

APCSP\Create Performance Task\create_performance_task.html

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Public Forum Debate</title>
6      <style>
7          body {
8              padding: 15px 0;
9              border: 10px double rgb(115, 147, 179);
10             text-align: center;
11             background:rgb(128, 128, 128);
12             font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida
13 Sans', Arial, sans-serif;
14             color: rgb(1, 37, 0)
15         }
16
17         table {
18             margin-left: auto;
19             margin-right: auto;
20
21         }
22
23         td,
24         th {
25             padding: 10px;
26
27         }
28
29         .customstopwatch {
30             font-size: 150%;
31         }
32     </style>
33     <script>
34         let teamOne
35         let teamTwo
36         let startTime;
37         let stopwatchInterval;
38         let elapsedPausedTime = 0;
39         let speechList = []
40         let number = 1
41         let specificSpeech = 0
42
43         // Description: User chooses their side and then the first speech is displayed
44         // Citation: None
45         // Input/Parameters: Pro or con side
46         // Output/Return: Team one constructive speech stopwatch
47         function chooseSide() {
48             let selected1 = argueTeam1.value
49             let selected2 = argueTeam2.value
50
51             if (selected1 == "Con") {
52                 teamOne = "Con"
53             }
54             else if (selected1 == "Pro") {
```

```

53         teamOne = "Pro"
54     }
55
56     if (selected2 == "Con") {
57         teamTwo = "Con"
58     }
59     else if (selected2 == "Pro") {
60         teamTwo = "Pro"
61     }
62
63     idDivSpeeches.innerHTML = ""
64     let everySpeech = "<br>" + "<br>" + teamOne + " Constructive: " + "<input type='
button' value='Start' id='idStartTimer' onclick='startStopwatch()'>";
65     everySpeech += "<input type='button' value='Stop' id='idStop Timer' onclick='
stopStopwatch()'>";
66     everySpeech += "<input type='button' value='Reset' id='idResetTimer' onclick='
resetStopwatch()'>";
67     everySpeech += "<input type='button' value='Save' id='idSaveTimer' onclick='
saveStopwatch()'>";
68
69     idDivSpeeches.innerHTML = everySpeech
70     idDivNext.innerHTML = "Press '1' to move on to the next speech"
71 }
72
73 // Description: Moves on to the next speech every time the key "1" is pressed
74 // Citation: None
75 // Input/Parameters: The key "1"
76 // Output/Return: The upcoming speech
77 function next(event) {
78     pressKey = event.key
79     let everySpeech = ""
80     idDivSpeeches.innerHTML = ""
81     idDivNext.innerHTML = "Press '1' to move on to the next speech"
82     let speechNames = [teamTwo + " Constructive: ", "1st Crossfire: ", teamOne + "
Rebuttal: ", teamTwo + " Rebuttal: ", "2nd Crossfire: ", teamOne + " Summary: ", teamTwo + "
Summary: ", "Grand Crossfire: ", teamOne + " Final Focus: ", teamTwo + " Final Focus: "]
83
84     while (pressKey == "1" && number <= 11) {
85         resetStopwatch();
86
87         everySpeech = "<br><br>" + speechNames[number - 1];
88         everySpeech += "<input type='button' value='Start' id='idStartTimer' onclick=
'startStopwatch()'>";
89         everySpeech += "<input type='button' value='Stop' id='idStopTimer' onclick=
'stopStopwatch()'>";
90         everySpeech += "<input type='button' value='Reset' id='idResetTimer' onclick=
'resetStopwatch()'>";
91         everySpeech += "<input type='button' value='Save' id='idSaveTimer' onclick=
'saveStopwatch()'>";
92
93         number = number + 1;
94         idDivSpeeches.innerHTML = everySpeech;
95         idDivSave.innerHTML = "";
96         pressKey = '';
97         if (number == 11) {
98             idDivNext.innerHTML = "Press 1 to see the final times"
99         }
100     }

```

```

101         if (number == 12) {
102             idDivNext.innerHTML = "Press 1 to see the final times"
103             idDivSpeeches.innerHTML = "<br>" + "<input type='button' value='Display
Times' id='idDisplay' onclick='displayTimes()'"
104             stopwatch.innerHTML = ""
105         }
106     }
107
108     // Description: Checks if the stopwatch isn't running then proceeds to start the
stopwatch
109     // Author: Educative.io
110     // Date Accessed: February 15th
111     // URL: https://www.educative.io/answers/how-to-create-a-stopwatch-in-javascript
112     function startStopwatch() {
113         if (!stopwatchInterval) {
114             startTime = new Date().getTime() - elapsedPausedTime; // get the starting
time by subtracting the elapsed paused time from the current time
115             stopwatchInterval = setInterval(updateStopwatch, 1000); // update every
second
116         }
117     }
118
119     // Description: Stops the stopwatch and calculates the elapsed pause time so that it
can start from the same spot
120     // Author: Educative.io
121     // Date Accessed: February 15th
122     // URL: https://www.educative.io/answers/how-to-create-a-stopwatch-in-javascript
123     function stopStopwatch() {
124         clearInterval(stopwatchInterval); // stop the interval
125         elapsedPausedTime = new Date().getTime() - startTime; // calculate elapsed paused
time
126         stopwatchInterval = null; // reset the interval variable
127     }
128
129     // Description: Stops the timer and resets the stopwatch through the innerHTML
130     // Author: Educative.io
131     // Date Accessed: February 15th
132     // URL: https://www.educative.io/answers/how-to-create-a-stopwatch-in-javascript
133     function resetStopwatch() {
134         stopStopwatch(); // stop the interval
135         elapsedPausedTime = 0; // reset the elapsed paused time variable
136         document.getElementById("stopwatch").innerHTML = "00:00:00"; // reset the display
137     }
138
139     // Description: Gets the current time and calculates the seconds, minutes, and hours
of the stopwatch then updates it on the display
140     // Author: Educative.io
141     // Date Accessed: February 15th
142     // URL: https://www.educative.io/answers/how-to-create-a-stopwatch-in-javascript
143     function updateStopwatch() {
144         let currentTime = new Date().getTime(); // get current time in milliseconds
145         let elapsedTime = currentTime - startTime; // calculate elapsed time in
milliseconds
146         let seconds = Math.floor(elapsedTime / 1000) % 60; // calculate seconds
147         let minutes = Math.floor(elapsedTime / 1000 / 60) % 60; // calculate minutes
148         let hours = Math.floor(elapsedTime / 1000 / 60 / 60); // calculate hours
149         let displayTime = pad(hours) + ":" + pad(minutes) + ":" + pad(seconds); // format
display time

```

```

150     stopwatch.innerHTML = displayTime; // update the display
151 }
152
153 // Description: Adds a zero in front of the number if it is less than ten
154 // Author: Educative.io
155 // Date Accessed: February 15th
156 // URL: https://www.educative.io/answers/how-to-create-a-stopwatch-in-javascript
157 function pad(num) {
158     // add a leading zero if the number is less than 10
159     return (num < 10 ? "0" : "") + num;
160 }
161
162 // Description: Saves the innerHTML of the stopwatch and displays a save message
163 // Citation: None
164 // Input/Parameters: Save button click
165 // Output/Return: Output message saying Your time has been saved
166 function saveStopwatch() {
167     speechList.push(stopwatch.innerHTML)
168     idDivSave.innerHTML = "Your time has been saved"
169     specificSpeech = specificSpeech + 1
170
171     if (specificSpeech == 11) {
172         formatArray()
173     }
174 }
175
176 // Description: Pushes the speech names to the list of times
177 // Citation: None
178 // Input/Parameters: None
179 // Output/Return: New list with the speech names and the times
180 function formatArray() {
181     let spliceIndex = 0
182     let speechNames = [teamOne + " Constructive: ", teamTwo + " Constructive: ", "1st
Crossfire: ", teamOne + " Rebuttal: ", teamTwo + " Rebuttal: ", "2nd Crossfire: ", teamOne +
" Summary: ", teamTwo + " Summary: ", "Grand Crossfire: ", teamOne + " Final Focus: ",
teamTwo + " Final Focus: "]
183
184     for (i = 0; i < speechNames.length; i++) {
185         speechList.splice(spliceIndex, 0, speechNames[i])
186         spliceIndex = spliceIndex + 2
187     }
188 }
189
190 // Description: Displays the speech times for the pro and con sides along with the
crossfire times
191 // Citation: None
192 // Input/Parameters: Display results button click
193 // Output/Return: All the speech times that were saved
194 function displayTimes() {
195     idDivNext.innerHTML = ""
196     displayOneId.innerHTML = "Team One: " + teamOne + " " + schoolOne.value + "<br>"
+ speechList[0] + speechList[1] + "<br>" + speechList[6] + speechList[7] + "<br>" +
speechList[12] + speechList[13] + "<br>" + speechList[18] + speechList[19]
197     displayTwoId.innerHTML = "Team Two: " + teamTwo + " " + schoolTwo.value + "<br>"
+ speechList[2] + speechList[3] + "<br>" + speechList[8] + speechList[9] + "<br>" +
speechList[14] + speechList[15] + "<br>" + speechList[20] + speechList[21]
198

```

```

199         crossId.innerHTML = "Cross Times:" + "<br>" + speechList[4] + speechList[5] + "<br>" + speechList[10] + speechList[11] + "<br>" + speechList[16] + speechList[17]
200     }
201
202     </script>
203 </head>
204
205 <body onkeyup="next(event)">
206     <h1>Public Forum Debate</h1>
207     <div id="idDivInstructions">This is a Public Forum Debate Stopwatch to assist you in your
208     debate rounds. First
209     Choose Sides and then the constructive speech timer will appear. Press '1' to move on
210     to the next speech and use
211     the start, stop, and reset buttons for the functionality of the stopwatch. Press the
212     save button after every
213     speech so that all
214     times are displayed at the end.</div>
215
216     <table>
217     <tr>
218         <th><span id="team1Id">Team 1</span></th>
219         <th><span id="team2Id">Team 2</span></th>
220     </tr>
221     <tr>
222         <th>Choose Sides: <select id="argueTeam1">
223             <option>Con</option>
224             <option>Pro</option>
225         </select>
226     </th>
227     <th>Choose Sides: <select id="argueTeam2">
228         <option>Pro</option>
229         <option>Con</option>
230     </select>
231     </th>
232     <tr>
233         <th>School/Team Name: <input type="text" id="schoolOne"></th>
234         <th>School/Team Name: <input type="text" id="schoolTwo"></th>
235     </tr>
236     </table>
237     <input type="button" value="Choose Sides" onclick="chooseSide()">
238     <div id="idDivSpeeches"></div>
239     <div class="customstopwatch" id="stopwatch">00:00:00</div>
240     <div id="idDivNext"></div>
241     <div id="idDivSave"></div>
242     <table>
243     <tr>
244         <th><span id="displayOneId"></span></th>
245         <th><span id="displayTwoId"></span></th>
246         <th><span id="crossId"></span></th>
247     </tr>
248     </table>
249
250 </body>
</html>

```