


The code below can be used to make a countdown timer in Qualtrics that crosses pages with a variety of functionalities (e.g., blocking participants from moving on or auto-advancing them based on the timer).

If you use the code, please cite the following article in your methods section as to where the code derived from:

Brodsky, A., & Amabile, T. M. (2018). The downside of downtime: The prevalence and work pacing consequences of idle time at work. *Journal of Applied Psychology, 103*(5), 496-512.

- 1) Under survey "Look and Feel" make sure the transitions are set to "None."
- 2) To initiate the timer, add the following line in the JavaScript (addOnLoad) section of the first question where you want to use the timer: `add_timer(sec);`
 - a. Replace "sec" with the time, in seconds, you want the timer to run for.
- 3) On each page that will use the timer (including the first one), add this line to the JavaScript (addOnLoad) section of the first question on the page: `timer(this, 'auto-advance', 'disable');`
 - a. 'auto-advance' will move the subject on when the timer expires. Change it to 'no-advance' if you want them to click the button to move ahead.
 - b. 'disable' will disable the Next button until the timer expires. Change it to 'enable' to allow them to click any time.
- 4) You can display the time remaining or not on each page. The timer will work regardless of whether the time is displayed.
 - a. To display the timer, add this to the HTML view of a question where you want the timer to display: ``
- 5) Also, at any point, you can use the `add_timer(sec);` function in the JavaScript for a question to create a new timer that will override the previous one.
- 6) If you want the timer to pause, add this line on the page you want the timer to pause to the JavaScript (addOnLoad) section of the first question: `pause_timer();`
- 7) Then, on the page you want the timer to continue, add this line to the JavaScript (addOnLoad) section of the first question: `restart_timer();`
- 8) If you want, there is an embedded variable called `remaining_time` with whatever the current time left is. This is the 'raw' time (that is, unformatted) so it is in milliseconds. Create it by making an embedded variable with `remaining_time` and leaving the assigned value blank. Make sure that you create it at or near the top of the survey flow, before the block with the timers.
- 9) So that the timer works, add all of the following code below to the Header Source [to get there, go to survey Look & Feel -> click the 'edit' button below the header bar-> click the source button (looks like this: ) ->copy and paste the code]

```
<script> function add_timer(sec){  
  
    //var advance = arguments[2];  
  
    if(isNaN(sec)){  
  
        alert("Invalid starting time");  
        return;  
    }  
}
```

```

        // convert start time from seconds to milliseconds
        var time = sec * 1000;

        var currentTime = new Date();

        var startTime = currentTime.getTime();
        var endTime = new Date(currentTime.getTime() + time);

        createCookie('qualtrics_timer', endTime);
    }

function timer(q, advance, button){

    if(button == 'disable'){
        q.disableNextButton();
    }

    // Subtracts the current time from the initial time value stored in the cookie
    time_now = new Date();
    time_num = time_now.getTime();

    the_end = readCookie('qualtrics_timer');
    the_end = new Date(the_end);
    ending = the_end.getTime();
    remaining = ending - time_num;

    clock = document.getElementById('timer');

    Qualtrics.SurveyEngine.setEmbeddedData('remaining_time', remaining);

    // If there is any tme remaining, it displays it
    if(remaining > 0){

        if(clock){
            display = display_time(remaining);
            clock.innerHTML = display;
        }

        setTimeout(function() {
            timer(q, advance, button);
        }, 1000);
    }

    // If no time is left, the timer is set to display zero
    else {

```

```

        if(clock){
            clock.innerHTML = '00:00';
        }
        if(advance == 'auto-advance'){
            q.clickNextButton();
        }

        if(button == 'disable'){
            q.enableNextButton();
        }
    }
}

function pause_timer(){

    time_now = new Date();
    time_num = time_now.getTime();

    the_end = readCookie('qualtrics_timer');
    the_end = new Date(the_end);
    ending = the_end.getTime();
    remaining = ending - time_num;

    createCookie('qualtrics_pause_time', remaining);

}

function restart_timer(){

    var remaining = parseInt(readCookie('qualtrics_pause_time'));

    var currentTime = new Date();

    var endTime = new Date(currentTime.getTime() + remaining);

    createCookie('qualtrics_timer', endTime);

}

function createCookie(name,value) {

    var date = new Date();

```

```
date.setTime(date.getTime()+ (24*60*60*1000) ); // expires in 1 day
var expires = "; expires="+date.toGMTString();
```

```
document.cookie = name+"="+value+expires+"; path=/";
```

```
}
```

```
function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
    }
    return false;
}
```

```
function display_time(timer){

    var theTime = new Date(timer);

    var minutesDisplay = pad(theTime.getMinutes(), 2);
    var secondsDisplay = pad(theTime.getSeconds(), 2);

    return minutesDisplay+':'+secondsDisplay;

}
```

```
function pad(number, length) {

    var str = " " + number;
    while (str.length < length) {
        str = '0' + str;
    }

    return str;

}
</script>
```