

# Tutorial

## a. Arrays

## b. Sound

### Array review and exercise

Review if needed:

1. [W3School's tutorial on arrays](http://www.w3schools.com/js/js_arrays.asp) ([http://www.w3schools.com/js/js\\_arrays.asp](http://www.w3schools.com/js/js_arrays.asp))
  - (only until "**Using the JavaScript Keyword new**")
2. [W3School's tutorial on While Loops](http://www.w3schools.com/js/js_loop_while.asp) ([http://www.w3schools.com/js/js\\_loop\\_while.asp](http://www.w3schools.com/js/js_loop_while.asp))
  - (only through the first two examples - up to (not including) do/while)

### Short exercises

1. [array 1](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays1) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_arrays1](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays1))
2. [array 2](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays2) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_arrays2](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays2))
3. [array 3](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays3) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_arrays3](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays3))
4. [array 4](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays4) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_arrays4](http://www.w3schools.com/js/exercise.asp?filename=exercise_arrays4))
5. [while 0](http://www.w3schools.com/js/tryit.asp?filename=tryjs_loop_while_cars) ([http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_loop\\_while\\_cars](http://www.w3schools.com/js/tryit.asp?filename=tryjs_loop_while_cars))
6. [while 1](http://www.w3schools.com/js/exercise.asp?filename=exercise_while1) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_while1](http://www.w3schools.com/js/exercise.asp?filename=exercise_while1))
7. [while 2](http://www.w3schools.com/js/exercise.asp?filename=exercise_while2) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_while2](http://www.w3schools.com/js/exercise.asp?filename=exercise_while2))
8. [while 3](http://www.w3schools.com/js/exercise.asp?filename=exercise_while3) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_while3](http://www.w3schools.com/js/exercise.asp?filename=exercise_while3))
9. [while 4](http://www.w3schools.com/js/exercise.asp?filename=exercise_while4) ([http://www.w3schools.com/js/exercise.asp?filename=exercise\\_while4](http://www.w3schools.com/js/exercise.asp?filename=exercise_while4))

## Challenges

Next we'll add sound to enrich an application. If you prefer, you can use a copy of your H06 (the bouncing ball) as a "starting template" instead of the one provided here - it is very similar to the template, so will work for the sound challenges below.

**For these challenges**, please open an account for yourself on [freesound.org](http://freesound.org). **Explore** the hundreds of thousands of sounds they have there (contributed by users), and find and download some good sounds you can:

- Give users audio controls to play,
- use for a background sound,
- Sound for when ball bumps into a wall.

(There are some sounds in the 'resources' folder already you can use, but have some fun exploring [freesound.org](http://freesound.org) to find ones you like for any application!)

Next, we will explore the HTML 5 <audio> element.

NOTE: Browser policy prevents sounds from playing before some kind of user interaction on a page – in particular, you can't play a sound as the page loads!

### Starting from the template (or your own completed H06) code:

1. Download your sounds from freesound.org (or from wherever you like!) into a subdirectory of today's project directory named 'resources'.
2. Create an HTML <audio> element with 'controls' inside the <aside> element on your main html page. Have it load the background sound which you stored in your resources directory. Run your browser page to see that it is working.
  - Hint: Look here ([https://www.w3schools.com/html/html5\\_audio.asp](https://www.w3schools.com/html/html5_audio.asp)) to see how to use the audio element.
  - Note: You probably don't actually need to include the type attribute on the source element in Chrome, but it is a good idea.
  - Different audio file formats: such as .mp3, .ogg, .wav files all work in Chrome. .aif files do not.
  - Note: You could have also specified a full URL if you have one for a sound on the web and don't want to store it locally.
3. Next let's start/stop the animation with the toggle button.
  - Create a <input> element of type 'button' on the 'nav' side panel. Set its text to be 'start' when the program loads, and then toggle between 'start' and 'stop' each time it is pushed.
  - Use it to start/ stop the the animation by starting and stopping the interval timer callbacks to your draw function.
4. Next, we will add a sound to the toggle button that plays whenever the animation starts.
  - Create an HTML 5 audio element directly in our JavaScript code, then play the sound whenever the animation is started with the toggle button:
    - `let variable_name = new Audio(audio_uri); // where audio_uri is either a path name or a web address`
  - Use it to toggle between playing and pausing the sound. Note: audio elements have a .play() and a .pause() method, a .loop property that takes a boolean (true or false) value, and a .volume attribute taking values in [0,1]. Try looping it just to see how that works.
  - Note: you can't play **any** sound in most browsers without first having some sort of user interaction. In particular, you can't have a background sound play "automatically" when the page loads. That is one reason we are using a button.
5. Now for the bounce: play another audio element whenever the ball hits a wall.
  - Does it work when the ball hits two walls one right after the other? **Why not?**
  - Fix this! Audio elements have a .currentTime property that you can set to 0 after pausing in order to start the sound from the beginning again.

6. Get a total of 4 different sounds to use for bouncing, and put them in an array.  
Play a different one of the sounds for each wall

