Bird watch website 1.0

Learn to make a website

Step 1 Introduction

Learn how to code your first website!

What you will make

Build a website like the one in the trinket below.

What you will learn

- Using an online editor to create a website made up of HTML files and a CSS file
- Building a HTML website that has headings, paragraph text, lists, and images
- Using CSS code to control the look of your website, including designing a simple menu bar and changing colours, backgrounds, and borders
- Linking pages to each other, and linking to other websites
- Creating a table that includes headings and multiple rows

What you will need

Hardware

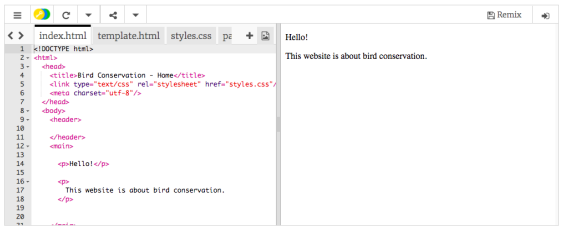
- A computer capable of accessing **trinket.io** (https://trinket.io)

Software

- This project can be completed in a web browser using **trinket.io** (https://trinket.io).

Step 2 Getting set up

- Go to **the starter trinket** (http://dojo.soy/se-html1-start). You will see a box containing an example website project. On the right-hand side is the website, and on the left-hand side is the code that makes the website.



Let's start coding!

Step 3 Your first webpage!

- In the left-hand panel, the **code panel**, click on the tab that says `index.html`.
- Find the line that says `Hello!` and change it to your own message — be careful **not** to delete the tags`<p>` at the start of the line and `</p>` at the end of the line. You should see your webpage update in the right-hand panel.

- Now on the same line, change the `<p>` and `</p>` to `<h1>` and `</h1>`. Do you notice any change in the result on the right?

```
<h1>Hello!</h1>
```

- Try changing the numbers in your **heading** tags to see the different sizes they give you. They can go from `<h1>` all the way up to `<h6>`. Remember to change both the opening and closing tag so that they match.
- Find the code for the paragraph that says 'This website is about bird conservation.' and change it so that it looks like this:

```
<p>
  <em>This website</em> is about <strong>bird conservation</strong>.
</p>
```

Can you work out what the `<em> </em>` and `<strong> </strong>` tags do?

Challenge: add some more text of your own

- Try adding a new paragraph or heading to your page using some of the tags you've learned about.

The code for headings looks like this:

```
<h1>This is a heading.</h1>
```

Headings will normally be displayed bigger or bolder than the paragraphs.

Congratulations, you've built your first webpage! On the next card, you'll find out how to control how it looks.

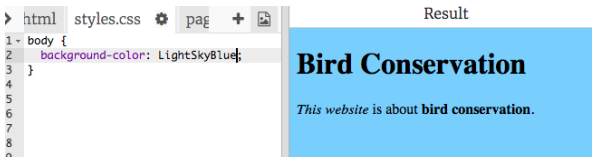
Step 4 Controlling how it looks

The code that describes what a website looks like is called **CSS**.

- Look at the tabs at the top of the code panel, and go to the file `styles.css` by clicking on the tab with that name.
  The file contains the following text:

```
body {
    background-color: white;
}
```

- Change the `white` colour to `LightSkyBlue` and see what happens. Your website should now have a blue background!



- Lets add rules to change how the text looks. Add two new lines inside the curly braces:

```
body {
  background-color: LightSkyBlue;
  font-family: "Helvetica", sans-serif;
  color: purple;
}
```

- Look at how this has changed the webpage.

The `color` property is always for text. Here, you are setting the colour of all text in the `body` of your webpage.

- You can also write separate rules for the headings and the paragraphs. For `<h1>` headings, you use the `h1` selector. Below the closing curly brace containing the CSS rule for the body, add the following code.

```
h1 {
  color: orange;
  font-family: "Times New Roman", serif;
}
```

Your heading text should be orange now, with the paragraph in purple as before.

Notice how the letters also look different as well as being a different colour? This is because you changed their **font family**. You can find some more fonts **here** (http://dojo.soy/se-font-families).

- Try adding a set of rules for the `<h2>` headings, using the `h2` selector.
- Why not experiment with different colour combinations for the text and background? There are lots of colours available to use. Find a full list of them **here** (http://dojo.soy/se-color-names).

Step 5 Adding pictures

Let's add a picture!

- Go to the tab named `index.html`. Find the `</main>` tag and type the following **above** it:

```
<img src="barn-owl.jpg" alt="A barn owl" width="200px" />
```

Here's what the result should look like:

Notice that this tag has extra bits of information inside it. They are called **attributes**.

- Find the bit of code that says `width="200px"` and try experimenting with different numbers to see if you can figure out what this attribute does. Don't delete the letters `px`!

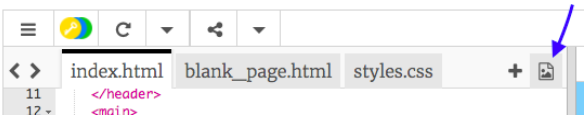
Now that you know the code to put a picture on your website, you probably want to change the picture, right?

- The first thing you will need is, of course, a picture! You can either use one you've already got on your computer, such as a photograph you took, or you can get one from the internet.

Note: not all images you will find on the internet are free for anyone to use. If you download a picture, you should make sure it is one that you are allowed to use. Find out more about this here:

Once you have a picture, you can **upload** the file to Trinket:

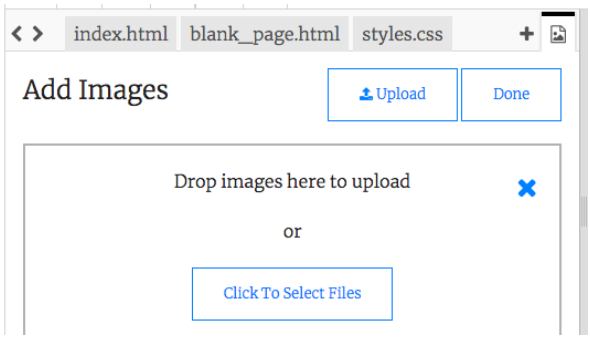
- In your trinket, click on the **image** icon next to the **+** sign.



This is where you can see the pictures that you are able to use on your website. You should see the picture of the barn owl.

- Click the button **Add Image** and then click **Upload**.

- Click on the button **Click To Select Files**. Find and double-click your image file in the window that opens.
- Click **Done**.



Your picture will be uploaded and should be ready to use.

- Go to the file `index.html` and find the `<img>` tag. Change the text `barn-owl.jpg` so that it exactly matches the name of the image file you've chosen. Note that its name might end in `.png` instead of `.jpg`!

The text you just changed is the attribute called `src`, which tells the browser which file to display.

Note: the value you type for an attribute must have quotation marks `""` around it!

Challenge: change the alt text of the picture

- Find the `alt` attribute of your image element and change the text in it to a short description of your picture.

Step 6 Adding a map or video

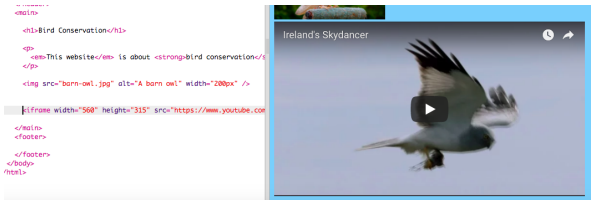
YouTube provides an easy way to add its videos to your website. Adding elements from other online sources to your website is also called **embedding**.

- Find a video on YouTube that you want to show on your website.
- Click on the **Share** button below the video. Select the option **Embed**.

You will see a text box with all the text selected. If you accidentally unselect the text, you can select it all again by clicking on it and pressing the `Ctrl` (or `cmd` on a Mac) and `A` keys at the same time.

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/nqQk711V4Vc?
rel=0" frameborder="0" allow="autoplay; encrypted-
media" allowfullscreen></iframe>
```

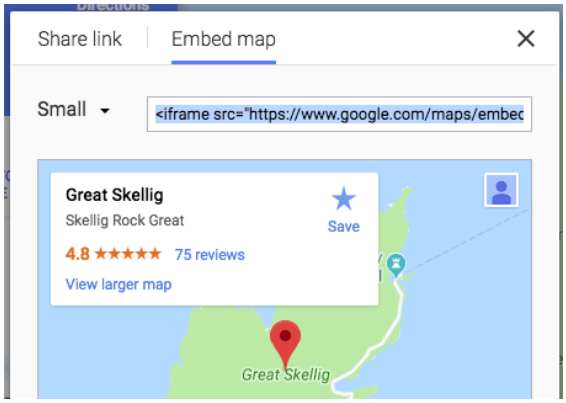
- Press the `Ctrl` (or `cmd` on a Mac) and `C` keys together to copy the text.
- Then go back to your website's HTML code, and click in the place where you want to put the video, for example below a heading or paragraph. **Paste** the code by pressing `Ctrl` (or `cmd` on a Mac) and `V` on your keyboard at the same time. Don't worry about understanding all the code you just pasted!



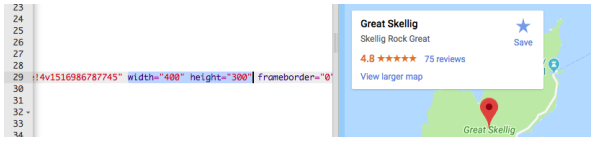
You should see the video appear on your webpage.

The same technique works for Google maps as well. Give it a go!

- Go **here** (http://dojo.soy/se-maps) and search for a place you want to show on your website. **Note:** do **not** share personal information such as your home address on a website!
- Click on the result, then click the **Share** button, and copy the code and add it to your website as above.



- If you look carefully, you should find `width` and `height` **attributes** in the pasted code. You can change their values to make the map appear bigger or smaller.



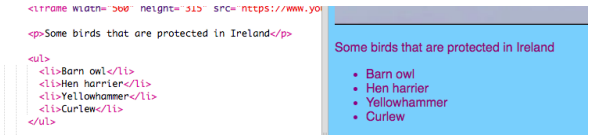
Step 7 Making a list

Now you will learn how to turn a list of items, such as “unicorns, robots, cats”, into a nicer-looking list that you can do cool things with later.

- In the `index.html` file, add the following code just above the line with `</main>` on it:

```
<ul>
    <li>Barn owl</li>
    <li>Hen harrier</li>
    <li>Yellowhammer</li>
    <li>Curlew</li>
</ul>
```

The result should be a nice list like this:



Notice that there is a separate pair of `<li> </li>` tags around each item in the list.

This is a list of some protected birds in Ireland. You can change the items on the list to things that make sense for your website, and add a paragraph above the list to describe what it’s a list of, if you like!

How about if you wanted a numbered list? It’s almost the same, but instead of `<ul>`, you use `<ol>`. A numbered list is also called an **ordered** list.

- Add the following code below the code you just wrote — make sure it’s **below** the `</ul>` tag!

```
<p>
    Threats to birds:
</p>
<ol>
    <li>Habitat destruction</li>
    <li>Pollution</li>
    <li>Climate change</li>
</ol>
```

Here’s what it should look like now:

Challenge: add style to your lists

- See if you can add **CSS rules** to your stylesheet to change how your lists look.

Step 8 Creating links

On this card you'll learn how to make a link that takes you to another page when it's clicked.

- Add the following code to the body section of `index.html`:

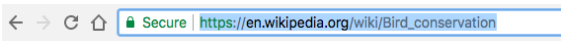
```
<a href="">Click here</a>
```

The `<a> </a>` tags turn whatever is in between them into a link.

- Try clicking your link to see what happens. It does nothing, right?

That's because the `href` attribute is empty at the moment. It needs to contain the **URL** (web address) of the page that you want to link to.

- Go to Wikipedia and find a page about something on your website. I'm going to use the page about bird conservation.
- Click in the address bar and select all of the text in i5. That's the complete URL of the page you're on. Press the Ctrl (or cmd) and C keys at the same time to copy it.



- In your trinket, click in between the quotation marks after `href=` and press the Ctrl (or cmd) and V keys at the same time to paste in the URL you just copied. Your code should look something like this now:

```
<a href="https://en.wikipedia.org/wiki/Bird_conservation">Click here</a>
```

You just created your first link! Click on it to see if it works now.

- Try putting a picture in between the `<a> </a>` tags instead of the words `Click here`, like this:

```
<a href="https://en.wikipedia.org/wiki/Bird_conservation">
    <img src="barn-owl.jpg" alt="A barn owl" width="200px" />
</a>
```

- Click on your picture. Do you see that it was turned into a link?

You can put a link into other elements of your webpage too, such as in a paragraph or even in a list. Here is an example of a sentence with a link in it:

```
<p>
  <a href="https://en.wikipedia.org/wiki/Bird_conservation">Click here</a> to read about bird
conservation on Wikipedia.
</p>
```

Challenge: put a link into a list

- See if you can make a list that contains a link inside one of the list items.

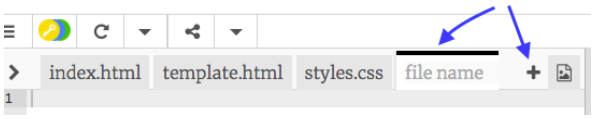
In the following list, the 'Hen harrier' list item has been turned into a link.

```
<ul>
    <li>Barn owl</li>
    <a href="https://en.wikipedia.org/wiki/Hen_harrier"><li>Hen harrier</li></a>
    <li>Yellowhammer</li>
    <li>Curlew</li>
</ul>
```

Step 9 Adding more pages

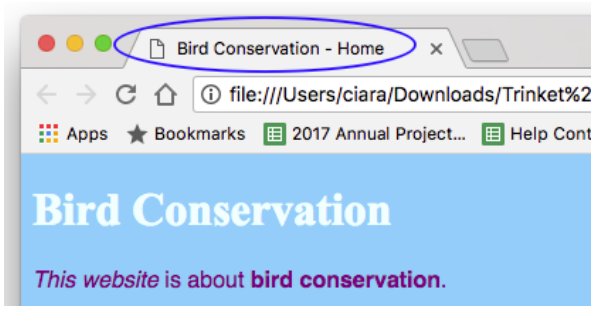
This card will show you how to add more pages to your website.

- At the top of the code panel, click on the **+** symbol next to the tabs, and type in a name for your new file. It must end in `.html` (including the dot!) so that the browser knows it's a webpage.



- Find the file `template.html` and copy and paste all of the code from it into your new file. Since you want to copy the whole thing, you can click anywhere on the code and use the keyboard shortcut `Ctrl` (or `cmd`) and `A` to select all of it at once.

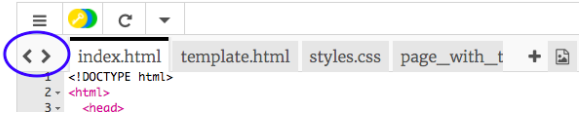
- Change the text in between the `<title> </title>` tags so your new page has a suitable title. Trinket won't display the title, but you can see it at the top of your browser window if you download your project.



- In between the `<main> </main>` tags in the new file, use the tags you have learned about to add stuff to the page, such as paragraphs, headings, images, and lists!

- Repeat the steps above for each new page that you want to add.

When there are too many tabs for Trinket to show at once, you can use the **<** and **>** icons in the top left-hand corner of the tabs to scroll between them.



Now you need to make links so that you can get to each of your new pages! Let's put all the links in a list.

- In the `index.html` file, add the following code to the body of your webpage:

```
<ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="birds.html">Protected Birds</a></li>
    <li><a href="conservation.html">Conservation</a></li>
    <li><a href="sanctuaries.html">Bird Sanctuaries</a></li>
</ul>
```

- Change the value of the `href` attribute for each link (remember, that's the text inside the quotation marks) so that it exactly matches the name of each HTML file that you have created.
- Change the text in between the `<a> </a>` tags to suitable descriptions of your pages.

Now you can navigate to your new pages!

Step 10 Navigating your website

Many websites have a **navigation** menu to help visitors move between pages. Now that you've got a bunch of pages, a homepage, and links to each page, let's move the list of links to a navigation section at the top of every page.

- Find the code for your list of links that you created in the previous step.
- Just before the opening `<ul>` tag, press **Enter** to create a new blank line, then on the new line type the following tag: `<nav>`. Trinket automatically adds the closing tag right after, but you can delete that — it's not in the right place.
- Just **after** the closing `</ul>` tag, press **Enter** to create a new blank line, and type in the closing tag `</nav>` there.
- Now select your entire `<nav>` section and list by clicking just before the opening `<nav>` tag and dragging the mouse all the way down to just after the closing `</nav>` tag, so that all of the text including the opening and closing tags becomes highlighted. Make sure all of the **angle brackets** `<` and `>` at the start and end are highlighted as well!

- You are going to **cut** this time instead of copying. Hold down the `Ctrl` (or `cmd`) key, and while holding it, press the `X` key. The highlighted code will disappear, but don't panic!

- At the top of the file, click in the space between the `<header> </header>` tags. Make sure you see the cursor flashing there. Now paste in the code by pressing **`Ctrl`** (or **`cmd`**) and **`V`** as usual. The code should look something like this:

```
<header>
    <nav>
        <ul>
            <li><a href="index.html">Home</a></li>
            <li><a href="birds.html">Protected Birds</a></li>
            <li><a href="conservation.html">Conservation</a></li>
            <li><a href="sanctuaries.html">Bird Sanctuaries</a></li>
        </ul>
    </nav>
</header>
```

- Try out your links to make sure they are still working.

Challenge: navigation menus for all pages

- Put this code section into the header section of each HTML file that you've created. This will make the navigation menu appear at the top of every page on your website.

Select the entire `<nav>` section like you did before, and press the **`Ctrl`** (or **`cmd`**) and **`C`** keys together to copy it.

Then, in each of your `.html` files, click inside the `<header> </header>` section and paste the code exactly like you did earlier.

Now you will be able to click the links no matter which page you are on.

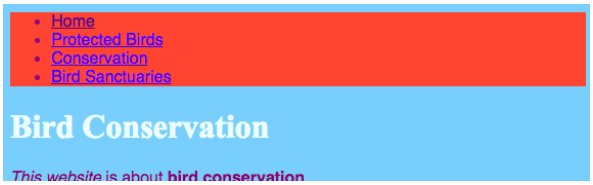
Step 11 Making a menu bar

On this card you will see how you can transform your navigation menu into a cool-looking menu bar, just by adding more CSS rules in the style sheet.

- Go to the style sheet file in the `styles.css` tab. Click **below** a closing curly brace `}`, and press **Enter** to create a new blank line. Add the following CSS rule:

```
nav ul {
    background-color: tomato;
}
```

Notice how you used two selectors instead of one? If you used the `ul` selector on its own, the rule would affect all unordered lists on your website. Adding the `nav` selector as well makes it only apply to lists that are in between `nav` tags.



Let's get rid of the bullet points. Those are the dots in front of each list item.

- Add the following to the `styles.css` file. Again, type it on a new line after a `}` so it's not inside any other block of rules.

```
nav ul li {
    list-style-type: none;
}
```

Notice this set of rules has three selectors: it selects all `li` elements that are in a `ul` list which is inside a `nav` section. Phew!

Now let's make the list horizontal (across) instead of vertical (down).

- Inside the new CSS rule you just created, add the following line: `display: inline;`.

- The menu items are now all squashed together, so let's also add the properties `margin-right` and `margin-left` to space them out a bit. The block of CSS code should look like this now:

```
nav ul li {
    list-style-type: none;
    display: inline;
    margin-right: 10px;
    margin-left: 10px;
}
```

Remember: `10px` means ten pixels.

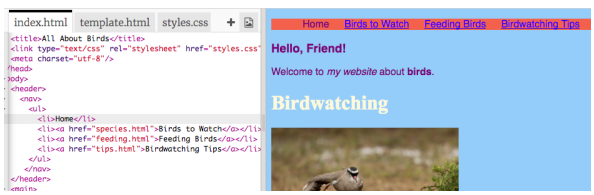
How about making the menu change to tell you which page you are on? This part won't be in the style sheet.

- Start with the homepage. Go to the `index.html` file. In the list of menu links, remove the link tags before and after the word `Home`, so that the list item for the homepage is just text in between `<li> </li>` tags, like this: `<li>Home</li>`.

- Now go to each of your other files, and do the same thing, each time removing the link tags for the page you are editing. So, for example, on the `birds.html` file, I've removed the link tags in the `Protected Birds` list item:

```
<header>
    <nav>
        <ul>
            <li><a href="index.html">Home</a></li>
            <li>Protected Birds</li>
            <li><a href="conservation.html">Conservation</a></li>
            <li><a href="sanctuaries.html">Bird Sanctuaries</a></li>
        </ul>
    </nav>
</header>
```

- Explore your pages by clicking the links. See how the menu bar shows the page you're on as plain text instead of a link?



On the next card you'll learn even more CSS tricks to make the menu bar look awesome.

Step 12 Styling the menu bar

With CSS, the possibilities for making your menu bar look great are endless.

- Move to the `styles.css` file again — the place where the cool stuff happens!
- Find your `nav ul` selector, and add more rules so that it looks like this:

```
nav ul {
  background-color: tomato;
  border-style: solid;
  border-color: MediumVioletRed;
  border-width: 2px;
  padding: 10px;
}
```

The `padding` property adds space. Can you work out what each of the other properties do? Try experimenting with different colours and numbers of pixels.

- To get rid of the underlining of the links, add the following code on a new line after the closing curly brace `}` for the `nav ul li` rules. You could put it after any `}`, but it's a good idea to keep related stuff together so it's easier to find!

```
nav ul li a {
    text-decoration: none;
}
```

The above rule applies to links `<a>` inside list items `<li>` in an unordered list `<ul>` inside a navigation section `<nav>`. Wow, that's four selectors!

Remember how you removed the link tags from some list items in the `<nav>` so you can easily see what page you're on? Why not also change the text colour of those navigation list items which are not links!

- Find your `nav ul li` selector, and **inside** the curly braces add the line:

```
color: PapayaWhip;
```

You can choose any colour you like!

You can add the `color` property to the `nav ul li a` rule as well if you want the menu links to be a different colour from other links on your website.

- How about some rounded corners for your menu? Try adding the following code to the `nav ul` rule to see what happens: `border-radius: 10px;`.

The `border-radius` property is a really easy way to make anything look cooler!

Challenge: make your pictures have rounded corners

- In your style sheet, create a new set of rules for pictures using the `img` selector, and add in a `border-radius` rule there.

Step 13 Adding a table

Sometimes it can be useful to show information in a table. For example, you might want to list member information on a website for a local sports club or school, or information about your top ten favourite songs.

A table is a grid made up of **rows** and **columns**. Most tables also include titles at the top of each column, called the **header**. Here's an example:

Name	Location	Birds found there
Skellig Michael	Island off coast of Kerry	Puffins
Bird Island	Strangford Lough	Cormorants
Burren Birds of Prey Centre	County Clare	Various birds of prey including eagles, hawks, owls and falcons

- Go to the file `page_with_table.html`. There you will see a bunch of code in between `<table> </table>` tags.
- Select all of the code from the start of the `<table>` tag to the end of the closing `</table>` tag and copy it. Then go to one of your files where you would like to put a table, and paste in the code.

At the moment your table is empty.

- Have a go at filling your table with anything you like! Simply put text in between the `<td> </td>` tags and in between the `<th> </th>` tags. You can add more tags if you need them.

To add another **row**, add another set of `<tr> </tr>` tags. In between them, you put the same number of **data** items with `<td> </td>` tags as you have in the other rows.

To add another **column**, add an extra **data** item with a set of `<td> </td>` tags to **every** row. Also add an extra **header** item to the first row, using `<th> </th>` tags.

- If you look at the end of the `styles.css` file, you will see the CSS code that describes how the table should look. You don't have to understand all of it! But you can experiment with changing the text, border, and background colours to design your own style.

```
table, th, td {
  border: 1px solid HoneyDew;
  border-collapse: collapse;
}
tr {
  background-color: PaleTurquoise;
}
th, td {
  vertical-align: top;
  padding: 5px;
  text-align: left;
}
th {
  color: purple;
}
td {
  color: purple;
}
```

Notice how some of the selectors use commas, for example `table, th, td`? That's a **list of selectors**: it means it applies to all `<th>` elements and all `<td>` elements. It saves typing out the same set of rules for each selector!
