Making a Timer in Alice
Step 1: Adding the Text Object

This tutorial will teach you how to make a timer in Alice. Timers can be very useful if you are interested in making timed games. Start a new Alice world. Now add a text object to that world. When it asks you what you want the text object to say, type in 0.0.

Now in your object tree right click on 0.0 and rename it timer. Now let’s get started coding our timer.
Step 2: Creating a Variable

We need to have a place where we can know how much time is left on the timer, and where we can subtract from how much time is left whenever a second goes by. We will need to create a variable. The key to a variable is that it can change, or vary, and the number of seconds left needs to change a lot for our timer.

Click on timer in your object tree, and then go to the properties tab. At the top of the properties, you should see the create new variable button. Click on that, and create a Number variable named timeLeft. For now, set its value to 0. That 0 is just a placeholder. We will write code in the method editor so that we can enter in whatever value we want before we play the world.
Step 3: Creating an Initialize method

Now we’ll create a method that will play at the very beginning of the world and will set a value for `timeLeft`, which will be the number of seconds the timer will count down. Create a class-level method for `timer` called `initialize`. The only command we’ll need in this method is one that sets the value of `timeLeft`. So click on `timeLeft` and drag it into the `initialize` method. When the number menu pops up, just click 1 for now.
Step 3: cont.

We want to be able to set `timeLeft` to whatever we want. What if we use the timer more than once in the same world with different values every time? We can’t just go into the code and change it every time we want the timer to start over with a different number of seconds. So let’s create a number parameter in `initialize` called `amountOfTime`. After you create it, drag and drop it over the 1 in your `set value to` command. Now we can set the number to a different value every time we use a timer, without having to change the `initialize` code.

Now drag your `initialize` method into `world.my first method` so that it happens right when your world starts. Set `amountOfTime` to any number you want.
Step 4: Creating the **countDown** Method

Now we need to write a method that will actually decrement the `timeLeft` variable, and have our text object display the seconds as they tick down. Create another class-level method, called **countDown**. Drag a **Do in order** inside the method, and then drag a **While** loop inside that.
Step 4: cont.

Now, we only want the timer to keep counting down while there are more than 0 seconds left; we don’t want it to keep counting negative seconds into infinity. Click on world in your object tree and then click on the functions tab. Find the a>b button under math.

Drag that button over the true part of your While loop. Just choose 1 and 0 for the values for now. Now find timeLeft in the timer’s properties tab, and drag it over a. Now whatever we put in the While loop will only happen when timeLeft is more than 0, or there are more than 0 seconds left on the timer.
Step 4: cont.

Drop a **Do in order** inside the **While** loop. Now what we need to do is write code so that whenever **timeLeft** changes values, our **timer** text object changes also. We need to change the text of our text object every time **timeLeft** changes. Click on **timer** in the object tree and then click on the **properties** tab. You should see the **text** button.

Click that button and drag it into the **Do in order** inside the **While** loop. When the method pops up, just click **default string** for now.
Step 4: cont.

Now we want to set our text object to `timeLeft`. But, if we try to drag `timeLeft` over to our command to replace `default string`, it won’t work! This is because we need to set the text object equal to a new `string`, and `timeLeft` is a `number`, not a `string`. We need to turn `timeLeft` into a `string`. To do this, click on `world` and then the `functions` tab, and scroll down until you see `what as a string`.

Now each time your `While` loop starts, your text object is set to display the value of `timeLeft`. Set the duration of this command to 0 seconds so that it’s value is set instantaneously.
Step 4: cont.

So far, when your While loop starts, your timer will display the value of timeLeft. Now we need to make sure that it takes exactly one second before the value is reset. So, we will use the Wait command, which is located under your method editor. Drag that into your Do in order in your While loop and set it to 1 second.
Step 4: cont.

Next, at the end of the While loop, we need to subtract one from the value of timeLeft, to show that a second has gone by. Click on timer in the object tree, and then go to the properties tab. Click on timeLeft and drag it into your method editor right under your Wait command. On the menu that pops up, chose decrement timer.timeLeft by 1.

Now, so that the timer is decremented instantaneously, set the duration of the decrement command to 0 seconds.

Now drag your countDown method into world.my first method under your initialize method and play your world to see what happens.
Step 5: Finishing Up

Notice anything strange about your timer? No matter how many seconds it starts with, it always stops at 1!

This is why: Look at your While statement. It will only repeat itself if timeLeft is greater than 0 at the beginning of the statement. When timeLeft gets down to one, this is what the While statement thinks:

- Is timeLeft greater than 0? Yes it is, it equals one, so full steam ahead.
- Now it’s time to make the timer object display one.
- Now we wait 1 second…
- Now we subtract 1 from timeLeft again, and go back to the beginning.
- Is timeLeft greater than 0? No, it equals 0, so I’ll stop now.

So we need to add a command AFTER your While statement so the timer goes all the way down to 0.
Step 5: cont.

So the only step that’s missing is to set the timer object to the value of timeLeft again. So, go to timer on the object tree and then go to the properties tab. Find the text button and drag it into your countDown method under your While statement. Reproduce the same timer set text to command that you have inside your While statement. Your final code will look like this:

![Diagram of the object tree and code blocks]

Now play your world again, and observe that sweet sweet timer action!
This timer can be very useful for games in which you have to beat the clock. You can also use these concepts to create a scorekeeper. In that case, instead of decrementing your variable every time a second goes by, you would increment your variable every time someone scores a point in your game. Happy scorekeeping!