

Scoring

This sheet explains how to add some text to the screen, and how to keep a score.

Text

When someone starts programming, it's traditional to for them to introduce themselves by saying "Hello world!". Let's do that now.

Create a `Text` object, and set its text.

```
var label = new Text
label.text = "Hello world!"
```

Text color

You can change the colour of text. The sort of colors you can use are the same ones we used to [00-new-world#change-the-background-colour](#) of the world. Instead of `world.background`, we use `label.fill`.

```
label.fill = '#007de0'
```

👉 Change it to your favourite color.

Keeping Score

Our label has a `text` attribute, but we can't use that to keep score, because it's a `String` (a piece of text), rather than a number. If we try and add two strings together, they will be joined rather than mathematically summed!

You can try typing `'10' + '1'` in the Console to see what happens when you add two Strings. Compare it with `10 + 1` (adding two numbers).

So, let's make a separate variable which stores our score as a number.

```
var score = 0
```

👉 Add a score variable to the top of your program.

Updating the score

You can update the score like you update any other number.

```
score += 1
```

👉 Put this wherever you want to give the player a point.

Displaying the score

We need to take what's in our score variable and display it on the screen. Let's create a Text object to show it.

```
var scoreLabel = new Text  
scoreLabel.text = "score: " + score
```

👉 Put this at the top of your program, right after where you made the score variable.

You may notice the text doesn't update, even when you update the score variable. You'll need to write another line of code to update the text whenever the score changes:

```
score += 1  
scoreLabel.text = "score: " + score
```

👉 Update the text whenever you give the player a point.

High score

You can easily extend this to keep a high score as well:

- Create a `highScore` variable
- Use an `if` condition to update the `highScore` variable after you update the score
- Create a `highScoreLabel` to display it on the screen
- Use `localStorage` to save the high score, so it doesn't get forgotten when you refresh the page