
Chapter 3

Cascading Style Sheets

What is in This Chapter ?

This chapter explains how to use **Cascading Style Sheets** to style the content on our webpages. Again, this is only a very basic introduction. There is a lot more that we can learn. As a nice example, we will use **HTML** and **CSS** to create a nice-looking website for a fictional company. We will continue working on the website in later chapters.



3.1 The Basics

Now that we know how to place content on our page, it is time to discuss ...

CSS (*Cascading Style Sheets*) - *the language used to style and visually format the contents of a web page.*

CCS allows us to control things on our webpage such as colors (text and background), fonts (types and size) and layout (margins, padding, grid).

A webpage without any styling will use the default browser settings ... which is a boring black standard-size text on white background.

While we can use plain **HTML** (with its various tags and attributes) to obtain webpage styling that is somewhat visually appealing, there are some significant advantages of using **CSS**:

✓ Better Styling Control

- We can easily style **HTML** elements (i.e., colors, fonts, spacing, etc.) better than using just **HTML** ... which is very limited.

```
<h1 style="color: blue; font-size: 32px;">Styled heading</h1>
```

✓ Cleaner and More Easily-Managed Code

- **CSS** keeps the styling (i.e., design) separate from the **HTML** (i.e., content).

```
<h1 class="title">Welcome</h1>
```

← put this in the **.html** file

```
.title {  
  color: darkgreen;  
  font-family: Arial;  
}
```

← then put this in the **.css** file

✓ Reusability and Efficiency

- We can style thousands of elements with just a few lines of **CSS**, instead of repeating styles in every tag.

```
p {  
  color: gray;  
  font-size: 16px;  
}
```

← This styles all paragraphs consistently

✓ Responsive Design (more on this later)

- **CSS** makes it easy to adapt our site to different devices (i.e., phones, tablets, and desktops).

```
@media (max-width: 600px) {  
  body {  
    font-size: 14px;  
  }  
}
```

✓ Easier Maintenance

- Change one **CSS** rule, and it updates across the entire site, as opposed to needing to change every **HTML** tag manually.

✓ Animations and Transitions

- With **CSS**, we can animate buttons, menus, and effects ... which are things that **HTML** cannot do alone.

```
button:hover {
    background-color: red;
    transition: 0.3s;
}
```

For an example that shows the power of **CSS**, go to this website:

<http://www.csszengarden.com/>

It is a website where each webpage consists of the exact same **HTML** code but different **CSS** files are used to make the page look completely different!

There are three different ways to use **CSS**:

- **Inline CSS** (not good for large projects):

```
<p style="color: red;">This is red text</p>
```

- **Internal CSS** (inside the `<style>` tag):

```
<head>
  <style>
    p {
      color: red;
    }
  </style>
</head>
```

- **External CSS** (best practice):

```
<link rel="stylesheet" href="styles.css">
```

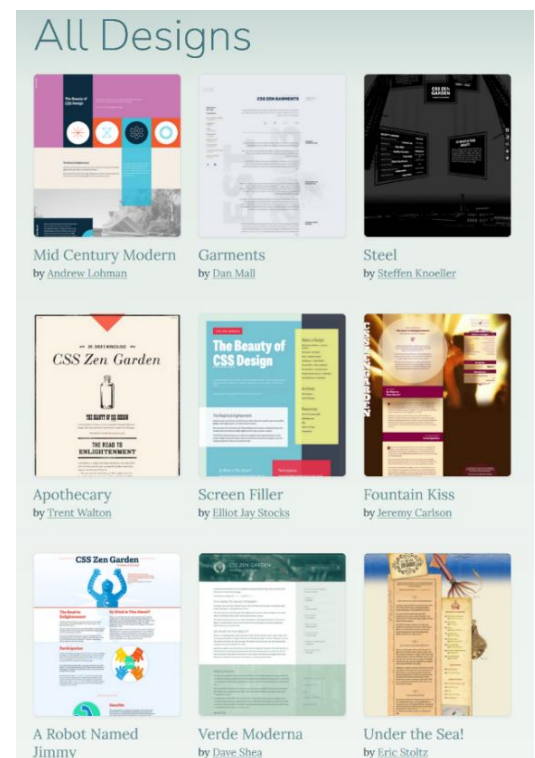
← in the **.html** file

```
p {
  color: red;
}
```

← in the **styles.css** file

In all three cases above, we are simply setting the paragraph color to red by using the `color` property specifier.

There are lots of **CSS** properties ... here are just a few:



Property	Description	Example
color	text color	"color: red;"
background-color	page background color	"background-color: yellow;"
font-size	size of text	"font-size: 18px;"
font-family	font style	"font-family: Arial;"
text-align	aligns text	"text-align: center;"
margin	space outside an element	"margin: 20px;"
padding	space inside an element	"padding: 10px;"
border	adds border	"border: 1px solid black;"
width / height	sets size	"width: 300px;"

For a more thorough reference to **CSS** options, please visit this website:

<https://www.w3schools.com/css/default.asp>

Let's do one example. I added a table to our earlier **HTML** file and called it **dentalHygiene(BeforeCSS).html**. Let's see if we can put some inline and internal **CSS** code to color the text.

What happens when we add this code within the **<head>** tag wrapper?

```
<style>
p {
  color: rgb(170, 7, 7);
}
</style>
```

We will notice that all paragraphs end up with red text. Now ... let's change the font for the main header using inline **CSS**.

We change this:

```
<h1>Dental Hygiene for Dolphins</h1>
```

to this:

```
<h1 style="font-family: 'Papyrus', fantasy;">Dental Hygiene for Dolphins</h1>
```

Then we can center it on the page like this:

```
<h1 style="font-family: 'Papyrus', fantasy; text-align: center;">
  Dental Hygiene for Dolphins
</h1>
```

Here is what it looks like →

Dental Hygiene for Dolphins

Why Dolphin Dental Care Matters

Dolphins, like humans, rely on their teeth for everyday activities. Their teeth are essential not just for eating, but also for grasping objects and communicating through jaw movements. In the wild, dental health can affect a dolphin's ability to hunt, socialize, and survive. In captivity, poor dental care can lead to infection, loss of appetite, and serious health complications. That's why dental hygiene is an important part of dolphin care in marine environments.

Differences from Human Teeth

Unlike humans, dolphins only grow one set of teeth in their lifetime. They do not chew their food; instead, they swallow it whole. Most dolphins have between 80 to 100 conical-shaped teeth, depending on the species. Because they don't replace lost

Dental Hygiene for Dolphins

Why Dolphin Dental Care Matters

Dolphins, like humans, rely on their teeth for everyday activities. Their teeth are essential not just for eating, but also for grasping objects and communicating through jaw movements. In the wild, dental health can affect a dolphin's ability to hunt, socialize, and survive. In captivity, poor dental care can lead to infection, loss of appetite, and serious health complications. That's why dental hygiene is an important part of dolphin care in marine environments.

Differences from Human Teeth

Unlike humans, dolphins only grow one set of teeth in their lifetime. They do not chew their food; instead, they swallow it whole. Most dolphins have between 80 to 100 conical-shaped teeth, depending on the species. Because they don't replace lost

How Dental Checkups Are Performed

Veterinary Techniques

Marine veterinarians regularly examine dolphin teeth as part of routine health assessments, making it easier for vets to inspect their gums and tooth surfaces. Teeth like used. In some cases, digital X-rays may be taken to check for problems below the gum line.

Behavioral Training for Exams

Dolphins are trained using positive reinforcement to participate in their own health care. They perform behaviors like holding still or opening their mouths. This cooperative training is essential for safe and effective dental exams. It also strengthens the trust between dolphins and their caretakers.

Common Dental Issues in Dolphins

Some dolphins experience tooth wear from rubbing their teeth on pool walls or chomping on toys. Others may have gum disease, known as periodontal disease, which can lead to infection, abscesses, or even systemic illness.

Preventive Measures

Providing environmental enrichment and a well-balanced diet can reduce harmful behaviors that lead to tooth damage. Regular monitoring and professional cleanings are also crucial for maintaining dental health.

Dolphin Name	Teeth Count	Last Cleaning	Health Status
Flipper	88	2025-06-01	Excellent
Splash	86	2025-05-15	Good
Echo	89	2025-06-10	Fair (plaque buildup)

Note: Research on dolphin dentistry is ongoing, and practices may vary between facilities. Ensuring dolphin welfare involves a combination of preventive care and professional intervention.

Now, let's indent all the paragraphs. Since we want this to be for all paragraphs, we add the style within the `<head>` tag wrapper, along with our color setting:

```
<style>
  p {
    color: rgb(170, 7, 7);
    text-indent: 40px;
  }
</style>
```

This will indent the start of each paragraph by **40** pixels. But, what if we want the whole paragraph to be indented, not just the first line? Well, we can set the left margin for each paragraph instead:

```
<style>
  p {
    color: rgb(170, 7, 7);
    margin-left: 35px;
  }
</style>
```

Now, let's fix up our table with a border and some spacing. We can change the table to have internal **CSS** to set the border, border color and padding for each cell by adding it within the `<style>` tags in the `<head>` section of the document.

```
<style>
  p {
    color: rgb(170, 7, 7);
    margin-left: 35px;
  }
  table {
    border: 2px solid rgb(0, 0, 255);
    border-collapse: collapse; /* gets rid of internal spacing around cells */
  }
  th, td {
    border: 1px solid rgb(0, 0, 255);
    padding: 8px; /* spacing around the text of each cell */
  }
</style>
```

Then, we can center-align the table itself by using inline **CSS**. There are a few ways to do this, but one of them is to set the left and right margins to **auto**:

```
<table style="margin-left: auto; margin-right: auto;">
```

The result is this:

Dental Hygiene for Dolphins	
Why Dolphin Dental Care Matters Dolphins, like humans, rely on their teeth for everyday activities. Their teeth are essential not just for eating, but also for grasping objects and communicating through jaw movements. In the wild, dental health can affect a dolphin's ability to hunt, socialize, and survive. In captivity, poor dental care can lead to infection, loss of appetite, and serious health complications. That's why dental hygiene is an important part of dolphin care in marine environments.	
Differences from Human Teeth Unlike humans, dolphins only grow one set of teeth in their lifetime. They do not chew their food; instead, they swallow it whole. Most dolphins have between 80 to 100 conical-shaped teeth, depending on the species. Because they don't replace lost teeth, damage or decay can be particularly harmful. Their teeth are also used for social behaviors, including displays of dominance or play.	
How Dental Checkups Are Performed	
Veterinary Techniques Marine veterinarians regularly examine dolphin teeth as part of routine health assessments. Dolphins in managed care are trained to voluntarily open their mouths, making it easier for vets to inspect their gums and tooth surfaces. Tools like underwater cameras, soft brushes, and dental mirrors are commonly used. In some cases, digital x-rays may be taken to check for problems below the gumline.	

Dental Hygiene for Dolphins	
Why Dolphin Dental Care Matters Dolphins, like humans, rely on their teeth for everyday activities. Their teeth are essential not just for eating, but also for grasping objects and communicating through jaw movements. In the wild, dental health can affect a dolphin's ability to hunt, socialize, and survive. In captivity, poor dental care can lead to infection, loss of appetite, and serious health complications. That's why dental hygiene is an important part of dolphin care in marine environments.	
Differences from Human Teeth Unlike humans, dolphins only grow one set of teeth in their lifetime. They do not chew their food; instead, they swallow it whole. Most dolphins have between 80 to 100 conical-shaped teeth, depending on the species. Because they don't replace lost teeth, damage or decay can be particularly harmful. Their teeth are also used for social behaviors, including displays of dominance or play.	
How Dental Checkups Are Performed	
Veterinary Techniques Marine veterinarians regularly examine dolphin teeth as part of routine health assessments. Dolphins in managed care are trained to voluntarily open their mouths, making it easier for vets to inspect their gums and tooth surfaces. Tools like underwater cameras, soft brushes, and dental mirrors are commonly used. In some cases, digital x-rays may be taken to check for problems below the gumline.	

intervention are key to preventing long-term dental damage. Here is a report showing some recent results for our three dolphins:

Dolphin Dental Hygiene Report

Dolphin Name	Teeth Count	Last Cleaning	Health Status
Flipper	88	2025-06-01	Excellent
Splash	86	2025-05-15	Good
Echo	90	2025-06-10	Fair (plaque buildup)

Note:

Research on dolphin dentistry is ongoing, and practices may vary between facilities and regions. Marine mammal specialists continue to develop new tools and methods for improving dental care. Ensuring dolphin welfare involves a team effort between trainers, vets, and researchers.

We can try even changing the whole page to have a light gray background simply by altering the `<body>` tag using inline **CSS** as follows:

```
<body style="background-color: rgb(240, 240, 240);">
```

Dental Hygiene for Dolphins

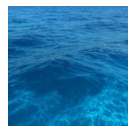
Why Dolphin Dental Care Matters

Dolphins, like humans, rely on their teeth for everyday activities. Their teeth are essential not just for eating, but also for grasping objects and communicating through jaw movements. In the wild, dental health can affect a dolphin's ability to hunt, socialize, and survive. In captivity, poor dental care can lead to infection, loss of appetite, and serious health complications. That's why dental hygiene is an important part of dolphin care in marine environments.

Differences from Human Teeth

Unlike humans, dolphins only grow one set of teeth in their lifetime. They do not chew

Hmmm ... seems kinds plain. Let's use this background image (called [ocean.jpg](#)) of an ocean water instead of setting a background color.



We just need to change the `<body>` tag to this:

```
<body style="background-image: url('ocean.jpg');
background-size: cover;
background-repeat: no-repeat;">
```

Notice that there are some options to ensure that it covers the background and does not repeat. Make sure that the background image has a high enough resolution.

Dental Hygiene for Dolphins

Why Dolphin Dental Care Matters

Dolphins, like humans, rely on their teeth for everyday activities. Their teeth are essential not just for eating, but also for grasping objects and communicating through jaw movements. In the wild, dental health can affect a dolphin's ability to hunt, socialize, and survive. In captivity, poor dental care can lead to infection, loss of appetite, and serious health complications. That's why dental hygiene is an important part of dolphin care in marine environments.

Differences from Human Teeth

Unlike humans, dolphins only grow one set of teeth in their lifetime. They do not chew their food, instead they swallow it whole. Most dolphins have between 80 to 100 conical shaped teeth, depending on the species. Because they don't replace lost teeth, damage or decay can be particularly harmful. Their teeth are also used for social behaviors, including displays of dominance or play.

How Dental Checkups Are Performed



Oh! The color scheme sucks now.

Let's change the paragraph text to white and headings to yellow. We can make these additions/changes in the `<style>` tag wrapper within the `<head>` tag:

```
h1, h2, h3, h4, h5, h6 {
  color: yellow;
}
p {
  color: rgb(255, 255, 255);
  margin-left: 35px;
}
```

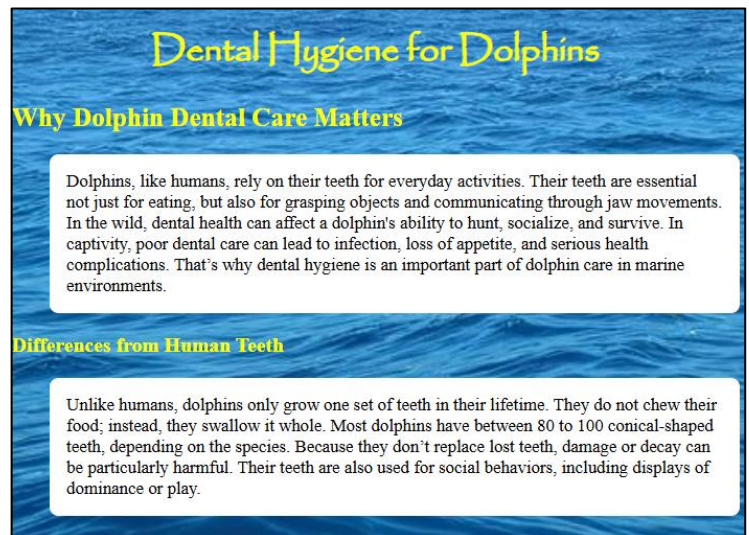
Well, it is still difficult to read. How about making a white background box for the paragraphs, and then we can put the text black again:

```
p {
  background-color: white;
  padding: 15px;
  color: rgb(0, 0, 0);
  margin-left: 35px;
  border-radius: 8px;
}
```

The padding allows us to leave some margins around the outside of the paragraph. The border-radius makes a nice rounded edge on the paragraph.

I am sure that we may have lots of artistic ideas on what to do with this page using **CSS**.

We can even try mimicking some of the **CSS** arrangements from <http://www.csszengarden.com/>



3.2 Our First Website – Content Only



When developing a website ... we have to have a clear goal in mind. In our case, the goal will be to develop a website for a fictional high-tech company called **FutureTech Corp.** that creates and sells futuristic high-tech devices for the public. So ... where do we begin?

It all begins with content ... which is the substance that a website uses to communicate its message and engage its audience. So, we will need to gather information about the company (likely from various individuals that work there, especially those involved with media relations).

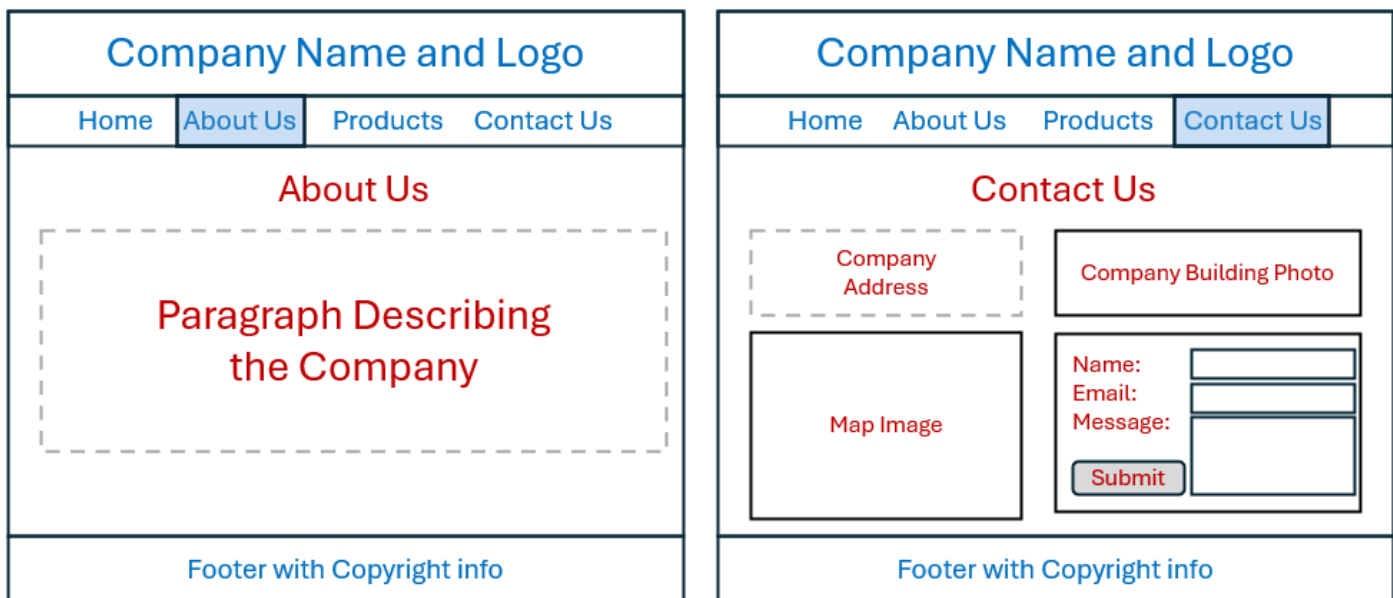
Here is useful information that we will need for our website:

- Information, specifications and pricing for all the products that will be sold on the website.
- The company address and contact information (e.g., for a “Contact Us” page).
- Information about what the company does & how it got started (e.g., for an “About Us” page).
- Images for all products, logos, media images of the facility etc..

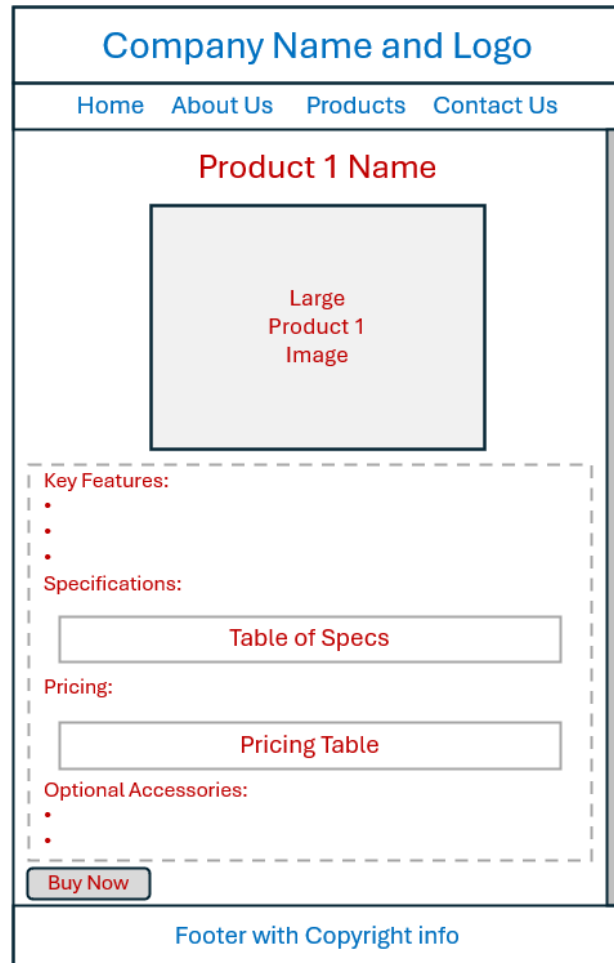
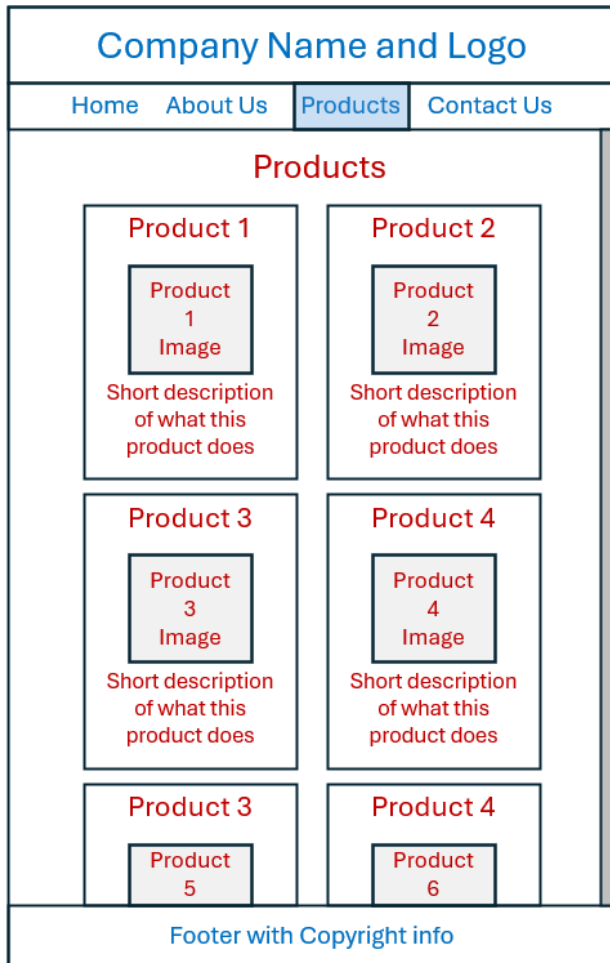
We may then need to make some decisions regarding each individual webpage that we would need to create and how to arrange things on those webpages. This is not easy. We can look around at various websites and get some ideas. For us, here is the layout for our main homepage:



Then as we click on the “**About Us**” and “**Contact Us**” links, we will go to these pages:



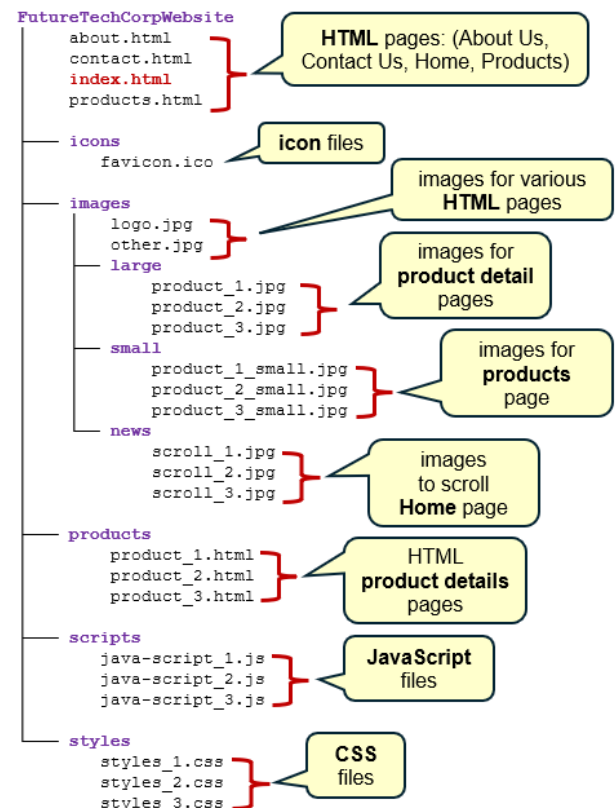
Finally, the **Products** page will be as shown below on the left and each individual **Product Details** page will be as shown on the right:



At this point, we have a rough idea of how our website will behave and we can start thinking of creating the HTML pages. But we want to stay organized ... so we will want to start preparing a proper hierarchy of folders and files.

To the right is a typical kind of directory structure for a website with pages, style files, images and script files. We will learn more about these in the course.

Our page will be arranged with a directory structure as shown here on the right. →





Now, the first step towards creating each individual page is to focus on the content. Let's do the **"About Us"** page first.

Before we start though, notice that ALL our pages have the same header and footer. So, we can make a little template that we can use for all of our pages. Let's prepare that first. We will set up our header with this [logo.jpg](#) image at the top of each page:



We will also use the following favicon called [logo-icon.png](#): 

To make our header's menu options (i.e., **Home**, **About Us**, **Products** and **Contact Us**), we can use the `<nav>` `</nav>` tags in HTML, so browsers and assistive tools recognize it being for navigation. allows us to supply a list of `<a>` `</a>` **href** links that can be clicked on ... each going to a different webpage. These navigation links are good for top menus and sidebars etc..

As for the footer, it is defined with the `<footer>` `</footer>` tag pair (for easier navigation by assistive tools again) and we will simply put a small piece of text there. So ... below is the template for ALL of our webpages. Take note of how the code uses a path that does into the **icons** and **images** folders since they are subfolders of the folder that this [about.html](#) page will be in.

We should understand all of this by now. Notice that we add `class="active"` to the **"About Us"** link. This will allow that link to be selected for that page. We would need to move that added text to the appropriate reference in each of the other pages so that the correct page is selected in the header:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>About Us</title>
  <link rel="icon" href="icons/logo-icon.png">
</head>

<body>
  <header>
    
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="about.html" class="active">About Us</a></li>
        <li><a href="products.html">Products</a></li>
        <li><a href="contact.html">Contact Us</a></li>
      </ul>
    </nav>
  </header>
```

```

<main>
  INSERT PAGE-SPECIFIC STUFF HERE
</main>

<footer>
  <p>&copy; 2025 FutureTech Corp. All rights reserved.</p>
</footer>

</body>
</html>

```

As for the page-specific information, we have to gather that from our fictional company media-relations people. We will assume that we received all that information as well as a nice image of the company headquarters (as shown here on the right called [headquarters.jpg](#)), and then we can produce the following main text for the [about.html](#) page:



```

<main>
  <section>
    <h2>About Us</h2>
    <div style="text-align: center;">
      
    </div>

    <div>
      <p> At FutureTech Corp., we don't just imagine the future, we invent it.
      Founded a decade ago in 2015 by a group of visionary engineers and futurists,
      our company has grown from a small startup into a global pioneer of
      revolutionary technologies that push the boundaries of possibility well beyond
      what exists today. From cutting-edge quantum devices to advanced AI-integrated
      systems, our products redefine what high-tech means in the 21st century.
      </p>

      <p> Over the past ten years, FutureTech Corp. has combined breakthrough research
      with impeccable design to create solutions that transform industries and
      enhance everyday life. Our dedicated team of scientists, developers, and
      creatives work tirelessly to deliver products that empower users to achieve
      the extraordinary. Milestones like the launch of our first quantum-powered
      wearable and the introduction of AI-driven neural interfaces have cemented
      our reputation as leaders in futuristic innovation.
      </p>

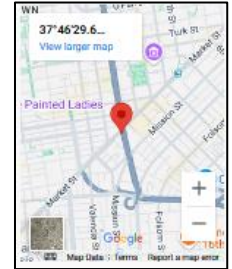
      <p> As a global leader in high-tech development, we remain committed to
      sustainability, security, and seamless integration - ensuring our innovations
      are not only advanced but responsible and accessible. Join us as we continue to
      shape tomorrow's world - today.
      </p>
    </div>
  </section>
</main>

```

Notice that we did an inline style to make sure that the image was centered on the page.

Our **Contact Us** page will have the same header and footer. The main part of the page will need to be arranged in a table. We will need the fictional company address, an image of the company building, a link to the google map and a form to be filled out.

To add a link to a google map, we just use the `<iframe>` `</iframe>` tag pair to add an *inline frame*. This tag allows us to embed another **HTML** page within our current page. We just supply the webpage address as the source and we can supply parameters that allow it to be sized and to grow) appropriately). The code is easy (since we just go to the google maps website and supply it with our company's latitude and longitude values (which we will make up) ... and then google does all the rest):



```
<iframe
  src="https://maps.google.com/maps?q=37.7749,-122.4194&z=14&output=embed"
  width="100%"
  height="300"
  style="border:0;"
  allowfullscreen=""
  loading="lazy"
  title="FutureTech Corp Map">
</iframe>
```

Then the last interesting part is the use of the form for submitting a question. We will supply for fields for name, email address and a message ... and then include a **Send Message** button. Two form fields will use the `<input>` tag (i.e., regular text field) using `required` to ensure that something is entered in them before the form can be submitted. The **name** one will have the `"text"` type while the other will have `"email"` type (which will ensure that there is at an @ symbol somewhere in the text). The message portion will require a `<textarea>` `</textarea>` tag pair. We will set it to 5 rows of text. Here is the code:

```
<form action="https://httpbin.org/post" method="POST" target="_blank">
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name" required><br><br>

  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email" required><br><br>

  <label for="message">Message:</label><br>
  <textarea id="message" name="message" rows="5" required></textarea><br><br>

  <button type="submit">Send Message</button>
</form>
```

The form will have the action set to send the data to <https://httpbin.org/post>. This is a nice text site that will accept form submissions. We can use it to ensure that the data being sent is correct. Later, we will learn for to handle these forms when we do server-side programming.

Here is the completed [contact.html](#) code for the main portion of the page. Take note of how the table is used to fill the data for two rows and two items per row:

```
<main>
  <section>
    <h2>Contact Us</h2>

    <table class="contact-table">
      <tr>
        <td>
          <h3>Our Office</h3>
          <p><strong>FutureTech Corp.</strong><br>
            4567 Hyperion Avenue<br>
            Sector 9, NeoCity, CA 94018<br>
            United States
          </p>
          <p><strong>Phone:</strong> (555) 013-2048<br>
            <strong>Email:</strong>
            <a href="mailto:support@futuretechcorp.com">support@futuretechcorp.com</a>
          </p>
        </td>
        <td></td>
      </tr>

      <tr>
        <!-- Left Column: Contact Info -->
        <td class="contact-info">
          <div class="map-container">
            <h3>Find Us</h3>
            INSERT GOOGLE MAP IFRAME STUFF (from above) HERE
          </div>
        </td>

        <!-- Right Column: Contact Form -->
        <td class="contact-form-box">
          <p>Feel free to get in touch using the form on the right or through
            the contact information below.</p>
          INSERT FORM STUFF (from above) HERE
        </td>
      </tr>
    </table>
  </section>
</main>
```

Now let's work on a **Product Details** page. We will need to make one for each product being sold online. It will be similar to the **About Us** page by having a product name, an image and then some text.

We need a product image and some text. We will discuss a **Jet Pack** product using the image here on the right.



Company Name and Logo	
Home	About Us Products Contact Us
Product 1 Name	
Large Product 1 Image	
Key Features:	
Specifications: Table of Specs	
Pricing: Pricing Table	
Optional Accessories:	
Buy Now	
Footer with Copyright info	

For the **Key Features**, **Specifications**, **Pricing** and **Optional Accessories** labels, we use an `<h3>` tag. Then we supply an unordered list of the key features and accessories, while the specs and pricing will each go into an organized table.

Finally, we conclude with a **Buy Now** button, but it doesn't really do anything. Here is the code for `product_01.html`:

```
<main>
  <section class="product-detail">
    <h2>SkyCore JetPack</h2>
    <div style="text-align: center;">
      
    </div>
    <p>
      Experience the future of personal flight with the <strong>SkyCore JetPack</strong>
      – engineered with next-gen quantum propulsion technology for unparalleled speed,
      agility, and eco-friendly performance. Designed for thrill-seekers and urban
      adventurers alike, the AeroX Quantum redefines freedom, allowing you to soar
      effortlessly above the city skyline or explore rugged terrains with ease.
    </p>
    <p>
      Built with lightweight carbon-titanium alloys and powered by a zero-emission quantum
      core, this jet pack is not only fast but also sustainable. Integrated AI navigation and
      safety systems ensure a smooth and secure flight experience, whether you're a beginner
      or a seasoned pilot.
    </p>

    <h3>Key Features:</h3>
    <ul>
      <li>Quantum Propulsion Engine: Silent, ultra-efficient engine offering up to 45 minutes
        of continuous flight at speeds reaching 120 mph.</li>
      <li>AI-Assisted Navigation: Real-time obstacle detection, automatic stabilization, and
        route optimization via onboard AI.</li>
      <li>Lightweight & Durable: Constructed with carbon-titanium alloy, weighing just 15 kg
        (33 lbs).</li>
      <li>Eco-Friendly Power Source: Zero-emission quantum core, rechargeable with standard
        power outlets or solar charging pads.</li>
      <li>Advanced Safety Systems: Includes automatic emergency landing, collision avoidance,
        and biometric access control.</li>
      <li>Modular Design: Interchangeable thrusters and customizable external panels for
        personalized style.</li>
      <li>Compact Foldability: Easily foldable wings and thrusters for convenient storage and
        transport.</li>
    </ul>

    <h3>Specifications:</h3>
    <table class="specs-table">
      <tbody>
        <tr><th>Flight Time</th><td>Up to 45 minutes</td></tr>
        <tr><th>Maximum Speed</th><td>120 mph (193 km/h)</td></tr>
        <tr><th>Weight</th><td>15 kg (33 lbs)</td></tr>
        <tr><th>Materials</th><td>Carbon-titanium alloy frame</td></tr>
        <tr><th>Power Source</th><td>Rechargeable quantum core</td></tr>
        <tr><th>Charging Time</th><td>90 minutes (standard outlet)</td></tr>
        <tr><th>Navigation System</th><td>AI-assisted with obstacle detection</td></tr>
        <tr><th>Safety Features</th><td>Emergency landing, collision avoidance, biometric
```

```

        access</td></tr>
    <tr><th>Dimensions</th><td>60 cm x 45 cm x 35 cm (folded)</td></tr>
    <tr><th>Operating Temperature</th><td>-20°C to 50°C (-4°F to 122°F)</td></tr>
    <tr><th>Connectivity</th><td>Bluetooth 6.0, WiFi 7</td></tr>
    <tr><th>Maximum Payload</th><td>120 kg (265 lbs)</td></tr>
</tbody>
</table>

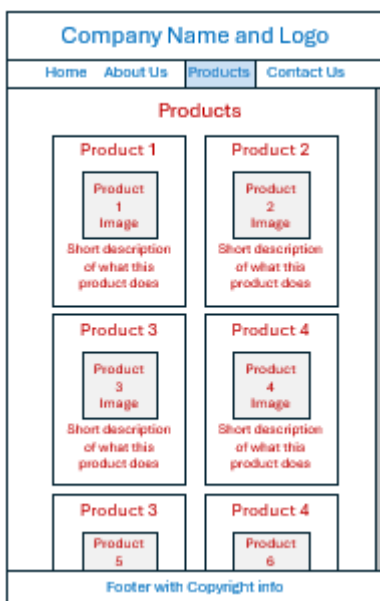
<h3>Pricing:</h3>
<table class="pricing-table">
    <thead>
        <tr><th>Package</th><th>Price (USD)</th><th>Includes</th></tr>
    </thead>
    <tbody>
        <tr><td>Standard Package</td><td>$19,999</td><td>Jet pack, charger, basic user manual</td></tr>
        <tr><td>Advanced Package</td><td>$24,999</td><td>Standard + AI flight coach app + extended warranty (3 years)</td></tr>
        <tr><td>Premium Package</td><td>$29,999</td><td>Advanced + custom design panels + on-site training</td></tr>
    </tbody>
</table>

<h3>Optional Accessories:</h3>
<ul>
    <li>Solar Charging Pad – $499</li>
    <li>Extra Modular Thrusters – $799</li>
    <li>Protective Flight Helmet with HUD – $1,199</li>
    <li>Portable Flight Case – $299</li>
</ul>

    <button>Buy Now</button>
</section>
</main>

```

Take note that we will use the larger product images on our **Product Details** pages ... so we go into the **images/large** subfolders to get the images that we need.



The **Products** page (i.e., the one that lists all products) will just use an **<h3>** heading for each product, then an image under it and then a single line of text using the **<p></p>** tag pair.

But we will make the image itself be a link to the **Product Details** page for that product. To do that, we use the **<a>** tag pair as follows:

```

<a href="products/product_01.html">
    
</a>

```

Notice that the reference will go into the **products** subfolder and then link to that product's details page. Also notice that we are now going to

use the **small** product images and we will even ensure that the **width** is set at 150 pixels for all images, for consistency. Because we put the image into the `<a>` tag, we will be able to click on the product image to get to its details page.

We will wrap the list and products in `<div></div>` tags with `class=` labels so that we can refer to them in our **CSS** file later to adjust how they will look and feel. We will make separate class labels: one for the list itself, and one for the products ... so that we can display them differently. Here is the structure of the main section of our `products.html` page with **6** items on it:

```
<main>
  <section>
    <h2>Our Products</h2>
    <div>
      <div class="product">
        INSERT <h3>, <a> AND <p> INFO for PRODUCT 1 AS DESCRIBED ABOVE .
      </div>
      <div class="product">
        INSERT <h3>, <a> AND <p> INFO for PRODUCT 2 AS DESCRIBED ABOVE .
      </div>
      <div class="product">
        INSERT <h3>, <a> AND <p> INFO for PRODUCT 3 AS DESCRIBED ABOVE .
      </div>
      <div class="product">
        INSERT <h3>, <a> AND <p> INFO for PRODUCT 4 AS DESCRIBED ABOVE .
      </div>
      <div class="product">
        INSERT <h3>, <a> AND <p> INFO for PRODUCT 5 AS DESCRIBED ABOVE .
      </div>
      <div class="product">
        INSERT <h3>, <a> AND <p> INFO for PRODUCT 6 AS DESCRIBED ABOVE .
      </div>
    </div>
  </section>
</main>
```



Finally, for the main part of the `index.html` page, there really isn't anything there except the images. All the interesting stuff will get done in the **CSS** and **JavaScript** files. In fact, for now, all we can do is show a single image. Think of it as a news item about a company lab. In fact, we will place all our images for the slider in a **news** folder under `images`:

```

```

So ... we have all our content now.



- [Home](#)
- [About Us](#)
- [Products](#)
- [Contact Us](#)

Contact Us

Our Office

FutureTech Corp.
 4567 Hyperion Avenue
 Suite B, New City, CA 94018
 United States
Phone: (555) 011-2048
Email: support@futuretechcorp.com



Find Us

37° 40'20" N 122° 47'42" W
 View map



The Pinpoint Location
 4567 Hyperion Avenue
 New City, CA 94018

Paid fees to get in touch using the forms on the right or through the contact information below:

Name:
 Email:
 Message:

© 2023 FutureTech Corp. All rights reserved.

FutureTech™

CORP.

- Home
- About Us
- Products
- Contact Us

SkyCore JetPack

Experience the future of personal flight with the **SkyCore JetPack** — engineered with next-gen quantum propulsion technology for unparalleled speed, agility, and eco-friendly performance. Designed for thrill-seekers and urban adventurers alike, the AeroX Quantum redefines freedom, allowing you to soar effortlessly above the city skyline or explore rugged terrain with ease.

Built with lightweight carbon-titanium alloys and powered by a zero-emission quantum core, this jet pack is not only fast but also sustainable. Integrated AI navigation and safety systems ensure a smooth and secure flight experience, whether you're a beginner or a seasoned pilot.

Key Features:

- **Quantum Propulsion Engine:** Silent, ultra-efficient engine offering up to 45 minutes of continuous flight at speeds reaching 120 mph.
- **AI-Assisted Navigation:** Real-time obstacle detection, automatic stabilization, and route optimization via onboard AI.
- **Lightweight & Durable:** Constructed with carbon-titanium alloy, weighing just 15 kg (33 lbs).
- **Eco-Friendly Power Source:** Zero-emission quantum core, rechargeable with standard power outlets or solar charging pads.
- **Advanced Safety Systems:** Includes automatic emergency landing, collision avoidance, and biometric access control.
- **Modular Design:** Interchangeable thrusters and customizable external panels for personalized style.
- **Compact Foldability:** Easily foldable wings and thrusters for convenient storage and transport.

Specifications:

Flight Time	Up to 45 minutes
Maximum Speed	120 mph (193 km/h)
Weight	15 kg (33 lbs)
Materials	Carbon-titanium alloy frame
Power Source	Rechargeable quantum core
Charging Time	80 minutes (standard outlet)
Navigation System	AI-assisted with obstacle detection
Safety Features	Emergency landing, collision avoidance, biometric access
Dimensions	60 cm x 45 cm x 35 cm (folded)
Operating Temperature	-20°C to 50°C (-4°F to 122°F)
Connectivity	Bluetooth 6.0, WiFi 7
Maximum Payload	120 kg (265 lbs)

Pricing:

Package	Price (USD)	Includes
Standard Package	\$19,999	Jet pack, charger, basic user manual
Advanced Package	\$24,999	Standard + AI flight coach app + extended warranty (3 years)
Premium Package	\$29,999	Advanced + custom design panels + on-site training

Optional Accessories:

- Solar Charging Pad — \$499
- Extra Modular Thrusters — \$799
- Protective Flight Helmet with HUD — \$1,199
- Portable Flight Case — \$299

[Buy Now](#)

© 2025 FutureTech Corp. All rights reserved.



3.3 Our First Website – Appearance

Now, even though we have all our necessary information the website, we still want to make it look more appealing. In the competitive business world, companies need every advantage that they can get. By using **CSS**, we will have more control over how our site will look and we will have additional features beyond basic **HTML**. So ... the next step is to arrange everything so that it looks sleek and attractive.

First, we change the `<div>` on the `index.html` page to center-align the image:

```
<div style="text-align: center;">
```

Now, let's ensure that all our pages have a nice-looking header and footer. We should add the highlighted text below to **ALL** our pages in the `<head>` tag. It tells the browser to go get the styles from the `header-footer.css` file ... which we are about to write:

```
<head>
  <meta charset="UTF-8">
  <title>About Us</title>
  <link rel="icon" href="icons/logo-icon.png">
  <link rel="stylesheet" href="styles/header-footer.css">
</head>
```

Currently, we have the header and footer shown here on the right. They are boring and perhaps “ugly”.

First, we can get the header the footer text to have the same blue background as the logo image background, using white text as well, to match the image. We will also center the text (to match the centered image).



Here is what we will put in the `header-footer.css` file:

```
/* Settings for the header tags on all pages */
header {
  background-color: #023670; /* set color to same as logo background image */
  text-align: center; /* center the text (and image) */
}

/* Settings for the footer tags on all pages */
footer {
  text-align: center; /* center the text */
  background-color: #023670; /* set color to same as logo background image */
  color: white; /* set font color to white */
  padding: 10px 0px 10px 0px; /* (top|right|bottom|left) leaves 10 pixels on top and bottom */
}
```

Now our header and footer look like banners. But there is still work to do. We need to adjust the font as well as the menu options (which are hard to see with the default link color).



We want to adjust the navigation options (i.e., menu items) to be laid out horizontally. We can start by adding this to the code (to `header-footer.css`) to adjust the unordered lists within the `<nav>` tag:

```
/* Navigation Menu Styling */
nav ul { /* "nav ul" means this is for all unordered lists within the nav grouping */
  list-style: none; /* removes the default bullets from the list */
  padding: 0; /* no inner spacing around the list */
  margin: 0; /* no outer spacing around the list */
  display: flex; /* allow items to be laid out as needed */
  justify-content: center; /* center the items */
  background-color: #003366; /* have a slightly darker background color */
}
```

That will lay the items side by side and centered (with no bullets). We can then space them out a bit by specifying a margin around the left and right of each option equal to one **em** (which is “letter M spacing of the current font”) as follows:

```
nav ul li { /* "nav ul li" = for all unordered list items within the nav grouping */
  margin: 0 1em; /* 0 margin top & bottom, 1 font size (i.e., em) spacing on left & right */
}
```

We can then be even more specific on options for the list items and the links:

```
nav ul li a { /* "nav ul li a" = for all anchors within the list items within the nav grouping */
  color: white; /* white lettering */
  text-decoration: none; /* remove the underlines on the links */
  padding: 0.5em 1em; /* slight spacing above items and a bit more on sides */
  display: inline-block; /* in-line but behave like a block */
}
```

Finally, we can get some automatic highlighting by setting the value for hovering on a list item and by highlighting the one that is active (recall that for each page, we indicated which option was active within the HTML code). We add this code to do that:

```
nav ul li a:hover,  
nav ul li a.active { /* "a:hover" and "a.active" = for hover over as well as active links */  
    background-color: #0055cc;  
    border-radius: 4px;  
}
```

Now we have a nice-looking header with navigation:



However, the font is “yucky”. As it turns out, we will want a consistent font for all our pages. So, rather than include it in the [header-footer.css](#) file, we will make another style file that can be applied to all pages ... we will call it [general-body.css](#). Add this to the `<head>` tag of [index.html](#):

```
<link rel="stylesheet" href="styles/general-body.css">
```

To begin the [general-body.css](#) code, we will add this:

```
/* General Page Styling */  
body {  
    font-family: Arial, sans-serif; /* font to be used on all page bodies */  
    margin: 0; /* leave no spacing around the outside */  
    padding: 0; /* leave no spacing around the inside */  
    background-color: #f9f9f9; /* white background for all page bodies */  
    color: #333; /* text color for all page bodies */  
    overflow-x: hidden; /* hide horizontal overflow globally */  
}
```

Fun Fact: Color **#333** (i.e., dark gray) is easier on the eyes than pure black (**#000**) on a light background, which improves readability.

The **overflow-x: hidden;** setting prevents horizontal scrolling by hiding any content that overflows the body width. It is commonly used to avoid layout shifts from wide elements or unintentional side-scrolling.

Now we have a finished header and footer that span the whole window and look wonderful:



© 2025 FutureTech. All rights reserved.

Since we want the header/footer and font to look the same on all our pages, we should add these within the **<head>** tag for all of our **.html** pages:

```
<link rel="stylesheet" href="styles/header-footer.css">
<link rel="stylesheet" href="styles/general-body.css">
```

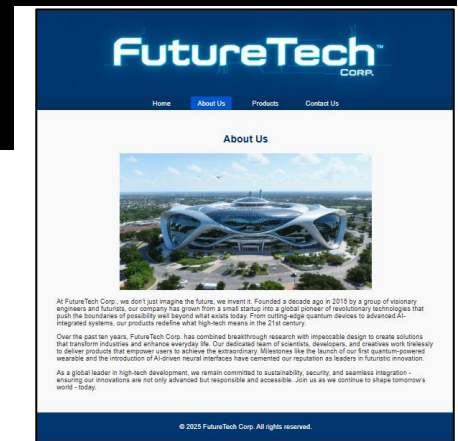
Now since all our other pages all have a title heading, we will likely want to make sure that they are centered and have a nice similar blue color by adding this to our **general-body.css** file:

```
/* Main Content Area */
h2 {
  text-align: center;
  color: #023670;
}
```

Now for each page, we can add specific styling code. We can put it in this same **general-body.css** file but indicate that it is for a specific page (and section of the page). For example, we can add the following for the **About Us** page. For this to work though, we will need to add a class called **about-page** to the **<body>** tag on the **about.html** page (i.e, **<body class="about-page">**):

```
/* For the main section of the About Us Page */
.about-page main { /* add the class to about.html page <body class="about-page"> */
  margin: 20px;
  padding: 15px;
  max-width: 900px;
  margin-left: auto;
  margin-right: auto;
}
```

This will leave a slight margin around all the text, with some spacing between the image and text and then limit the amount of stretching that the text will be allowed to make (up to **900** pixels). The page is done and looks as shown here on the right →



The **Contact Us** page needs some adjustment as well. We can now add the following additional code to the **general-body.css** file to format the **table**, **info** and **form-box** sections of that page:

```
/* For the Contact Us Page */
.contact-table { /* add the class to contact.html page <table class="contact-table"> */
  width: 100%;
  border-spacing: 2rem;
}
.contact-info { /* add the class to contact.html page <td class="contact-info"> TL & BL */
  vertical-align: top;
}
```

```
.contact-form-box { /* add the class to contact.html page <td class="contact-form-box"> */
  border: 1px solid #ccc;
  padding: 1.5rem;
  border-radius: 8px;
  background-color: #f9f9f9;
}
```

The style settings above, results in a nice rounded-rectangle border around the form box. Now, since we likely want our form fields to stretch with the window, so we add this as well:

```
/* Form Fields */
input[type="text"],
input[type="email"],
textarea {
  width: 100%;
  box-sizing: border-box;
}
```

The result of our **Contact Us** page is shown here on the right →

For our individual **Product Details** pages, we will add a separate **details.css** page that they can each use, in addition to the style files we already have ... so we will use 3 style files as follows:

```
<head>
  <meta charset="UTF-8" />
  <title>SkyCore JetPack - FutureTech Corp.</title>
  <link rel="icon" href="../icons/logo-icon.png">
  <link rel="stylesheet" href="../styles/header-footer.css">
  <link rel="stylesheet" href="../styles/general-body.css">
  <link rel="stylesheet" href="../styles/details.css">
</head>
```



The header, image and text will already look good. But we will add a margin around the page by starting off the **details.css** code with the code below (ensuring first that we add the class to the **<section>** tag of the html file (i.e., **<section class="product-detail">**):

```
/* Make a margin around the whole product box */
.product-detail { /* add the class to each product page <section class="product-detail"> */
  margin: 2rem;
}
```

Then, we will format the specifications table with left-alignment, shaded heading column and a nice border by using this (ensuring first that we add the class to the **<table>** tag of the html file (i.e., **<table class="specs-table">**):

```
/* General table styling */
.specs-table { /* add the class to each product page <table class="specs-table"> */
  width: 100%; /* table takes the whole window width */
}
```

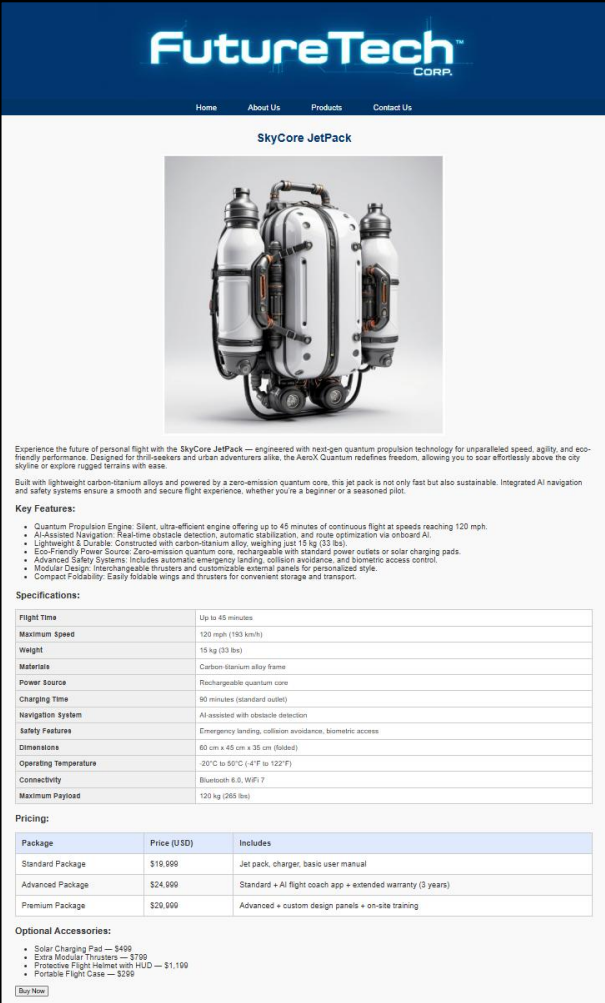


```
border-collapse: collapse; /* no outer border, just around each item */
margin-top: 1em; /* small spacing above the table */
background-color: #fff; /* white background */
}
.specs-table th,
.specs-table td {
border: 1px solid #ccc; /* solid light-gray border around each cell */
padding: 0.5em; /* small spacing around the cell text */
text-align: left; /* left-aligned */
}
.specs-table th {
background-color: #f0f0f0; /* headings have light-gray background */
font-weight: bold; /* headings have bold text */
}
.specs-table {
font-size: 0.9em; /* slightly smaller font size than the other text */
}
```

The pricing table will be similar, but will have a light blue header. We can also have zebra striping for our table (makes reading larger tables easier). Notice all of this in the code below that we add to the `details.css` page (ensuring first that we add the class to the `<table>` tag of the html file (i.e., `<table class="pricing-table">`):

```
.pricing-table { /* add class <table class="pricing-table"> */
width: 100%;
border-collapse: collapse;
margin-top: 1em;
background-color: #fff;
}
.pricing-table th,
.pricing-table td {
border: 1px solid #ccc;
padding: 0.75em;
text-align: left;
}
/* Header styling for pricing table */
.pricing-table th {
background-color: #e0eaff;
font-weight: bold;
}
/* Optional: Add zebra striping to pricing table */
.pricing-table tbody tr:nth-child(even) {
background-color: #f9f9f9;
}
```

The result is a very nice-looking product-details page as shown here on the right →



Finally, our **Products** list page needs to be adjusted to look nice. We will include a [products-style.css](#) file to the [products.html](#) page:

```
<head>
  <meta charset="UTF-8">
  <title>Our Products</title>
  <link rel="icon" href="icons/logo-icon.png">
  <link rel="stylesheet" href="styles/header-footer.css">
  <link rel="stylesheet" href="styles/general-body.css">
  <link rel="stylesheet" href="styles/products-style.css">
</head>
```

In the [products-style.css](#) file, we make similar settings for the products:

```
/* Styling for the overall products list */
.product-list { /* first, add the class to products page <div class="product-list"> */
  display: flex; /* lay products out one after another */
  flex-wrap: wrap; /* allow wraparound */
  gap: 1.5em; /* leave space between each */
  justify-content: center; /* center on each row whatever will fit */
  margin-top: 2em; /* leave a gap at the top */
  margin-bottom: 2em; /* leave a gap at the bottom */
}

/* General settings for each product */
.product { /* first, add the class to products page <div class="product"> for each product */
  background-color: white;
  border: 1px solid #ddd;
  border-radius: 8px; /* slightly rounded */
  padding: 1em; /* a little bit of padding around the inside */
  width: 250px;
  text-align: center;
  box-shadow: 0 2px 5px rgba(0,0,0,0.25); /* shadowed box around the item */
  transition: transform 0.2s ease; /* smooth transformation when hover */
}

/* Settings for the products */
.product img {
  max-width: 150px;
  height: auto;
  border-radius: 20px; /* rounded corners */
}

/* Header above the product image */
.product h3 {
  margin: 0.5em 0;
  color: #003366;
}

/* Text below the product image */
.product p {
  font-size: 0.95em;
  color: #555;
}
```

This will look nice with shadowed boxes around each item. We can even add the following to allow our boxes to lift up a little when we hover over them (which provides a nice visual effect) by adding this simple code:

```
/* Cause the product to lift up a bit when hovering over it */  
.product:hover {  
    transform: translateY(-10px);  
}
```

And VOILA ... the result of our finished page:

