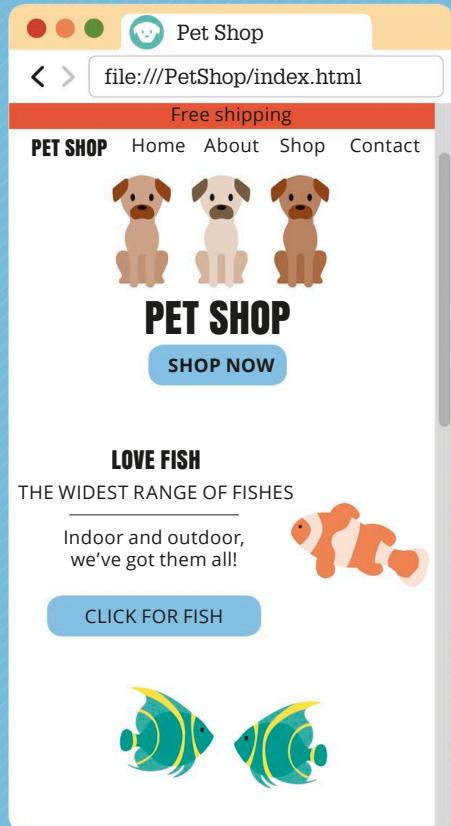


Build a web page

A modern website is built using more than one programming language. In this project, you will learn to create a basic web page—in this instance, a pet store. You will need to combine HTML, CSS, and JavaScript, but the project is made up of three parts. First, you will learn how to build an HTML framework.

How it works

The use of elements from HTML, CSS, and JavaScript will make the website structured, intuitive to navigate, and interactive.

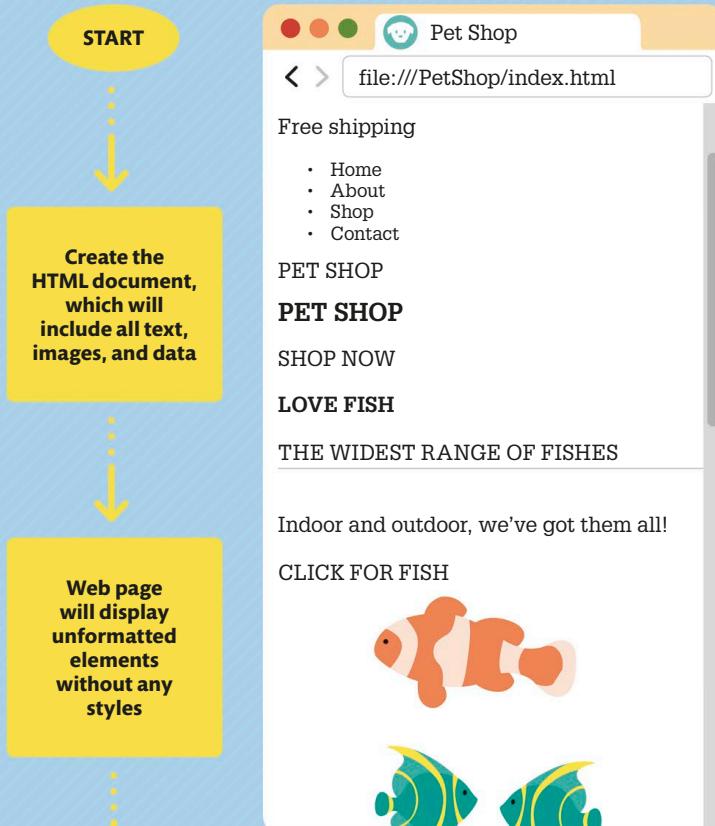


Final website

The website will be the home page of a pet supplies retail store. CSS (see pp.242–263) will add the visual styles and layout definitions, while JavaScript (see pp.288–303) will add interactive behaviors to enrich user experience on the page.

The HTML stage

You will create all the HTML elements of the web page in the first part of this project. This will include all the text, information, and data that need to appear on the website.



HTML document

At the end of the first stage, you will see a long, vertical web page. The HTML file will contain the minimum amount of code required for a valid website. This file defines the basic structure of the website.



YOU WILL LEARN

- › How to structure a page
- › How to create feature boxes
- › How to use HTML tags and attributes



Time:
2–3 hours



Lines of code:
182



Difficulty level



WHERE THIS IS USED

HTML is the backbone of all websites, even the most complicated ones. The HTML code used in this project can be reused to create different types of web pages. Any web browsers can be used to read the HTML document as a web page, including Google Chrome, Internet Explorer, and Safari.

Program requirements

You will need a few programming elements to build this website. You may also have to download and install certain components before you can start writing the code.



Development environment

The IDE (see pp.208–209) “Microsoft Visual Studio Community 2019” will be used in this project. It is a free software available for both Windows and macOS and supports various programming languages.



Browser

This project uses the Google Chrome browser to run and debug the code. You are welcome to use a different browser if you are more comfortable with it.



Images

You will need to download the images folder from www.dk.com/coding-course. A copy of this folder is required to create the home page of the website. You can also use your own images if you like.

1 Installing an IDE

To write the code for the website, you will first need a development environment. Follow the steps given below to install “Microsoft Visual Studio Community 2019” on your computer.

1.1 DOWNLOAD VISUAL STUDIO

Open a browser, go to the website mentioned below, and download the Community edition of Visual Studio. The browser will download a .dmg file to the Downloads folder on a Mac. If it does not run automatically, go to the folder and double-click the file to run it. On a Windows computer, save the installation .exe file to your hard drive and then run it.

www.visualstudio.com/downloads

1.2 INSTALL COMPONENTS

The Visual Studio Installer will display a list of languages you can program in. This project only requires languages for web development, so make sure to select the component **.Net Core** or **ASP.net and web development**. The installer will then download and install the necessary components.



Visual Studio for Mac

8.1

Create apps and games across web, and desktop with .NET, Unity, Azure, and Docker support is included by default.



.NET Core

2.1

The open source, cross-platform .NET framework and runtime.

MAC



ASP.NET and web development

✓

Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support

WINDOWS

INSTALLING AN IDE

1.3

OPEN VISUAL STUDIO

Allow any updates and then open Visual Studio. On a Mac, you can open Visual Studio by clicking its icon in the Applications folder, taskbar, or the desktop. To open it in Windows, click the icon on the startup menu, taskbar, or desktop.



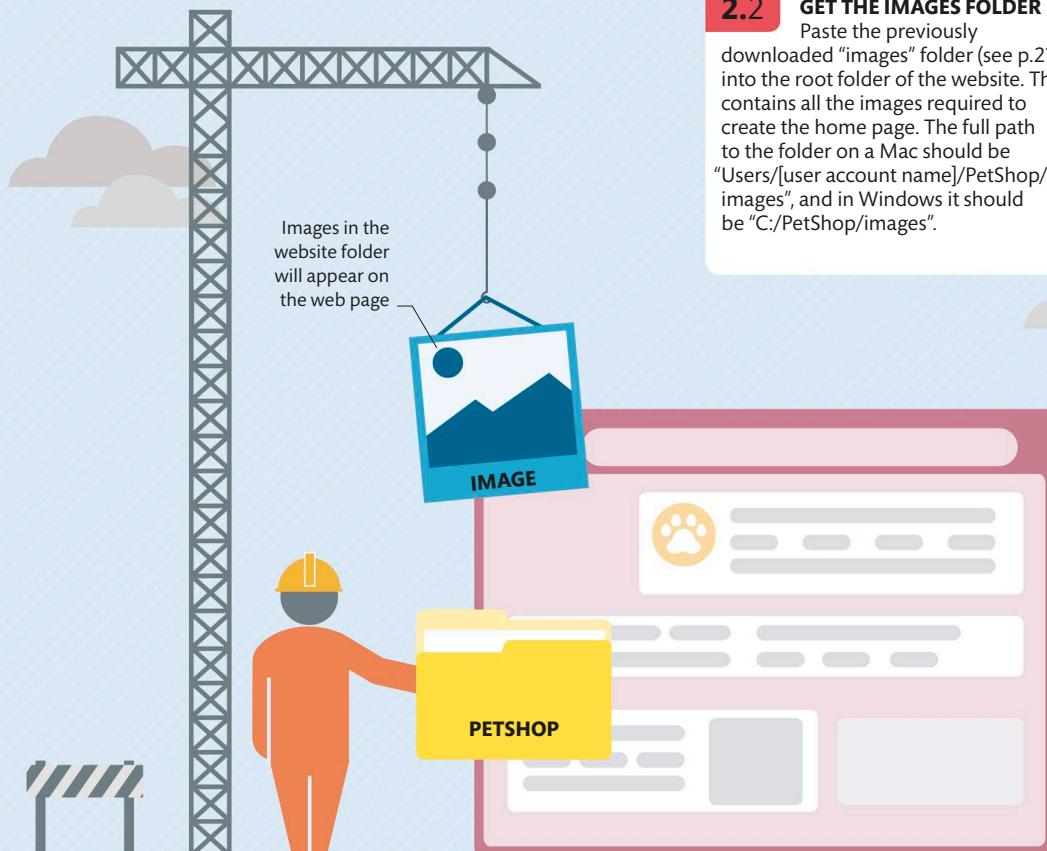
REMEMBER

HTML, JavaScript, and CSS files are just text files. Their code can be written in a simple text editor, such as Notepad orTextEdit. However, a dedicated development environment, such as Visual Studio, offers tools to improve the coding experience.

2

Getting started

After installing an IDE, it is important to get the basic elements required for coding the website. The next few steps will teach you how to create the root folder for the website, as well as the solution and index file required for writing the HTML code.



2.1

CREATE ROOT FOLDER

You will need a folder to hold all the files for the website. Use Finder to navigate to the "Users/[user account name]" folder on a Mac, or use File explorer to navigate to the "C" drive on a Windows computer. Right-click and choose New Folder to create the website folder. Name it "PetShop".



2.2

GET THE IMAGES FOLDER

Paste the previously downloaded "images" folder (see p.217) into the root folder of the website. This contains all the images required to create the home page. The full path to the folder on a Mac should be "Users/[user account name]/PetShop/images", and in Windows it should be "C:/PetShop/images".



PROJECT



2.3 OPEN A NEW PROJECT

The next step is to open a website project in Visual Studio. On a Mac, open Visual Studio, go to the File menu, and select New Solution. In the Other section, select Miscellaneous.

Click here to open a new project

Select this option to create the project



NET

Miscellaneous

Generic

Blank Solution

Workspace

Generic Project

MAC

and then Blank Solution. In Windows, open the File menu, select Open, and then select Web Site. Choose the PetShop folder that was created in the previous step.

Click here to open a new project in Windows

File

Edit

View

Website

Build

Team

New

Open

Project/Solution...

Folder...

Web Site...

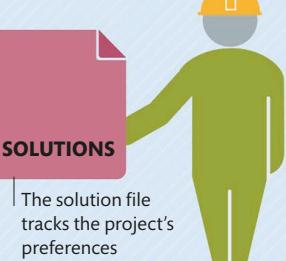
WINDOWS

2.4 CREATE A SOLUTION FILE

You now need a solution file to keep track of the project preferences. On a Mac, enter the solution name "PetShop" in the "Configure your new solution" window, then enter the location of the website folder. Click Create to save a file called "PetShop.sln" to this folder. In Windows, save the project to create a .sln file. Click the File menu and choose Save All. This will open a dialog box to save a file called "PetShop.sln". Save this file in the website folder.



LOOK FOR THE "VIEW" MENU TO DISPLAY THE "SOLUTION EXPLORER" IF IT IS NOT VISIBLE

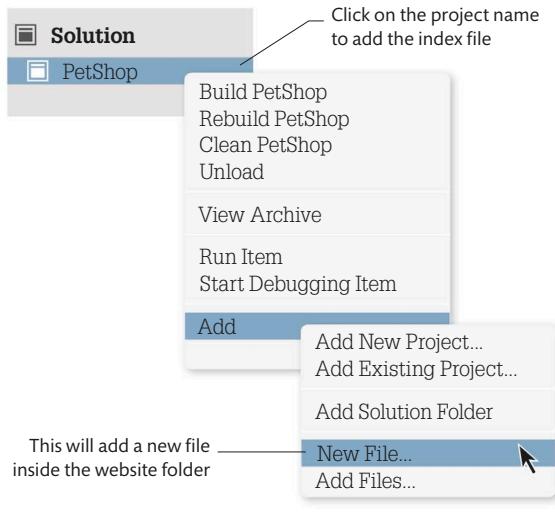


The solution file tracks the project's preferences



2.5 ADD AN INDEX FILE

Next, add an "index.html" file to the root folder of the website. In the Solution Explorer, right-click on the project name "PetShop" and select Add, then select New File on a Mac or Add New Item in Windows. Now choose HTML Page and name it "index.html". Visual Studio will add a file called "index.html" to the website folder.



GETTING STARTED

2.6

HTML PAGE

Visual Studio will create the "index.html" file with the minimum code required for a valid HTML page. If you are using another development environment, type the code shown here into the new index file.

The "charset" attribute specifies the character encoding for the HTML document. A character encoding tells the computer how to interpret binary data into real characters

This tag contains all the text, data, and images visible on the web page

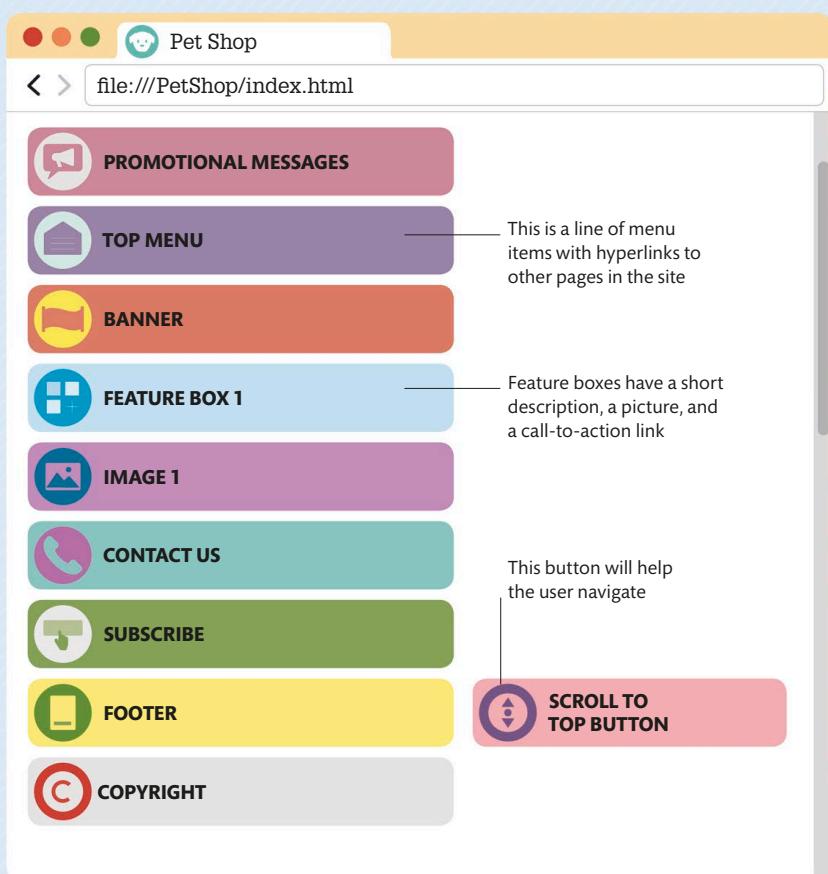
Outer tag for the HTML document

```
<!DOCTYPE html> ----- Document type declaration  
<html>  
<head>  
    <meta charset="utf-8" />  
    <title></title>  
</head>  
<body>  
    This tag is a container for the text that will appear as the page title in the browser  
</body>  
</html>
```

3

Structure the home page

In HTML, the home page will be a series of horizontal layers one above the other. The first layer will contain an animated line of promotional messages. This will be followed by the "Top Menu", a banner with a large picture, company logo, and a call-to-action button. The next element will be a feature box and then a large image. More layers repeating this pattern of alternating feature boxes and large images will be included. The layers at the bottom will contain contact details, a subscription link, hyperlinks, and a copyright notice. We will also add a "Scroll to top" button to help the user navigate back to the top of the page easily.



**3.1****ADD THE WEBSITE NAME**

Before adding the text, images, and data to the page, add the website name into the `<head>` tag by adding the page title definition into the `<title>` tag.

The `<head>` tag loads the metadata before the page is displayed

```
<head>
  <meta charset="utf-8" />
  <title>Pet Shop</title>
</head>
```

This text will appear as the tab title in the browser

3.2**ADD THE FAVICON DEFINITION**

Next, add the favicon `<link>` tag below the `<title>` tag to add the icon for the website. The "href" attribute points to the icon file in the images folder of the website.

```
<title>Pet Shop</title>
<link rel="icon" type="image/png"
      href="images/favicon.png">
```

This icon has been used in the book to split code across two lines

FAVICON

The favicon is a small image that appears in the browser tab next to the page title. It is a square image that makes it easier for a user to find the tab for the web page in a browser. The favicon can have a solid or transparent background and must have a .png or .ico file format.

**3.3****ADDING TEXT**

You can now start adding the text, data, and images inside the `<body>` tag. This will make these elements visible when the HTML document opens in the browser. To add the promotional messages, add an outer `<div>` tag, followed by the child tags to contain

the messages. All the message divs, except the first, must have a style attribute instructing the browser not to display them. For now, we will only show a single promotional message. JavaScript will be added later to the project to cycle through the promotional messages.

This tag will contain the elements that can be styled as a group

Child tags are indented under the parent "promo" `<div>` tag

The promotional messages

```
<body>
  <div id="promo" >
    <div>Free shipping</div>
    <div style="display:none;">New toys for puppies</div>
    <div style="display:none;">Buy 5 toys and save 30%</div>
    <div style="display:none;">Same day dispatch</div>
  </div>
</body>
```

STRUCTURE THE HOME PAGE

3.4 VIEW THE PAGE

You can now view this HTML page inside a browser. Save the HTML file, then in the Solution Explorer window, right-click on "index.html" and select "View in Browser". You can also open a web browser and type the URL into the address bar. On a Windows computer, the URL will be "file:///C:/PetShop/index.html". On a Mac, the URL will be "file:///Users/[user account name]/PetShop/index.html". You will now be able to view the page title in the tab name, the URL in the address bar, and the text "Free shipping" in the browser window.



GOOGLE CHROME BROWSER

DEVELOPER TOOLS

To see what is happening on an HTML page inside the browser, open its Developer tools. The Developer tools allow you to select individual HTML elements and see what CSS styles are being applied to them.

KEYBOARD SHORTCUT TO OPEN DEVELOPER TOOLS

Browser	Keyboard shortcut macOS	Keyboard shortcut Windows
Chrome	Cmd+Option+J	Ctrl+Shift+J
Opera	Cmd+Option+I	Ctrl+Shift+I
Safari	Cmd+Option+C	n/a
Internet Explorer	n/a	F12
Edge	n/a	F12

3.5 ADD THE TOP MENU SECTION

Next, it is time to add the Top Menu section. Under the "promo" div, add a new div with id ="topMenu". To make the Top Menu run across the full screen with the text inside centered on the page, surround it with a "wrap" div. This "wrap" class will be defined later in the CSS project "Styling the website"

to instruct the browser to display the Top Menu in the center of the page. Inside the "topMenu" div, add a div with class="wrap", then inside the "wrap" div, add another div with id="topLinks". This div will contain the list of hyperlinks in the Top Menu.

```
<div style="display:none;">Same day dispatch</div>
</div>
<div id="topMenu">
  <div class="wrap">
    <div id="topLinks">
      </div>
    </div>
  </div>
</div>
```

The "class" attribute describes the name of a group that the element is a member of

The "id" attribute describes the identity of the element and can be used to select the element in CSS and JavaScript

Closing tag for the "topMenu" div

Closing tag for the "wrap" div

Closing tag for the "topLinks" div



3.6 ADD THE HYPERLINK LIST

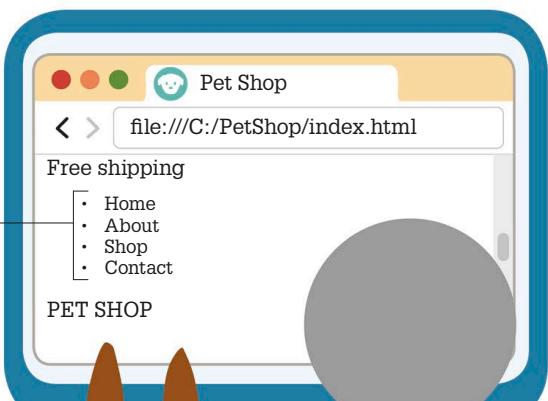
Inside the “topLinks” div, add an unordered list that will contain the actual hyperlinks in the Top Menu for the HTML pages: Home, About, Shop, and Contact. Add a small link, which appears as the company name, back to the home page. Just below the “topLinks” div, add another anchor tag to contain the name of the website. This will hyperlink to the home page.

```
...<div id="topLinks">
    <ul>————— Tag for unordered list
        <li>————— The <a> anchor tag is used
                    to describe a hyperlink
            <a href="/">Home</a>
        </li>————— The <li> tag is placed inside
                    a <ul> tag to represent each
                    individual item within a list
        <li>—————
            <a href="/">About</a>
        </li>————— The “href” attribute contains the URL that
                    points to the hyperlink destination
        <li>
            <a href="/">Shop</a>
        </li>————— The <li> tag represents
                    a list item in an ordered
                    or unordered list
        <li>
            <a href="/">Contact</a>
        </li>
    </ul>
</div>
```

PET SHOP

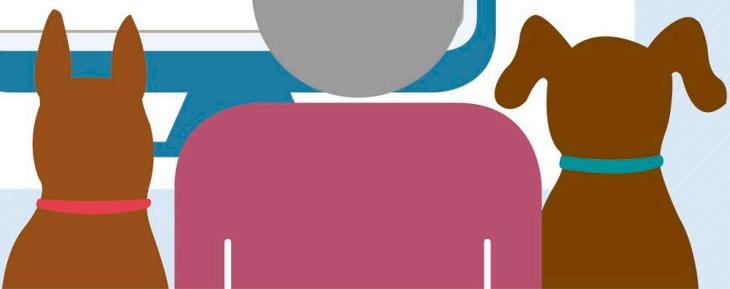
This anchor tag contains
the name of the website

The Top Menu
list appears as
hyperlinks



View the website

Save the HTML file and then refresh the page in the browser. You will see the promotional message on top, with the Top Menu list below it, followed by the hyperlink to the home page with the company logo.



STRUCTURE THE HOME PAGE

3.7 ADD THE BANNER

Next, add the banner section that will contain a large company logo and a call-to-action button to visit the shopping page. This banner image and heading should appear centered on the page, so it needs to be surrounded by a div with class="wrap".

```
</div>
<div class="wrap">
  <div id="banner">
    <h1 class="logo">PET SHOP</h1>
    <div id="action">
      <a href="/Shop">SHOP NOW</a>
    </div>
  </div>
</div>
```

The `<h1>` tag contains the company logo

The "topMenu" div closing tag

Add a div with id="banner" inside the `<div>` tag with class="wrap"

This contains the call-to-action button

Closing tag for the "banner" div

This is the hyperlink to the URL "/shop"

3.8 ADD VERTICAL SPACE

Below the "banner" div, add another div with class="spacer clear v80". The "spacer v80" classes will be used in Styling the web page (see pp.242-263) to define a standard vertical spacer between the elements. The "clear" class will be used later to instruct the browser to add the next element on a new line.

```
</div>
<div class="clear spacer v80"></div>
</div>
```

4

Feature box control

The next step is to add a feature box control to advertise the fish department. This feature box control can be reused a few more times on the page, each time alternating the side of the page that has the image and text.



Feature box structure

The left half of the feature box will contain a heading, a subheading, a text description, and a link to the category on the website. The right half will hold an image.



**4.1****USING THE CLASS ATTRIBUTE**

Add this code below the "spacer" div to define the left and right columns of the feature box for the fish department. The "class" attribute is used instead of the "id" attribute to style the HTML tag because the feature box will be used multiple times on the same page.

```
<div id="fishFeature" class="feature">
    <div class="leftColumn">
        </div>
        <div class="rightColumn">
            </div>
    </div>
```

This will contain elements for the left column

This will contain elements for the right column

4.2**DEFINE THE LEFT COLUMN ELEMENTS**

Inside the "leftColumn" div, add a div class="text" to contain the text elements. This code will add the heading, the subheading, a horizontal rule, and a description of the feature box in the left column. Add a div class="spacer" to define the vertical space between the elements, then add an anchor tag to hyperlink to the page.

This text will appear as the subheading of the feature box

This icon has been used to split code over two lines

```
<div class="leftColumn">
    <div class="text">
        This tag contains the _____ <h2>LOVE FISH</h2>
        _____ <div class="subHeading">THE WIDEST RANGE OF
        _____ FISHES</div>
        Adds a _____ <hr />
        _____ <p>Indoor and outdoor, we've
        _____ got them all!</p>
        _____ <div class="spacer v40"></div>
        _____ <a href="">CLICK FOR FISH</a>
        _____ </div>
    </div>
```

This appears as the _____
description text

This text will appear as
a hyperlink to the page

FEATURE BOX CONTROL

4.3 RIGHT COLUMN ELEMENTS

Now add this code to define the elements for the right column. In the "rightColumn" div, add an anchor tag, and inside the tag, add an with the picture for the fish department. Add a vertical spacer that can be reused throughout the website to give consistent vertical height between elements.

```
<div class="rightColumn">  
  The <a> tag describes a hyperlink  
  -----<a class="featureImage" src="/Fish">  
      
  </a>  
</div>  
</div>----- Closing tag for the "fishFeature" div  
<div class="clear spacer v20"></div>
```

The "src" attribute contains the URL to the image file

This is the name of the picture in the images folder

Adds a vertical space between elements

4.4 ADD A NEW DIV

Below the "spacer" div, add a new div with id="fishImage" and class="middleImage". This will contain the second image for the fish department and will appear under the main fish feature box. These "middleImage" containers will be used again later in this page.

Defines the middle image for the fish department

```
<div id="fishImage" class="middleImage">  
    
</div>
```

This points to the image file location

```
<div class="spacer v80"></div>
```

Adds another vertical space after the middle image

</div>----- Closing tag for the "wrap" div



5 Advertising the dog category

It is now time to add a second category to the website to advertise the dog department. This feature box will appear below the fish department and will have the image on the left and all the text elements on the right.



5.1 DOG FEATURE

To create the dog feature control box, add a div beneath the "spacer" div. Inside this new "dogFeature" div, add the "leftColumn" and "rightColumn" divs with all the text and image elements required for the dog department.

File name of the dog image
in the images folder

```
<div id="dogFeature" class="feature">
    <div class="leftColumn">
        <a class="featureImage">
            
        </a>
    </div>
    <div class="rightColumn">
        <div class="text">
            Heading for the _____ <h2>HAPPY DOGS</h2>
            dog feature box
            <div class="subHeading">EVERYTHING YOUR DOG
            NEEDS</div>
            <hr />
            <p>Make sure your pooch eats well and feels good
            with our range of doggie treats.</p>
            <div class="spacer v40"></div>
            <a href="">CLICK FOR DOGS</a>
        </div>
    </div>
</div>
```

The "href" attribute contains
the URL that points to the
hyperlink's destination

This text will appear
as a hyperlink for the
dog department

Description text for
the dog department

This text will appear
as the subheading

ADVERTISING THE DOG CATEGORY

5.2 MIDDLE IMAGE

Below the "dogFeature" div, add another div with class="clear" to start a new line, then add the second image for the dog feature box. Next, add another vertical spacer under the image.

```
<div class="clear"></div>
<div id="dogImage" class="middleImage">
    
</div>
<div class="spacer v80"></div>
```

Adds a vertical space below the middle image

File name for the middle image

6 Advertising the bird category

The next feature box to be included is for the bird department. Similar to the fish feature box, this department will have the text elements on the left and the image on the right. The bird category will appear below the dog category on the website.



6.1 BIRD FEATURE

Type the following lines of code below the "spacer" div to add another feature box to advertise the bird department. This will include the "leftColumn" and "rightColumn" text and image elements for this category.

Defines the feature box control for the bird department

Subheading for the bird department

```
<div id="birdFeature" class="feature">
    <div class="leftColumn">
        <div class="text">
            <h2>BIRDY NUM NUMS</h2>
            <div class="subHeading">KEEP YOUR BIRDS<br/>CHIPPER</div>
        </div>
    </div>
    <div class="rightColumn">
        <img alt="Birds in cages and a birdhouse" />
    </div>
</div>
```

This is the heading for the bird department





```

Description for the bird department


Yummy snacks and feeders for every kind of bird.


<div class="spacer v40"></div>
<a href="">CLICK FOR BIRDS</a>
</div>
</div>
<div class="rightColumn">
    <a class="featureImage" src="/Bird">
        
    </a>
</div>
</div>

```

Closing tag for the right column

File name for the bird image in the feature box

Hyperlink for the bird department

6.2 ADD ANOTHER IMAGE

Now add another "clear" div to start on a new line. Next, add the middle image for the bird department. Inside the "birdImage" div, add the `` tag with the URL to the image.

Closing tag for the "birdFeature" div

```

</div>
<div class="clear"></div>
<div id="birdImage" class="middleImage">
    
</div>

```

File name of the middle image



The bird image is picked up from the main images folder and added to the website



SCROLL TO THE TOP

7 SCROLL TO THE TOP

Add a button to allow the user to scroll back to the top of the page. Create a "scrollToTop" div and then add a `` tag inside it with an HTML entity (see p.233) indicating the upward arrow. The "title" attribute adds a

"tooltip" to the button so that when a user hovers their mouse over the button, a label will appear on top saying "Scroll to top". Another vertical spacer is added below the button.

```
</div> ----- Closing tag for the "birdImage" div
```

```
<div id="scrollToTop" title="Scroll to top">-----
```

```
    <span>&uarr;</span>-----
```

```
    </div>
```

```
<div class="clear spacer v40"></div>-----
```

Defines the scroll to top button

The `` tag contains the HTML entity indicating the upward arrow

Adds a vertical space below the scroll to top button

8 ADD A CONTACT SECTION

Add the contact section for the website immediately below the "spacer" div from the previous step. You can reuse the feature box controls that were used to split the page into a left and right column. In the left column, we will add the address and other contact details.



Defines the contact section

```
<div id="contactUs" class="feature">
```

```
    <div class="leftColumn">-----
```

```
        <div class="text">
```

```
            <h2>CONTACT US</h2>-----
```

TELEPHONE

EMAIL

ADDRESS

Defines the elements in the left column of the feature box

Header for the section

```
        <hr />
```

The `<p>` tag makes the information appear as a paragraph

```
        <p>TEL : 012-345-6789
```

```
    </p>
```

```
    <p>
```

```
        EMAIL : <a href="mailto:INFO@PETSHOP.COM" class="emailLink">INFO@PETSHOP.COM</a>
```

```
    </p>
```

This will appear as the email link to the website



<p>

PET SHOP
—
1450 Broadway

New York, NY

10018

The
 tag tells
the browser to start
a new line

This text will appear _____
as the address of
the Pet Shop

</p>

</div>

</div>

<div class="rightColumn">

</div>

</div>

Defines the right
column of the feature box

8.1 ADD A MAP TO THE CONTACT SECTION

In the right column, you can now embed a map to show the location of the Pet Shop. Inside the "rightColumn" div, add an <iframe> tag with the "src" attribute set to the URL of the map on Google Maps. The <iframe> tag is used to insert content from another web page onto your page.

Link to the map
in Google Maps

```
<div class="rightColumn">
    <iframe src="https://www.google.com/maps/embed?pb=1&1m1&1m12&1m3&1d3022.3334138729488&2d-3.98854988459377&3d40.75469097932727&2m3&1f0&2f0&3f0&3m2&1i1024&2i768&4f13.1&3m3&1m2&1s0x89c259ab2419ed09%3A0xb9f7694124a811f0&2s1450+Broadway%2C+New+York%2C+NY+10018%2C+USA!5e0!3m2!1sen!2sin!4v1560924564028!5m2!1sen!2sin" frameborder="0" style="border:0" allowfullscreen class="contactMap"></iframe>
</div> ----- Closing tag for the "contactUs" div
</div> ----- Closing tag for the "wrap" div
<div class="clear_spacer v80"></div> ----- Add this line to include a
vertical space after the map
```

ADD THE SUBSCRIBE SECTION

9 ADD THE SUBSCRIBE SECTION

Next, add the "Subscribe" section for the website. Below the "spacer" div, add the "subscribe" div to make this section run across the full screen. Inside the "subscribe" div, add a header, a form with action="/subscribe", and method="post". The "action"

and "method" attributes define where the form gets sent when the user clicks the "submit" button. Inside the <form> tag, add a text input field to allow the user to enter an email address and a button that says "Join Now".

```
<div id="subscribe">
    <h2>SUBSCRIBE TO OUR MAILING LIST</h2>
    <form action="/subscribe" method="post">
        <input name="email" type="text" placeholder="Enter
            your email address" />
        <input type="submit" value="Join Now" />
    </form>
</div>
```

Header for the
Subscribe section

Text for the button

10 ADD THE FOOTER

Let us now add the "footer" section after the "subscribe" </div> closing tag. This will contain the unordered list of footer hyperlinks for the website.

The "href" attribute describes
the URL of the page to link to

This will appear as the
first footer hyperlink

```
<div id="footer">
    <ul>
        <li>
            <a href="/storeFinder">Store Finder</a>
        </li>
        <li>
            <a href="/shipping">Shipping</a>
        </li>
        <li>
            <a href="/FAQ">FAQ</a>
        </li>
    </ul>
```

The second
hyperlink

The tag is a block element used
to designate an unordered list



11 ADD THE COPYRIGHT NOTICE

Add a copyright notice at the bottom of the page. This will contain the copyright message and the company logo. Notice that in the code below, the company logo is contained in a `` tag so that it can be styled later in Styling the web page (see pp.242–263). An HTML entity has also been used for the copyright symbol. You have now created the basic framework for your web page. Additional pages can also be created to build a fully functioning website (see p.303).

```
<div id="copyright">
    <div>&copy; 2020 <span class="logo">PET SHOP</span>
    </div>
</div>
</body>
```

Text for the
copyright message

HTML entity for the
copyright symbol

The company logo



ENTITIES

Some characters are not allowed in HTML because they are reserved by HTML, CSS, or JavaScript. So when you want these restricted characters to appear onscreen, they must be coded with an HTML entity so that they will render correctly in the browser.

This HTML entity will
display a © symbol

```
<p>
    &copy; DK Books 2020
</p>
```

Common entities

Here is a list of some of the most commonly used HTML entities. See <https://dev.w3.org/html5/html-author/charref> for a full list.

COMMON ENTITIES		
Symbol	Meaning	HTML entity
“ ”	Quotation marks	"
	Whitespace/spacebar	
&	Ampersand	&
%	Percent	&percent;
\$	Dollar	$
©	Copyright	©
'	Apostrophe	'

Cascading Style Sheets

Cascading Style Sheets (CSS) define how the contents of an HTML file should appear in a web browser. It allows for the design of a website to be easily updated by making changes to the CSS style definitions.

Why CSS?

Website styling, until 1996, was done inside individual HTML tags, making code extremely long and cluttered. CSS simplified this by separating style from content. A CSS file contains a list of rules that provide an easy

way to define the style of single elements and to share the same styles across multiple elements in an HTML document. The client web browser reads the CSS files and applies the style definition to each element in the HTML document.

Adding CSS styles to an HTML document

A CSS style can be defined in three places in an HTML document: in an external CSS file, in a `<style>` tag inside the HTML file, and in a “style” attribute inside an HTML tag.

In a `<style>` tag

An HTML file can contain CSS definitions inside a `<style>` tag, which is usually placed inside the `<head>` section. These CSS definitions do not apply to other pages on the website.

```
<head>      All <h1> headers  
            on this page will  
            be blue  
  
            <style>  
                h1 {  
                    color: blue;  
                }  
  
                p {  
                    color: red;  
                }  
  
            </style>  
</head>
```

Element to
be styled

All paragraph
text on this
page will
be red

External CSS file

When CSS style definitions are contained in a separate CSS file, the style definitions can be shared by all the pages in a website. Use a `<link>` tag to reference the sheet in an HTML document.

```
<head>      The <link> tag instructs the browser to  
            import CSS styles from an external file  
  
            <link rel="stylesheet" type="text/css"  
                  href="styles.css">——— Link to the  
            style sheet  
</head>
```

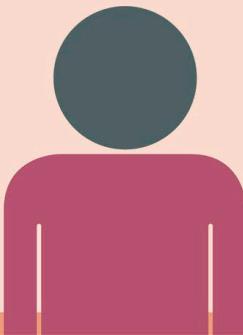


Inline CSS

CSS style definitions can be added inline as an attribute to an HTML tag. These inline style attributes will override any styles that are set globally in an external CSS file or in a `<style>` tag.

```
<p style="color: red;">Hello world!</p>
```

Only this paragraph
text will be red





Style options

CSS can define various aspects of an element's appearance onscreen, including its placement, font, color, border style, and special effects such as animation. CSS contains instructions that tell the browser how to render an HTML element onscreen. In order to work on all browsers, CSS expects precise names for the properties and values. For example, to make an HTML element invisible, the "display" property will have the value "none".

CSS STYLE OPTIONS	
CSS code	Output
<code>display:block;</code>	The element will appear as a block element.
<code>display:inline;</code>	The element will appear as an inline element.
<code>display:none;</code>	The element will not appear on the screen.
<code>font-family: "Times New Roman", serif; font-weight: bold; color: red;</code>	Specifies the font settings and font color.
<code>padding: 10px 12px 15px 30px; margin: 40px;</code>	Specifies the spacing settings (see p.245).
<code>background-color: white;</code>	Sets the background color.

How CSS works

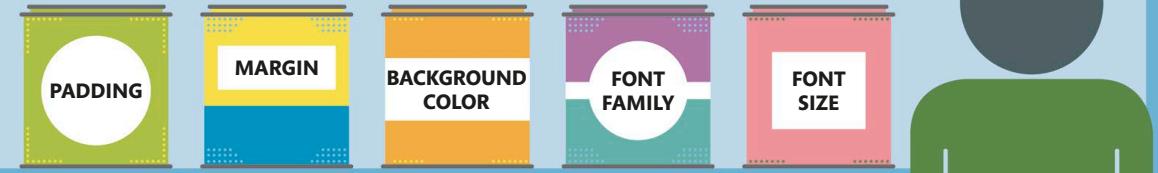
CSS works by selecting a set of HTML elements and then adding styles to all the elements in the set. Every CSS instruction consists of two parts: the selector and the style definition. The selector tells the browser which elements are to be included in the instruction, while the style definition specifies how to display the elements included in the selector.

CSS instructions

A CSS instruction contains a property and a value that defines how that property should be styled. These instructions are grouped together in a style definition. In this example, the selector is "body". The style definition is the set of style instructions contained in the curly brackets. Each style instruction has a semicolon at the end.

```
body {
    padding: 20px; — The "padding" property
    margin: 0; — has the value "20px"
    background-color: gray;
    font-family: "Open Sans", sans-serif;
    font-size: 16px;
}
```

Specifies the font name and font type



CSS selectors

A CSS selector tells the browser which HTML elements are to be styled. An element must satisfy the selection criteria in order to have the style applied to it. Selectors can target either a single element or a group of elements.

Basic selectors

A CSS selector is a pattern used to identify which HTML elements qualify to have the style applied to them. CSS selectors allow programmers to target specific HTML elements with style sheets. There are three basic selectors in CSS: the element selector, id selector, and class selector.



Element selector

This is used to target all the HTML elements of a certain type. For example, all elements inside the `<p>` tag will be colored red.

Selects all paragraph elements

`<p>This text to be in red</p>`

HTML

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

CSS

This text will implement the style definition applied to the `<p>` tag



Class selector

The dot (.) prefix indicates a class selector. Multiple elements in an HTML document can share the same class. A class selector is used to target a group of HTML elements that all have a certain "class" attribute value. For example, all elements that have the class "roundedCorners" will be styled.

`<div class="roundedCorners"></div>`

HTML

This div will implement the "roundedCorner" style definition

```
.roundedCorners{  
    border-radius: 20px;  
}
```

Selects all elements with the class "roundedCorners"

CSS





GROUPING SELECTORS

If multiple elements are to have the same style, it is not necessary to define them separately. Selectors can be grouped together using a comma. All grouped selectors will have the same style applied to them.

```
h1, h2, h3{
    font-size: 24px;
}
```

<h1>, <h2>, and <h3> tags are grouped and will all have a font size of 24px

Id selector

This selector is used to target a single HTML element with a specified "id" attribute value and is indicated with hash (#) prefix. In an HTML document, an id should be applied to only one element per page.

```
<div id="header"></div>
```

HTML

This div will implement the "#header" style definition

```
#header{
    text-align: center;
}
```

CSS



Selects the single element with the id "header"

Complex selectors

Selectors can also be combined to provide more specific definitions based on the relationships between the elements. You can combine the id, class, or tag type into a complex selector definition.

Child selector

This selector includes all the elements that are children of a particular element (see p.210). Use the greater than (>) symbol between the elements to indicate this selector. For example, `div > p`



Descendant selector

Indicated by a space, this defines all elements that are descendants of a particular element. This is similar to the child selector but will include children of the child element. For example, `div p`



General sibling selector

Defines all the elements that are siblings of a particular element. They will all have the same parent element. Use the tilde (~) symbol to indicate this selector. For example, `p ~ div`



Next sibling selector

Defines all the elements that are siblings of and follow on after a particular element. These selectors are indicated by a plus (+) symbol between them. For example, `div + p`



Multiple classes selector

Defines an element that must contain all the classes in the selector. The absence of a space between the class names indicates that all classes must be present. For example, `.roundedCorners.featureBox`



Combine id and class selector

This defines an element that must contain the id and all supplied classes. The absence of a space between the id and class name indicates that both must be present. For example, `#mainContent.minHeight`

CSS styling

CSS style definitions are used to set the background color, font size, font family, borders, and other elements in a web page. Styles are said to cascade, as they are inherited by child elements from the parent elements (the elements they are contained in).

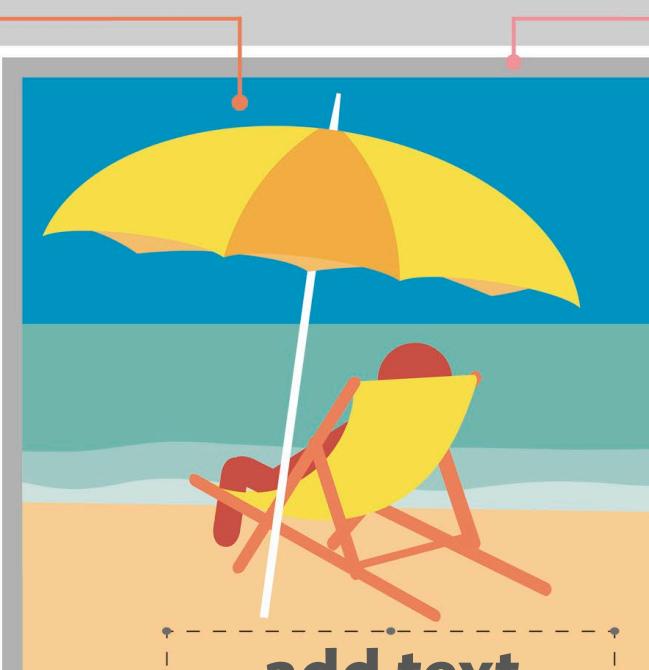
Styling color

CSS allows programmers to define the color of elements on a website, including background colors, borders, and text. The most common colors can be set with a text value of the color name—for example, white, red, or blue. All modern browsers support 140 HTML color names. Any other color value can be described in Hex, RGB, or RGBA format.

HTML COLOR CODES	
Format	Values for the color blue
Text	color : blue;
Hex full	color : #0000ff;
Hex shorthand	color : #00f;
RGB	color : rgb(0, 0, 255);
RGBA	color : rgba(0, 0, 255, 1);

`color:rgba(0,0,255, 0.5);`

RGBA format's alpha channel parameter describes the transparency of a color



add text

Font size options

In CSS, font size can be defined in several different ways.

- **Pixels:** This defines the size of the font in pixels. The number is followed by the letters "px".
- **Size:** Keywords such as "large" or "small" are used to define the size of the font.
- **Relative size:** Defines a font size relative to the parent element's font size. Uses keywords such as "larger".

- **Percentage:** Defines size relative to the parent element's font size. For example, "200%" implies twice the parent font's size.
- **Em:** This method is also relative to the parent element's font size. For example, $2\text{em} = 2 \times \text{parent font-size} = 200\%$



CSS HELPS MAINTAIN THE DESIGN ACROSS A WEBSITE

The diagram illustrates a website's visual hierarchy. On the left, a vertical sidebar contains icons for navigation (arrow up, arrow down), styling (brush, paint bucket), text (T), borders (square), circles (circle), and square borders (square). Below this is a section titled "Animation" with a play button icon. To the right of the sidebar is a main content area. At the top of the content area is a header with a square icon and the text "Border styling". The main content of this section describes CSS border properties and provides examples of CSS code. Below this is another section titled "Animation" with a play button icon, which describes the "transition" instruction.

Border styling

The CSS border properties allow programmers to specify the style, width, and color of an element's border. These border styles can be defined in several different ways.

- Define border settings in one line
`border: 1px solid black;`
- Define border settings in separate lines
`border-width: 1px;
border-style: solid;
border-color: black;`
- Define different vertical and horizontal borders
`border-width: 1px 0px;`
- Define different width for each side of the border
`border-width: 1px 0px
3px 2px;`

Animation

The “transition” instruction in CSS allows programmers to create simple animations in modern browsers, such as a button changing size or color when a mouse hovers over it. To do this, the property to be animated and the duration of the animation should be specified in the CSS file (see p.250).

Cascading styles

An HTML element can have multiple styles applied to it. Many HTML tags also inherit properties from the parent tags that contain them. The browser determines which style to apply based on the rules mentioned below.

Origin

Browsers have built-in default styles that they apply to HTML tags. These are known as user-agent styles. However, styles defined by the programmer, called author styles, will override these user-agent styles.

Importance

A style instruction that is marked with the “!important” declaration will be given priority over other instructions. It will always be applied to an element, regardless of the placement of the instruction in the CSS hierarchy.

Specificity

A style that has a more detailed selector will be applied before a style that is less specific. This means that the greater the number of elements in the selector, the higher the priority that the style will receive.

Instruction order

A style that is defined earlier in a CSS file will be overridden by a style that is defined later, and a style that is defined in the CSS file will be overridden by the styles defined inline by a “style” attribute in an HTML tag.

IMPORTANT

The !important declaration is an easy way of instructing the browser to prioritize a style definition, but it should only be used as a last resort. It is preferable to make the selector more specific by including additional classes or id values in the selection criteria. CSS will apply the definition with the most specific selector, avoiding the need to use the !important declaration.

Responsive layouts

A responsive website has a design that can adapt to display correctly on any size screen, from desktop monitors and laptops to smartphones and tablets. This is achieved by a clever combination of HTML, CSS, and JavaScript.

Viewport

The “viewport” declaration tells a browser that a website has a responsive layout. The declaration is placed alongside other metadata. The “content” attribute instructs the browser to set the page width the same as the screen width. It also sets the initial zoom level. These meta instructions allow the page elements to adjust to the maximum width of the screen and improves the user experience by displaying the correct styling and layout for any screen size. Without these instructions, the browser will zoom out to show the whole page rather than allowing the page elements to reshape to the width of the screen.



MULTIPLE CSS FILES CAN
BE LINKED TO AN HTML
DOCUMENT TO MAKE A
RESPONSIVE WEB PAGE

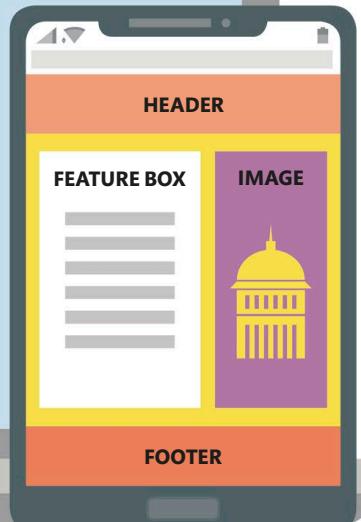
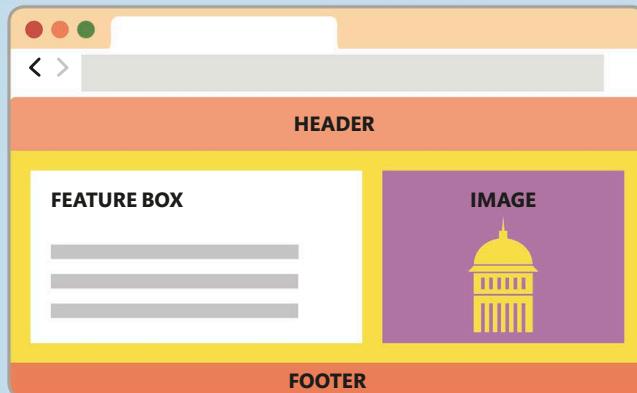
```
<head>
  <meta name="viewport" content="width=device-width,
  initial-scale=1.0">
</head>
```

The “viewport” declaration is placed inside the <head> tag

The “content” attribute sets the page width and the initial zoom level to 1

Flexible layouts

Including the “viewport” declaration in a responsive design allows the page elements to adjust to fit the screen size.





Why responsive?

When the Web was first created, almost all users viewed websites on a desktop monitor. Early websites were programmed to be viewed at a fixed width of 800px. This width was gradually increased over the years as the average user's screen size increased. The arrival of browsers on smartphones

with narrow screens forced programmers to maintain multiple versions of their sites, each designed to display correctly on a different size of screen. Today, there are many different device sizes that can display web pages. The solution to this is to have a single flexible layout that can adjust to any screen size.

Media query

Media queries are used to switch between different layout styles, depending on the width of the page. This is the primary way to create responsive web pages that can scale to fit correctly on any screen size. For example, in the code below,

the background color of elements in the "specials" class will change depending on the width of the screen. The default background color is red. If the screen is more than 993px wide, then the background color will be blue.

```
.specials {  
    background-color: blue;  
}  
  
@media screen and (min-width: 993px) {  
    .specials {  
        background-color: red;  
    }  
}
```

Flexible styles

Media queries fine-tune the CSS styles so that each element looks its best on any screen size.

