**Piano Keys**

You’re a web developer who has been hired by a music education company. This client wants you to create an interactive game to help their beginner-level piano students study. Create a piano player with DOM events in JavaScript!



If you get stuck during this project, check out the **project walkthrough video** which can be found in the “get help” section in the bottom-right of this window.



**Tasks**

**0/21Complete**

Mark the tasks as complete by checking them off

**Create An Interactive Piano Game**

**1.**

At the beginning of the code, we have variable name assignments for the keys array and the notes array. There is also a function looping through the elements of the keys array and pushing them to the notes array.

After the second comment, create a function named keyPlay that changes the background color of the keys when they are pressed down.

Be sure to use the .target property in the function because the target is being modified in this case.

Hint

1. You can declare the function however you’d like, but make sure it’s a named function because you’ll need it later.
2. When changing a hyphenated style property like background-color in JavaScript use camel case.

**2.**

Next, create a function named keyReturn that returns the background color of the keys to their default with an empty string '' when the mouse is released on the element. Be sure to use the .target property.

Hint

1. You can declare the function however you’d like, but make sure it’s a named function because you’ll need it later.
2. When changing a hyphenated style property in JavaScript use camel case.
3. Setting a style property equal to an empty string '' in JavaScript will return the element to its original style.

**3.**

Now that you have created two functions that change the color of the key elements, you must assign them as the values of event handler properties.

Create a named function and leave the code block blank for now. This function will be used later on to assign events to the keys, so it should take one parameter — you can call it note.

Hint

The note parameter is a placeholder for key elements that the function will be run on.

**4.**

Inside the function, create an event handler that runs the keyPlay as an event handler when a mousedown event fires on any note.

Hint

Use the event handler property syntax:

eventTarget.onevent = eventHandlerFunction;

**5.**

Inside the function, create a second event handler property that runs the keyReturn when a mouseup event fires on any note.

Hint

Use the event handler property syntax:

eventTarget.onevent = eventHandlerFunction;

**6.**

At the beginning of the code, we have variable name assignments for the keys array and the empty notes array. There is also a function looping through the keys array and pushing the keys elements to the notes array to be assigned a variable name.

Now, you must create a .forEach loop that will pass the elements in the notes array through your event assignment function.

Hint

The .forEach() method runs a function once “for each” element in an array. Check out the sytax:

arrayName.forEach(functionName);

**7.**

Now, the program knows what to do when each piano key has a mousedown or mouseup event fired on it. Run your code and see how it works!

**8.**

Next, there are variables that represent the progress buttons in the song box below that allow students to progress the piano.

The nextOne, nextTwo, nextThree, and startOver change the lyrics and musical notes of the song to help the student play along. In the beginning of the song the only button the student needs is nextOne. Because of this the .hidden properties of the other buttons are assigned the value of true.

Now you must create events on all the progress buttons. First, create an event handler property with a click event on the nextOne element.

Hint

Remember, the syntax for an event handler property is:

eventTarget.onevent

**9.**

To begin modifying the song box, you must switch the progress buttons first.

Using an anonymous event handler function, make the following changes to the button that appears after nextOne is clicked:

1. Reveal the nextTwo button by changing the .hidden property to make the nextTwo button appear.
2. Hide the nextOne button by changing the .hidden property to hide the nextOne button.

Hint

1. Remember the syntax for an anonymous event handler function:
2. eventTarget.onevent = function() {
3. ...

}

1. You can hide DOM elements in JavaScript with the .hidden property which takes Boolean values true or false.

**10.**

Next, a click event firing on the nextOne must change the music notes that guide the piano student through the song.

Add the following changes to the nextOne event handler function so the musical notes change when the button is clicked.

1. Change the content of the element with an ID of letter-note-five to D.
2. Change the content of the element with an ID of letter-note-six to C.

Hint

To change the content of an element in JavaScript use the .innerHTML property on the element.

**11.**

Create another event handler property with a click event on the button element called nextTwo. Then assign the property to the value of an anonymous event handler function.

Hint

Remember the syntax for an anonymous event handler function:

eventTarget.onevent = function() {

...

}

**12.**

Make the following changes to the button that appears when nextTwo is clicked:

1. Reveal the nextThree button by changing the .hidden property to make the nextThree button appear.
2. Hide the nextTwo button by changing the .hidden property to hide the nextTwo button.

Hint

You can hide DOM elements in JavaScript with the .hidden property which takes Boolean values true or false.

**13.**

Once the student has reached this point of the [Happy Birthday song](https://en.wikipedia.org/wiki/Happy_Birthday_to_You#%22Happy_birthday_to_you%22)the lyrics changes from HAP-PY BIRTH-DAY TO YOU to HAP-PY BIRTH-DAY DEAR FRI-END.

Make the make the following changes to the lyrics in the function when the button is clicked:

1. Change the content of the element with an ID of word-five to DEAR.
2. Change the content of the element with an ID of word-six to FRI-.

Stuck? Get a hint

**14.**

Now you have the lyrics HAP-PY BIRTH-DAY DEAR FRI-. To finish the line, you must add the -END to the song box under the piano.

The -END element is stored in the lastLyric variable.

Add a statement to the event handler function for nextTwo that changes the display property of lastLyric to 'inline-block'.

Hint

The syntax for changing the display property of an element in JavaScript is:

element.style.display = value;

**15.**

A click event firing on the second button must also change the music notes to guide the piano student through the song.

Add the following changes to the nextTwo event handler function so the musical notes change when the button is clicked:

1. Change the content of the element with an ID of letter-note-three to G.
2. Change the content of the element with an ID of letter-note-four to E.
3. Change the content of the element with an ID of letter-note-five to C.
4. Change the content of the element with an ID of letter-note-six to B.

Hint

1. To access elements by their IDs use

document.getElementByID('targetElement')

1. To change the content of an element in JavaScript use the .innerHTML property on the element.

**16.**

Create an event handler property with a click event on the nextThree element.

Hint

Remember the syntax for an event handler property:

eventTarget.onevent

**17.**

Using an anonymous event handler function, make the following changes to the button that appears when nextThree is clicked:

1. Reveal the startOver button by changing the .hidden property to make the startOver button appear.
2. Hide the nextThree button by changing the .hidden property to hide the nextThree button.

Hint

1. Remember the syntax for an anonymous event handler function:
2. eventTarget.onevent = function(){
3. ...

}

1. You can hide DOM elements in JavaScript with the .hidden property which takes Boolean values true or false.

**18.**

Add the following changes to the nextThree event handler function so the lyrics change when this button is clicked.

1. Change the content of the element with an ID of word-one to HAP-.
2. Change the content of the element with an ID of word-two to PY.
3. Change the content of the element with an ID of word-three to BIRTH.
4. Change the content of the element with an ID of word-four to DAY
5. Change the content of the element with an ID of word-five to TO.
6. Change the content of the element with an ID of word-six to YOU!.

Hint

1. To access elements by their IDs use

document.getElementByID('targetElement')

1. To change the content of an element in JavaScript use the .innerHTML property on the element.

**19.**

Add the following changes to the nextThree event handler function so the musical notes change when the button is clicked.

1. Change the content of the element with an ID of letter-note-one to F.
2. Change the content of the element with an ID of letter-note-two to F.
3. Change the content of the element with an ID of letter-note-three to E.
4. Change the content of the element with an ID of letter-note-four to C.
5. Change the content of the element with an ID of letter-note-five to D.
6. Change the content of the element with an ID of letter-note-six to C.

Hint

1. To access elements by their IDs use

document.getElementByID('targetElement')

1. To change the content of an element in JavaScript use the .innerHTML property on the element.

**20.**

Now you have the lyricsHAP-PY BIRTH-DAY TO YOU! -END — that couldn’t be right! To finish the line you must get rid of the “-end” in the song box.

Add a statement to the event handler function for nextThree that changes the display property of lastLyric back to 'none'.

Hint

The syntax for changing the display property of an element in JavaScript is:

element.style.display = value;

**21.**

Congrats, you’ve completed the Piano Player! Play around with the piano and the song box to fire all the events.