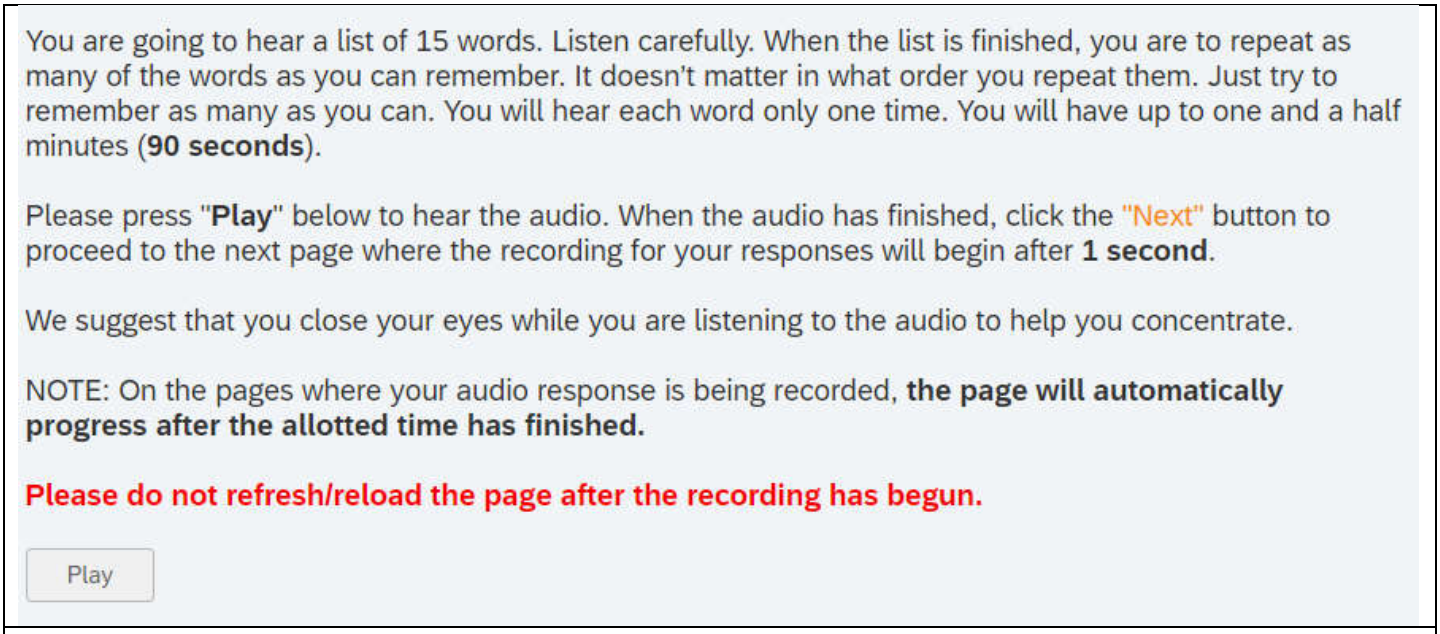
Here we provide the JavaScript codes developed by our team to perform the BTACT (Brief Test of Adult Cognition by Telephone) via Qualtrics Survey (instead of using telephone).




```
Qualtrics.SurveyEngine.addOnload(function()
{
	var play = document.querySelector('.play');
//var pause = document.querySelector('.pause');
var audio = document.getElementById('player');
var LOG = true; // logging to debug
var AutoplayEnabled = false;
var text = "The audio has ended."

function play_audio() {
	LOG && console.log("Audio started playing")
	audio.play();
}

play.onclick =  function(){
	play_audio();
}

/*
pause.onclick = function(){
   LOG && console.log("Paused");
   audio.pause();
}
*/

audio.onended = function() {
  LOG && console.log("Audio is completed")
  play.disabled = true;
  //pause.disabled = true;
  document.getElementById("text").innerHTML = text;
}
```

```
if(AutoplayEnabled){
  setTimeout(function(){ // Autoplay after 2 seconds after page load
     LOG && console.log("AutoPlay Started");
     play_audio();
  }, 2000);
}

});
```

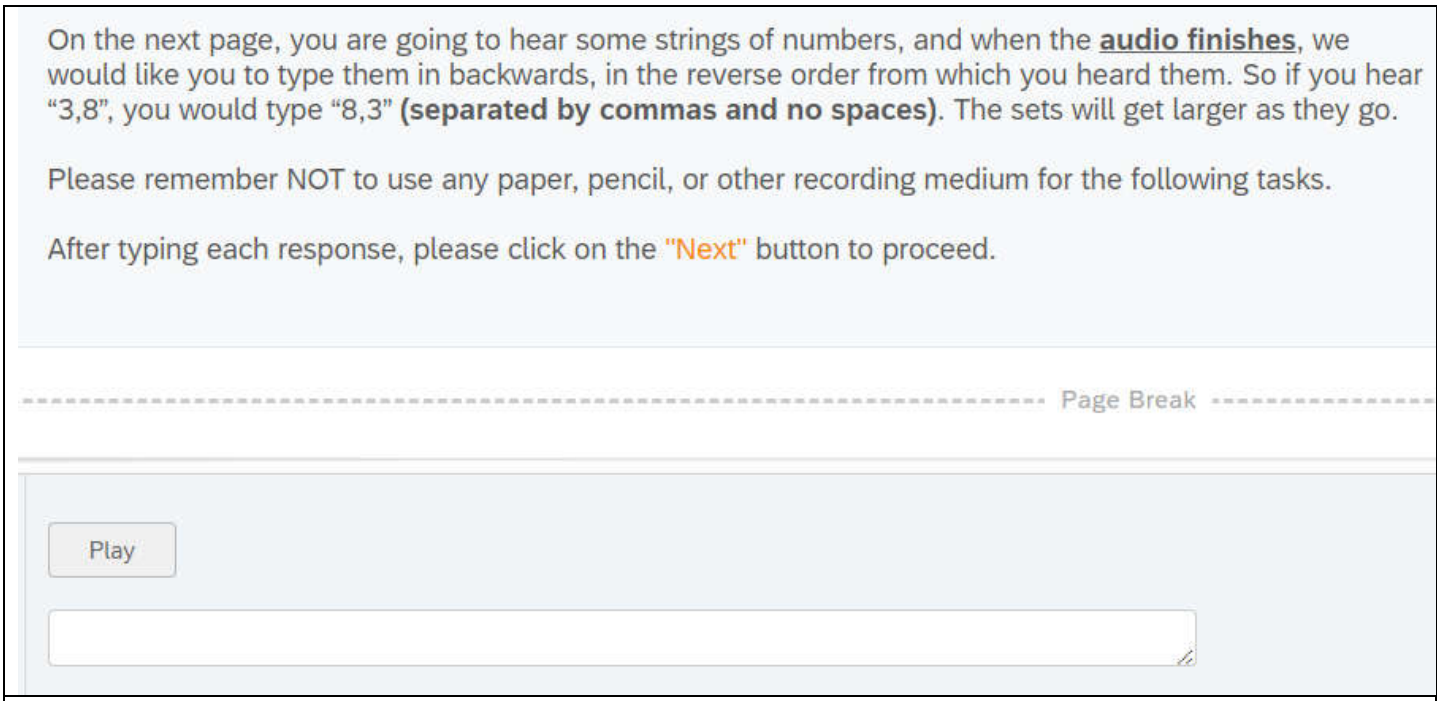
BEGIN.

```
Qualtrics.SurveyEngine.addOnload(function()
{
        var recordingStarted = "<b><font color='red'>Recording started.</font></b>"
   var recordingFinished = "Recording has ended. " + '<p style="font-family:arial;font-size:1em;">Please click the <span style="color: rgb(255, 140, 0);">"Next"</span> button to proceed.</p>'
        var midRecordingMsg = "<b><font color='red'>Please allow for full 90 seconds to elapse, after which you can proceed..</font></b>";
        var that = this;

        this.hideNextButton();

        /* getUserMedia is forked for multiple browser versions*/

navigator.getUserMedia = ( navigator.getUserMedia ||
                navigator.webkitGetUserMedia ||
                navigator.mozGetUserMedia ||
                navigator.msGetUserMedia);

//var stop = document.querySelector('#stopButton');
//var stop = document.getElementById("stopButton");
var isRecordingCompleted = false;

// disable stop button while not recording
//stop.disabled = true;

//main block for doing the audio recording

if (navigator.getUserMedia) {
  console.log('getUserMedia supported.');

  var constraints = { audio: true };
  var chunks = [];

  var onSuccess = function(stream) {
  var mediaRecorder = new MediaRecorder(stream);
```

```
 function record() {
    mediaRecorder.start(90000);
    console.log(mediaRecorder.state);
            document.getElementById("text").innerHTML = recordingStarted;
    console.log("recorder started");

            var aud =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_0p29hPTcJAU9DJb');
                    aud.volume = 0.5;
                    aud.play();

            (function()
             {
            document.getElementById("midMsg").innerHTML = midRecordingMsg;
            }).delay(45);

            (function()
                    {
//         that.showNextButton();
                                that.clickNextButton();
                    }
           ).delay(95);

            //     stop.disabled = false;
   }

                    setTimeout(function(){
       record();
      }, 1000);

/*   stop.onclick = function() {
    isRecordingCompleted = true;
            document.getElementById("text").innerHTML = recordingFinished;
    mediaRecorder.stop();
    console.log(mediaRecorder.state);
    console.log("recorder stopped");
    stop.style.background = "";

    stop.disabled = true;
   }
*/
   mediaRecorder.onstop = function(e) {
    console.log("MediaRecorder.stop() called.");

                    var aud1 =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_5olUCq0oC8qHO4d');
                    aud1.volume = 0.5;
                    aud1.play();

    var blob = new Blob(chunks, { 'type' : 'audio/ogg; codecs=opus' });
    var audioURL = window.URL.createObjectURL(blob);

    //convert current time as a string to use to rename the blob
    var current_date = new Date();
    var dd = current_date.getDate();
```

```
var mm = current_date.getMonth()+1;
var yyyy = current_date.getFullYear();
var hh = current_date.getHours()
var mi = current_date.getMinutes()
var ss = current_date.getSeconds()

if(dd<10)
  dd='0'+dd
if(mm<10)
  mm='0'+mm
if(hh<10)
  hh='0'+hh
if(mi<10)
  mi='0'+mi
if(ss<10)
  ss='0'+ss

current_date = mm+'_'+dd+'_'+yyyy+'_'+hh+mi+ss;

//Save the blob file to Dropbox
var xhr = new XMLHttpRequest();

xhr.onload = function() {
   /* if (xhr.status === 200) {
       var fileInfo = JSON.parse(xhr.response);
       // Upload succeeded. Do something here with the file info.
    }
    else {
       var errorMessage = xhr.response || 'Unable to upload file';
       // Upload failed. Do something here with the error.
    }*/
};

       /*
[redacted]
}));*/
       [redacted]
         var encdata = window.btoa("[redacted]");
xhr.open('POST', [redacted]);
xhr.setRequestHeader('[redacted]');
         xhr.setRequestHeader('[redacted]');
         xhr.setRequestHeader('Filename', 'Word_List_Recall');
xhr.send(blob);
console.log("file sent");
```

```
   }

   mediaRecorder.ondataavailable = function(e) {
     if (!isRecordingCompleted){
       isRecordingCompleted = true;
       console.log("timeslice end")
       console.log("blob size "+e.data.size)
       chunks.push(e.data);
       console.log("wav data saved");
                       mediaRecorder.stop();
//       stop.click();
     }
     console.log("blob size "+e.data.size)
     chunks.push(e.data);
     console.log("wav data saved")
   }
  }

  var onError = function(err) {
   console.log('The following error occured: ' + err);
  }

  navigator.getUserMedia(constraints, onSuccess, onError);
} else {
   console.log('getUserMedia not supported on your browser!');
}


});
```




```
Qualtrics.SurveyEngine.addOnload(function()
{
        var play = document.querySelector('.play');
```

```
//var pause = document.querySelector('.pause');
var audio = document.getElementById('player');
var LOG = true; // logging to debug
var AutoplayEnabled = false;
var text = "The audio has ended."

function play_audio() {
        LOG && console.log("Audio started playing")
        audio.play();
}

play.onclick =  function(){
        play_audio();
}

/*
pause.onclick = function(){
   LOG && console.log("Paused");
   audio.pause();
}
*/

audio.onended = function() {
  LOG && console.log("Audio is completed")
  play.disabled = true;
  //pause.disabled = true;
  document.getElementById("text").innerHTML = text;
}

if(AutoplayEnabled){
  setTimeout(function(){ // Autoplay after 2 seconds after page load
     LOG && console.log("AutoPlay Started");
     play_audio();
  }, 2000);
}

});
```

Q181

Now you are going to be given a category and you will name things that belong in that category. For example, if the category was "**fruit**," you could say things like "**peach**" or "**pear**."

In a moment you will be given a *different* category. When you see "Begin," you will name all the things from this new category you can think of, as fast as you can. You will have one minute (**60 seconds**) to do this.

When you proceed to the next page the recording will begin after **3 seconds.**

Page Break

Q182

The new category is **Animals**... BEGIN.

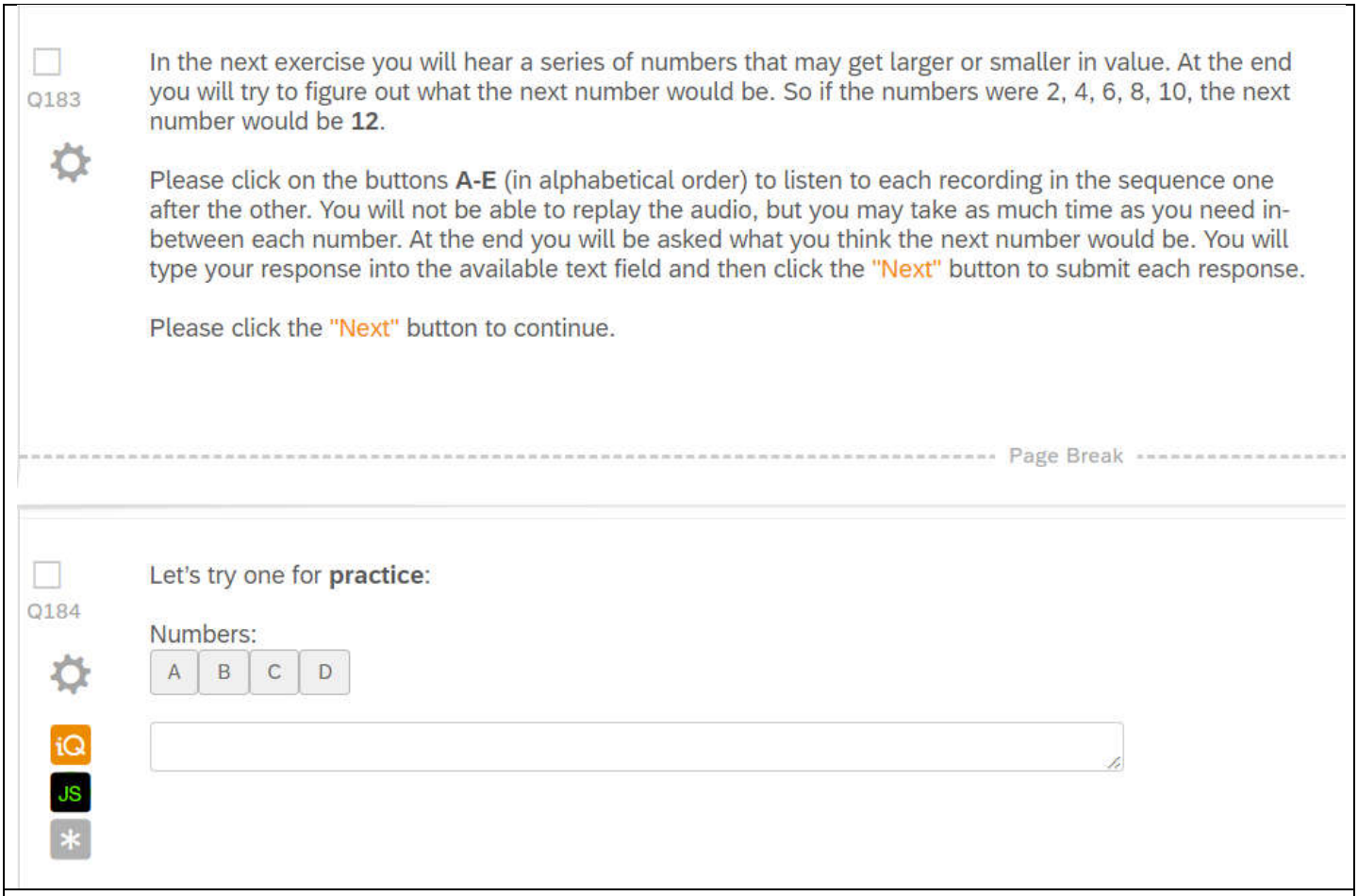
JS

```
Qualtrics.SurveyEngine.addOnload(function()
{
        var recordingStarted = "<b><font color='red'>The recording has started. </font></b>"
   var recordingFinished = "Recording has ended"

        var that = this;
        this.hideNextButton();
//      this.showNextButton().delay(63);

        /* getUserMedia is forked for multiple browser versions*/

navigator.getUserMedia = ( navigator.getUserMedia ||
                navigator.webkitGetUserMedia ||
                navigator.mozGetUserMedia ||
                navigator.msGetUserMedia);

//var stop = document.querySelector('.stop');
var isRecordingCompleted = false;

// disable stop button while not recording
//stop.disabled = true;

//main block for doing the audio recording

if (navigator.getUserMedia) {
```

```
  console.log('getUserMedia supported.');

  var constraints = { audio: true };
  var chunks = [];

  var onSuccess = function(stream) {
    var mediaRecorder = new MediaRecorder(stream);

  function record() {
     mediaRecorder.start(60000);
     console.log(mediaRecorder.state);
           document.getElementById("text").innerHTML = recordingStarted;
     console.log("recorder started");

           var aud =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_0p29hPTcJAU9DJb');
                 aud.volume = 0.5;
                 aud.play();

//               setTimeout(function(){
//                     mediaRecorder.stop();
//       }, 59000);

           (function()
                 {
                       that.clickNextButton();
                 }
           ).delay(65);

   //   stop.disabled = false;
    }

                 setTimeout(function(){
       record();
      }, 1000);

/*    stop.onclick = function() {
     isRecordingCompleted = true;
           document.getElementById("text").innerHTML = recordingFinished;
     mediaRecorder.stop();
     console.log(mediaRecorder.state);
     console.log("recorder stopped");
     stop.style.background = "";

     stop.disabled = true;
   }
*/
    mediaRecorder.onstop = function(e) {
     console.log("MediaRecorder.stop() called.");

                 var aud1 =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_5olUCq0oC8qHO4d');
                 aud1.volume = 0.5;
                 aud1.play();
```

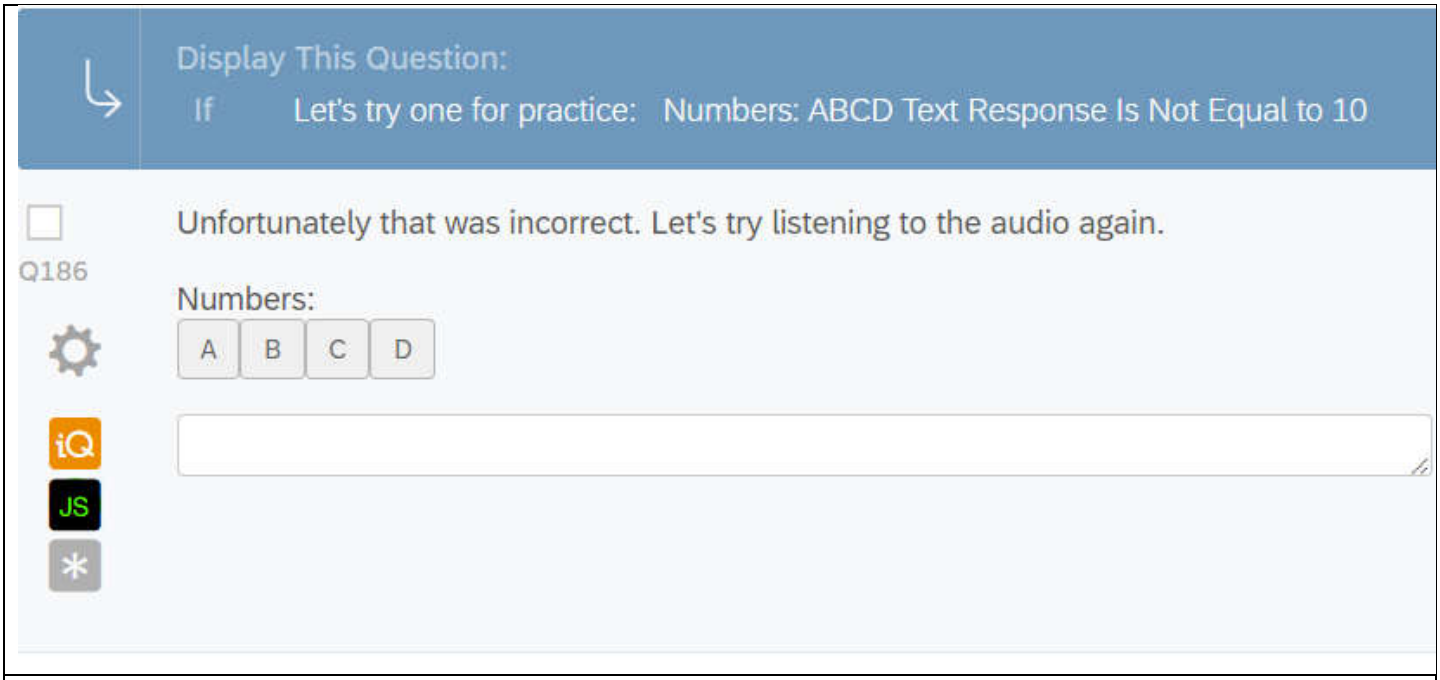
```
    var blob = new Blob(chunks, { 'type' : 'audio/ogg; codecs=opus' });
    var audioURL = window.URL.createObjectURL(blob);

    //convert current time as a string to use to rename the blob
    var current_date = new Date();
    var dd = current_date.getDate();
    var mm = current_date.getMonth()+1;
    var yyyy = current_date.getFullYear();
    var hh = current_date.getHours()
    var mi = current_date.getMinutes()
    var ss = current_date.getSeconds()

    if(dd<10)
      dd='0'+dd
    if(mm<10)
      mm='0'+mm
    if(hh<10)
      hh='0'+hh
    if(mi<10)
      mi='0'+mi
    if(ss<10)
      ss='0'+ss

    current_date = mm+'_'+dd+'_'+yyyy+'_'+hh+mi+ss;

    //Save the blob file to Dropbox
    var xhr = new XMLHttpRequest();

    xhr.onload = function() {/*
       if (xhr.status === 200) {
          var fileInfo = JSON.parse(xhr.response);
          // Upload succeeded. Do something here with the file info.
       }
       else {
          var errorMessage = xhr.response || 'Unable to upload file';
          // Upload failed. Do something here with the error.
       }*/
    };
/*
[redacted]
*/
[redacted]
          var encdata = window.btoa([redacted];
    xhr.open('POST',[redacted]);
          xhr.setRequestHeader([redacted]');
```

```
            xhr.setRequestHeader([redacted]);
    xhr.setRequestHeader([redacted]);
    xhr.send(blob);
    console.log("file sent")

   }

   mediaRecorder.ondataavailable = function(e) {
    if (!isRecordingCompleted){
      isRecordingCompleted = true;
      console.log("timeslice end")
      console.log("blob size "+e.data.size)
      chunks.push(e.data);
      console.log("wav data saved")
      //stop.click();
                    mediaRecorder.stop();
    isRecordingCompleted = true;
            document.getElementById("text").innerHTML = recordingFinished;
    }
    console.log("blob size "+e.data.size)
    chunks.push(e.data);
    console.log("wav data saved")
   }
  }

  var onError = function(err) {
   console.log('The following error occured: ' + err);
  }

  navigator.getUserMedia(constraints, onSuccess, onError);
} else {
   console.log('getUserMedia not supported on your browser!');
}


});
```



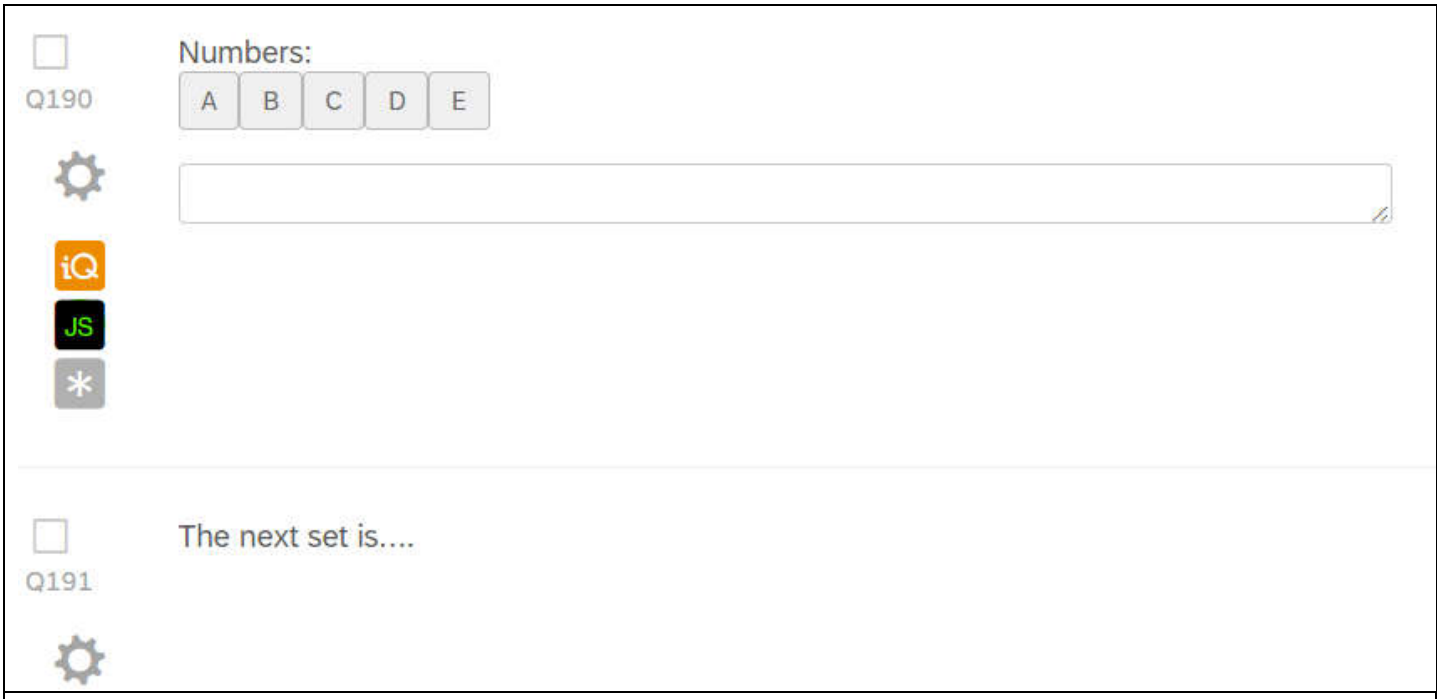

```
Qualtrics.SurveyEngine.addOnload(function()
{
var textbox = document.querySelector('.ChoiceStructure');
textbox.style.visibility = 'hidden';
var audio_button1 = document.querySelector('.a1');
var audio_button2 = document.querySelector('.a2');
var audio_button3 = document.querySelector('.a3');
var audio_button4 = document.querySelector('.a4');
var audio1 = document.getElementById('audio1');
var audio2 = document.getElementById('audio2');
var audio3 = document.getElementById('audio3');
var audio4 = document.getElementById('audio4');
var LOG = true; // logging to debug

audio_button1.onclick = function(){
  LOG && console.log("Audio1 started playing")
  audio1.play();
  audio_button1.disabled = true;
}
audio_button2.onclick = function(){
  LOG && console.log("Audio2 started playing")
  audio2.play();
  audio_button2.disabled = true;
}
audio_button3.onclick = function(){
  LOG && console.log("Audio3 started playing")
  audio3.play();
```

```
  audio_button3.disabled = true;
}
audio_button4.onclick = function(){
  LOG && console.log("Audio4 started playing")
  audio4.play();
  audio_button4.disabled = true;
}
audio4.onended = function() {
 textbox.style.visibility = 'visible';
}
});
```



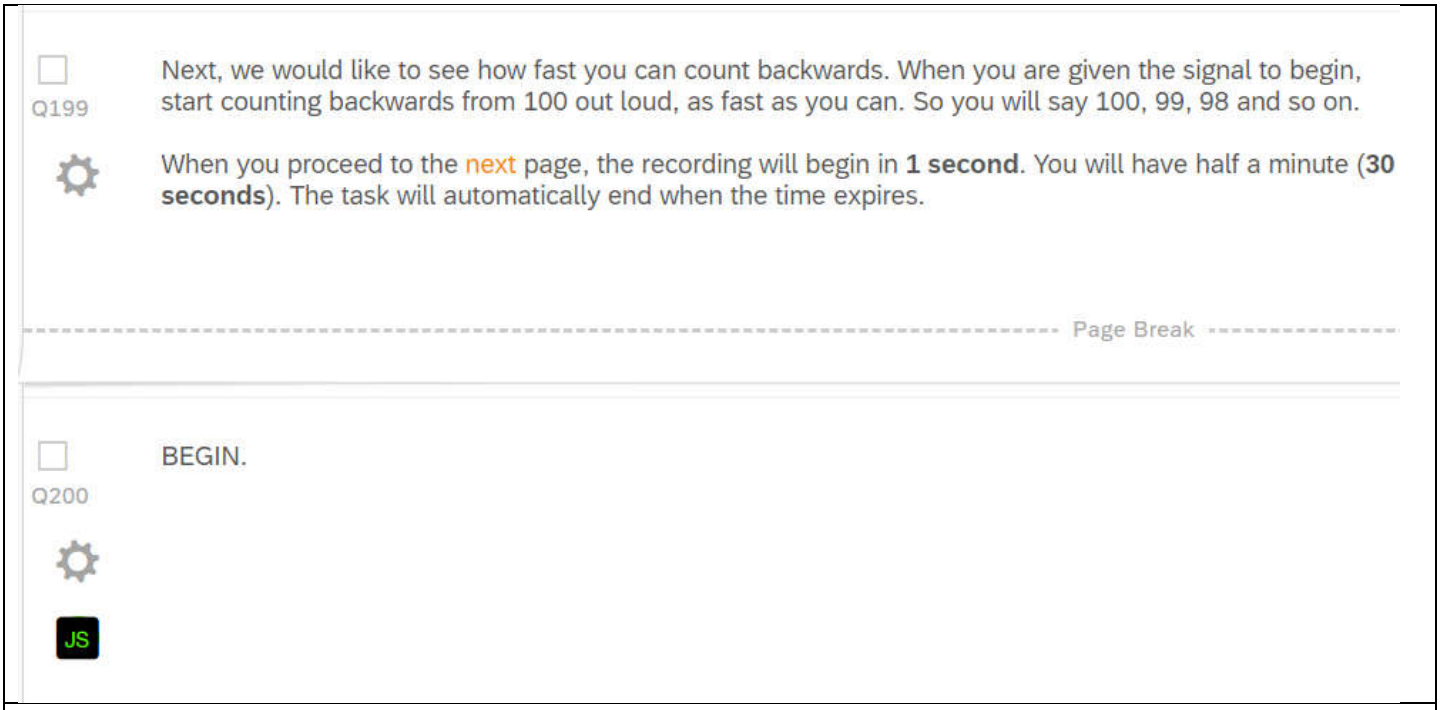

```
Qualtrics.SurveyEngine.addOnload(function()
{
var textbox = document.querySelector('.ChoiceStructure');
textbox.style.visibility = 'hidden';
var audio_button1 = document.querySelector('.a1');
var audio_button2 = document.querySelector('.a2');
var audio_button3 = document.querySelector('.a3');
var audio_button4 = document.querySelector('.a4');
var audio1 = document.getElementById('audio1');
var audio2 = document.getElementById('audio2');
var audio3 = document.getElementById('audio3');
var audio4 = document.getElementById('audio4');
var LOG = true; // logging to debug

audio_button1.onclick = function(){
  LOG && console.log("Audio1 started playing")
  audio1.play();
  audio_button1.disabled = true;
}
audio_button2.onclick = function(){
  LOG && console.log("Audio2 started playing")
```

```
  audio2.play();
  audio_button2.disabled = true;
}
audio_button3.onclick = function(){
  LOG && console.log("Audio3 started playing")
  audio3.play();
  audio_button3.disabled = true;
}
audio_button4.onclick = function(){
  LOG && console.log("Audio4 started playing")
  audio4.play();
  audio_button4.disabled = true;
}
audio4.onended = function() {
 textbox.style.visibility = 'visible';
}

});
```




```
Qualtrics.SurveyEngine.addOnload(function()
{
var textbox = document.querySelector('.ChoiceStructure');
textbox.style.visibility = 'hidden';
var audio_button1 = document.querySelector('.a1');
var audio_button2 = document.querySelector('.a2');
var audio_button3 = document.querySelector('.a3');
var audio_button4 = document.querySelector('.a4');
var audio_button5 = document.querySelector('.a5');
var audio1 = document.getElementById('audio1');
var audio2 = document.getElementById('audio2');
var audio3 = document.getElementById('audio3');
var audio4 = document.getElementById('audio4');
var audio5 = document.getElementById('audio5');
```

```
var LOG = true; // logging to debug

audio_button1.onclick = function(){
  LOG && console.log("Audio1 started playing")
  audio1.play();
  audio_button1.disabled = true;
}
audio_button2.onclick = function(){
  LOG && console.log("Audio2 started playing")
  audio2.play();
  audio_button2.disabled = true;
}
audio_button3.onclick = function(){
  LOG && console.log("Audio3 started playing")
  audio3.play();
  audio_button3.disabled = true;
}
audio_button4.onclick = function(){
  LOG && console.log("Audio4 started playing")
  audio4.play();
  audio_button4.disabled = true;
}
audio_button5.onclick = function(){
  LOG && console.log("Audio5 started playing")
  audio5.play();
  audio_button5.disabled = true;
}
audio5.onended = function() {
 textbox.style.visibility = 'visible';
}
});
```



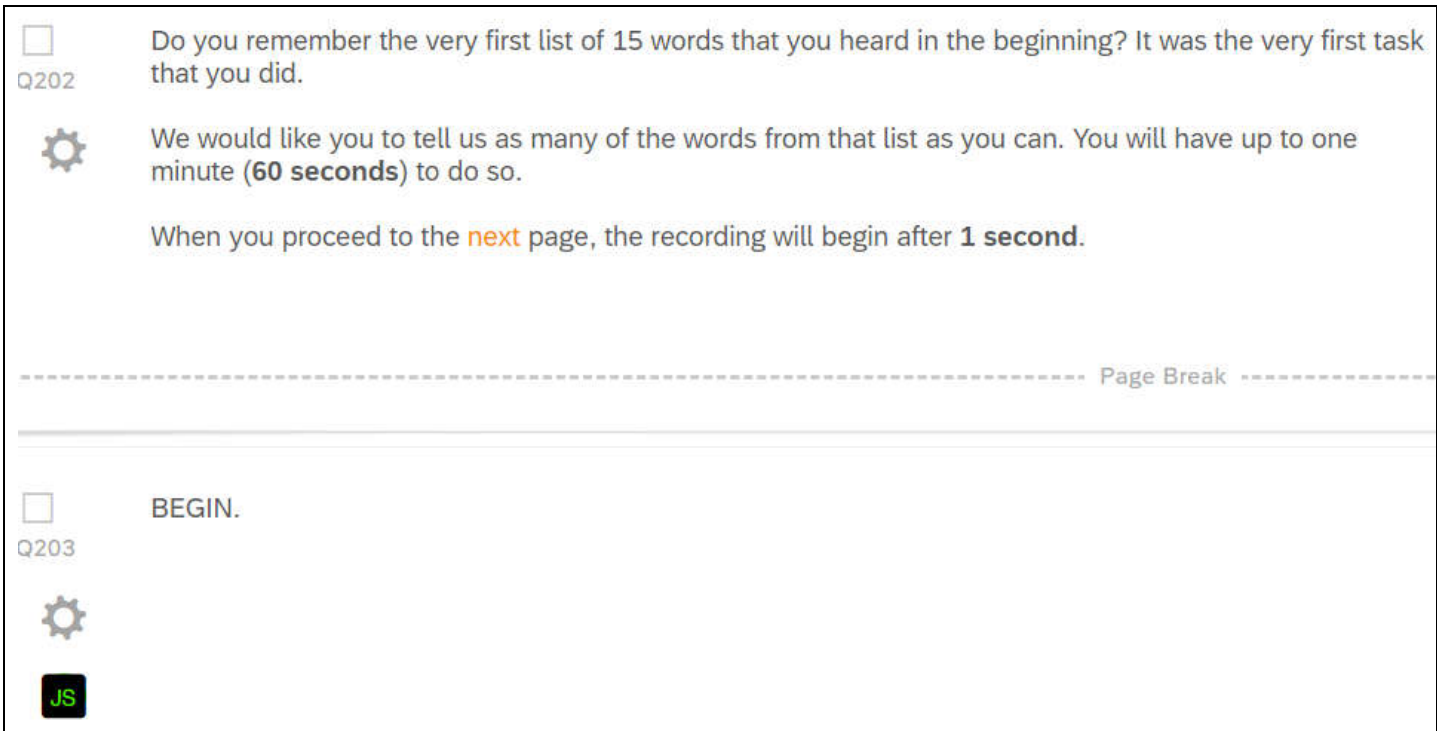

```
Qualtrics.SurveyEngine.addOnload(function()
```

```
{
        var recordingStarted = "<b><font color='red'>The recording has started</font></b>";
   var recordingFinished = "Recording has ended";
        var that = this;
        this.hideNextButton();
//    this.showNextButton.delay(33);


        /* getUserMedia is forked for multiple browser versions*/

navigator.getUserMedia = ( navigator.getUserMedia ||
                navigator.webkitGetUserMedia ||
                navigator.mozGetUserMedia ||
                navigator.msGetUserMedia);

//var stop = document.querySelector('.stop');
var isRecordingCompleted = false;

// disable stop button while not recording
//stop.disabled = true;

//main block for doing the audio recording

if (navigator.getUserMedia) {
 console.log('getUserMedia supported.');

 var constraints = { audio: true };
 var chunks = [];

 var onSuccess = function(stream) {
  var mediaRecorder = new MediaRecorder(stream);

 function record() {
    mediaRecorder.start(30000);
    console.log(mediaRecorder.state);
          document.getElementById("text").innerHTML = recordingStarted;
    console.log("recorder started");

          var aud =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_0p29hPTcJAU9DJb');
                aud.volume = 0.5;
                aud.play();

//          setTimeout(function(){
//                        mediaRecorder.stop();
//      }, 29000);

                (function()
                {
                        that.clickNextButton();
                }
                ).delay(33);

  //   stop.disabled = false;
```

```
  }

                    setTimeout(function(){
     record();
     }, 1000);

/*   stop.onclick = function() {
    isRecordingCompleted = true;
          document.getElementById("text").innerHTML = recordingFinished;
    mediaRecorder.stop();
    console.log(mediaRecorder.state);
    console.log("recorder stopped");
    stop.style.background = "";

    stop.disabled = true;
  }
*/
  mediaRecorder.onstop = function(e) {
    console.log("MediaRecorder.stop() called.");

                    var aud1 =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_5olUCq0oC8qHO4d');
                    aud1.volume = 0.5;
                    aud1.play();

    var blob = new Blob(chunks, { 'type' : 'audio/ogg; codecs=opus' });
    var audioURL = window.URL.createObjectURL(blob);

    //convert current time as a string to use to rename the blob
    var current_date = new Date();
    var dd = current_date.getDate();
    var mm = current_date.getMonth()+1;
    var yyyy = current_date.getFullYear();
    var hh = current_date.getHours()
    var mi = current_date.getMinutes()
    var ss = current_date.getSeconds()

    if(dd<10)
     dd='0'+dd
    if(mm<10)
     mm='0'+mm
    if(hh<10)
     hh='0'+hh
    if(mi<10)
     mi='0'+mi
    if(ss<10)
     ss='0'+ss

    current_date = mm+'_'+dd+'_'+yyyy+'_'+hh+mi+ss;

    //Save the blob file to Dropbox
    var xhr = new XMLHttpRequest();

    xhr.onload = function() {/*
```

```
        if (xhr.status === 200) {
          var fileInfo = JSON.parse(xhr.response);
          // Upload succeeded. Do something here with the file info.
        }
        else {
          var errorMessage = xhr.response || 'Unable to upload file';
          // Upload failed. Do something here with the error.
        }*/
    };
/*
[redacted]
*/
    [redacted]
            var encdata = window.btoa([redacted]);
    xhr.open('POST', [redacted]);
            xhr.setRequestHeader([redacted]);
            xhr.setRequestHeader([redacted]);
    xhr.setRequestHeader([redacted];
    xhr.send(blob);
    console.log("file sent")

   }

   mediaRecorder.ondataavailable = function(e) {
    if (!isRecordingCompleted){
      isRecordingCompleted = true;
      console.log("timeslice end")
      console.log("blob size "+e.data.size)
      chunks.push(e.data);
      console.log("wav data saved")
      //stop.click();
                    mediaRecorder.stop();
    isRecordingCompleted = true;
            document.getElementById("text").innerHTML = recordingFinished;
    }
    console.log("blob size "+e.data.size)
    chunks.push(e.data);
    console.log("wav data saved")
   }
  }

  var onError = function(err) {
   console.log('The following error occured: ' + err);
  }
```

```
  navigator.getUserMedia(constraints, onSuccess, onError);
} else {
   console.log('getUserMedia not supported on your browser!');
}


});
```




```
Qualtrics.SurveyEngine.addOnload(function()
{
	var recordingStarted = "<b><font color='red'>The recording has started</font></b>.";
   var recordingFinished = "Recording has ended";
	var midMsg = "<b><font color='red'>Please allow for the full 60 seconds to elapse and you will be able to proceed</font></b>";
	var that = this;
	this.hideNextButton();
  // this.showNextButton.delay(63);
	/* getUserMedia is forked for multiple browser versions*/

navigator.getUserMedia = ( navigator.getUserMedia ||
		navigator.webkitGetUserMedia ||
		navigator.mozGetUserMedia ||
		navigator.msGetUserMedia);

//var stop = document.querySelector('.stop');
var isRecordingCompleted = false;

// disable stop button while not recording
//stop.disabled = true;

//main block for doing the audio recording
```

```
if (navigator.getUserMedia) {
  console.log('getUserMedia supported.');

  var constraints = { audio: true };
  var chunks = [];

  var onSuccess = function(stream) {
    var mediaRecorder = new MediaRecorder(stream);

  function record() {
     mediaRecorder.start(60000);
     console.log(mediaRecorder.state);
            document.getElementById("text").innerHTML = recordingStarted;
     console.log("recorder started");

            var aud =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_0p29hPTcJAU9DJb');
                    aud.volume = 0.5;
                    aud.play();

//                  setTimeout(function(){
//                          mediaRecorder.stop();
//      }, 59000);

            (function()
                    {
                     document.getElementById("midRecordingMsg").innerHTML = midMsg;
                    }
           ).delay(30);

            (function()
                    {
                            that.clickNextButton();
                    }
           ).delay(65);

   //   stop.disabled = false;
    }

                    setTimeout(function(){
       record();
      }, 1000);

/*    stop.onclick = function() {
     isRecordingCompleted = true;
            document.getElementById("text").innerHTML = recordingFinished;
     mediaRecorder.stop();
     console.log(mediaRecorder.state);
     console.log("recorder stopped");
     stop.style.background = "";

     stop.disabled = true;
    }
```

```
*/
   mediaRecorder.onstop = function(e) {
    console.log("MediaRecorder.stop() called.");

                    var aud1 =  new Audio('https://ufl.qualtrics.com/CP/File.php?F=F_5olUCq0oC8qHO4d');
                    aud1.volume = 0.5;
                    aud1.play();

    var blob = new Blob(chunks, { 'type' : 'audio/ogg; codecs=opus' });
    var audioURL = window.URL.createObjectURL(blob);

    //convert current time as a string to use to rename the blob
    var current_date = new Date();
    var dd = current_date.getDate();
    var mm = current_date.getMonth()+1;
    var yyyy = current_date.getFullYear();
    var hh = current_date.getHours()
    var mi = current_date.getMinutes()
    var ss = current_date.getSeconds()

    if(dd<10)
      dd='0'+dd
    if(mm<10)
      mm='0'+mm
    if(hh<10)
      hh='0'+hh
    if(mi<10)
      mi='0'+mi
    if(ss<10)
      ss='0'+ss

    current_date = mm+'_'+dd+'_'+yyyy+'_'+hh+mi+ss;

    //Save the blob file to Dropbox
    var xhr = new XMLHttpRequest();

    xhr.onload = function() {/*
       if (xhr.status === 200) {
          var fileInfo = JSON.parse(xhr.response);
          // Upload succeeded. Do something here with the file info.
       }
       else {
          var errorMessage = xhr.response || 'Unable to upload file';
          // Upload failed. Do something here with the error.
       }*/
    };
/*
```

```
    [redacted]
    }));
*/
    [redacted]
            var encdata = window.btoa([redacted]);
    xhr.open('POST', [redacted]);
            xhr.setRequestHeader([redacted]);
            xhr.setRequestHeader([redacted]);
    xhr.setRequestHeader([redacted]);
    xhr.send(blob);
    console.log("file sent")

   }

   mediaRecorder.ondataavailable = function(e) {
    if (!isRecordingCompleted){
     isRecordingCompleted = true;
     console.log("timeslice end")
     console.log("blob size "+e.data.size)
     chunks.push(e.data);
     console.log("wav data saved")
     //stop.click();
                    mediaRecorder.stop();
    isRecordingCompleted = true;
            document.getElementById("text").innerHTML = recordingFinished;
    }
    console.log("blob size "+e.data.size)
    chunks.push(e.data);
    console.log("wav data saved")
   }
  }

  var onError = function(err) {
   console.log('The following error occured: ' + err);
  }

  navigator.getUserMedia(constraints, onSuccess, onError);
} else {
   console.log('getUserMedia not supported on your browser!');
}


});
```