the "child" div (and \*after\* its paragraph tags - not nested within them) - with the class of "grand-child". Give them unique inner text, like "grand-child 1", "grand-child 2", and "grand-child 3" | Hint: ...<p>child</p><div class="grand-child">grand-child 1</div><div class="grand-child">grand-child 2</div><div class="grand-child">grand-child 3</div>... | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style></style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body>  <div *class*="grand-parent">  <p>grand-parent</p>  <div *class*="parent">  <p>parent</p>  <div *class*="child">  <p>child</p>  <div *class*="grand-child">grand-child 1</div>  <div *class*="grand-child">grand-child 2</div>  <div *class*="grand-child">grand-child 3</div>  </div>  </div>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 5 | Add styling to your elements in the "style" section of your HTML document's header; pick different background colors for each, and then add the following additional css:  border-width: solid 50px;  width: 200px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px; |  | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style>  .grand-parent {  background: yellow;  border-width: solid 50px;  width: 400px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .parent {  background: aqua;  border-width: solid 50px;  width: 300px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .child {  background: lime;  border-width: solid 50px;  width: 200px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .grand-child {  background: rgb(255, 0, 221);  border-width: solid 50px;  width: 100px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  </style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body>  <div *class*="grand-parent">  <p>grand-parent</p>  <div *class*="parent">  <p>parent</p>  <div *class*="child">  <p>child</p>  <div *class*="grand-child">grand-child 1</div>  <div *class*="grand-child">grand-child 2</div>  <div *class*="grand-child">grand-child 3</div>  </div>  </div>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 6 | Create one more class-specific style to your header named "red" and give it a matching background color | Hint: .red {background-color: red;} | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style>  .grand-parent {  background: yellow;  border-width: solid 50px;  width: 400px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .parent {  background: aqua;  border-width: solid 50px;  width: 300px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .child {  background: lime;  border-width: solid 50px;  width: 200px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .grand-child {  background: rgb(255, 0, 221);  border-width: solid 50px;  width: 100px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .red {  background-color: red;  }  </style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body>  <div *class*="grand-parent">  <p>grand-parent</p>  <div *class*="parent">  <p>parent</p>  <div *class*="child">  <p>child</p>  <div *class*="grand-child">grand-child 1</div>  <div *class*="grand-child">grand-child 2</div>  <div *class*="grand-child">grand-child 3</div>  </div>  </div>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 7 | Add an alert to the body tag that, on click, shows a message like "The bubbling has reached here". Click every element to check 1) that the alert fires for all elements, but 2) that only the elements from "parent"-on-down turn red |  | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style>  .grand-parent {  background: yellow;  border-width: solid 50px;  width: 400px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .parent {  background: aqua;  border-width: solid 50px;  width: 300px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .child {  background: lime;  border-width: solid 50px;  width: 200px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .grand-child {  background: rgb(255, 0, 221);  border-width: solid 50px;  width: 100px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .red {  background-color: red;  }  </style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body *onclick*="alert(`the bubbling has reached here`)">  <div *class*="grand-parent">  <p>grand-parent</p>  <div *class*="parent">  <p>parent</p>  <div *class*="child">  <p>child</p>  <div *class*="grand-child">grand-child 1</div>  <div *class*="grand-child">grand-child 2</div>  <div *class*="grand-child">grand-child 3</div>  </div>  </div>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| 8 | Create a new JavaScript file named 5.5.Assignment.script.js. Store each set of class elements in a variable of the same name (so, const grandParent should contain any element with the class of ".grand-parent") and log them to check your work. | Hint: If you recall from earlier in the sprint, querySelector only grabs the \*first\* match - but you have three grand-children divs ...  Hint: use querySelectorAll | const grandParent = document.querySelector(".grand-parent");  const parent = document.querySelector(".parent");  const child = document.querySelector(".child");  const grandChild = document.querySelectorAll(".grand-child");  console.log(grandParent);  console.log(parent);  console.log(child);  console.log(grandChild); |
| 9 | Add an event listner to your parent element that, on click, will toggle on and off the class of "red". Test whether all child elements turn red on click. | Hint: you will need to target the event parameter | parent.addEventListener('click', (event) => {  event.target.classList.toggle("red")  }); |
| 10 | Stop your paragraph elements (for "parent" and "child") from turning red on click by using stopPropegation to target the click event | Hint: insert event.stopPropegation() into both <p> tags  Note: if done correctly, clicking on the words "parent" and "child" (on your page) should neither turn them red nor trigger the alert. However, you should still be able to get both events by clicking anywhere else in the parent (aqua-colored) or child (lime-colored) boxes | <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <style>  .grand-parent {  background: yellow;  border-width: solid 50px;  width: 400px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .parent {  background: aqua;  border-width: solid 50px;  width: 300px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .child {  background: lime;  border-width: solid 50px;  width: 200px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .grand-child {  background: rgb(255, 0, 221);  border-width: solid 50px;  width: 100px;  margin: 10px;  border: 4px solid black;  font-size: 110%;  text-align: center;  padding: 30px;  }  .red {  background-color: red;  }  </style>  <title>Sprint 10, Assignment 5.5</title>  </head>  <body *onclick*="alert(`the bubbling has reached here`)">  <div *class*="grand-parent">  <p>grand-parent</p>  <div *class*="parent">  <p *onclick*="event.stopPropagation()">parent</p>  <div *class*="child">  <p *onclick*="event.stopPropagation()">child</p>  <div *class*="grand-child">grand-child 1</div>  <div *class*="grand-child">grand-child 2</div>  <div *class*="grand-child">grand-child 3</div>  </div>  </div>  </div>  <script *src*="./5.5.Assignment.script.js"></script>  </body>  </html> |
| ... |  |  |  |