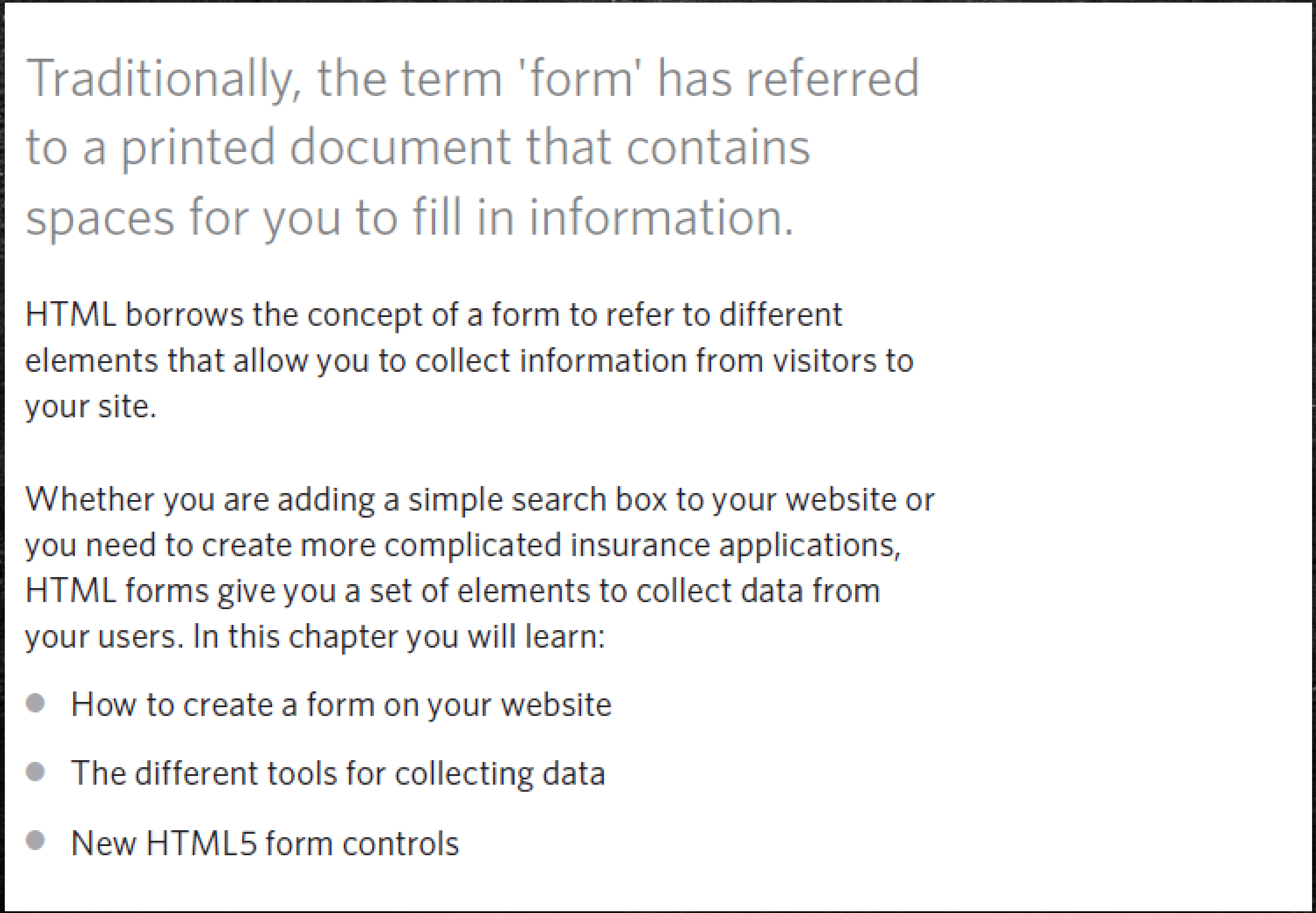
Module 4- Forms, then CSS



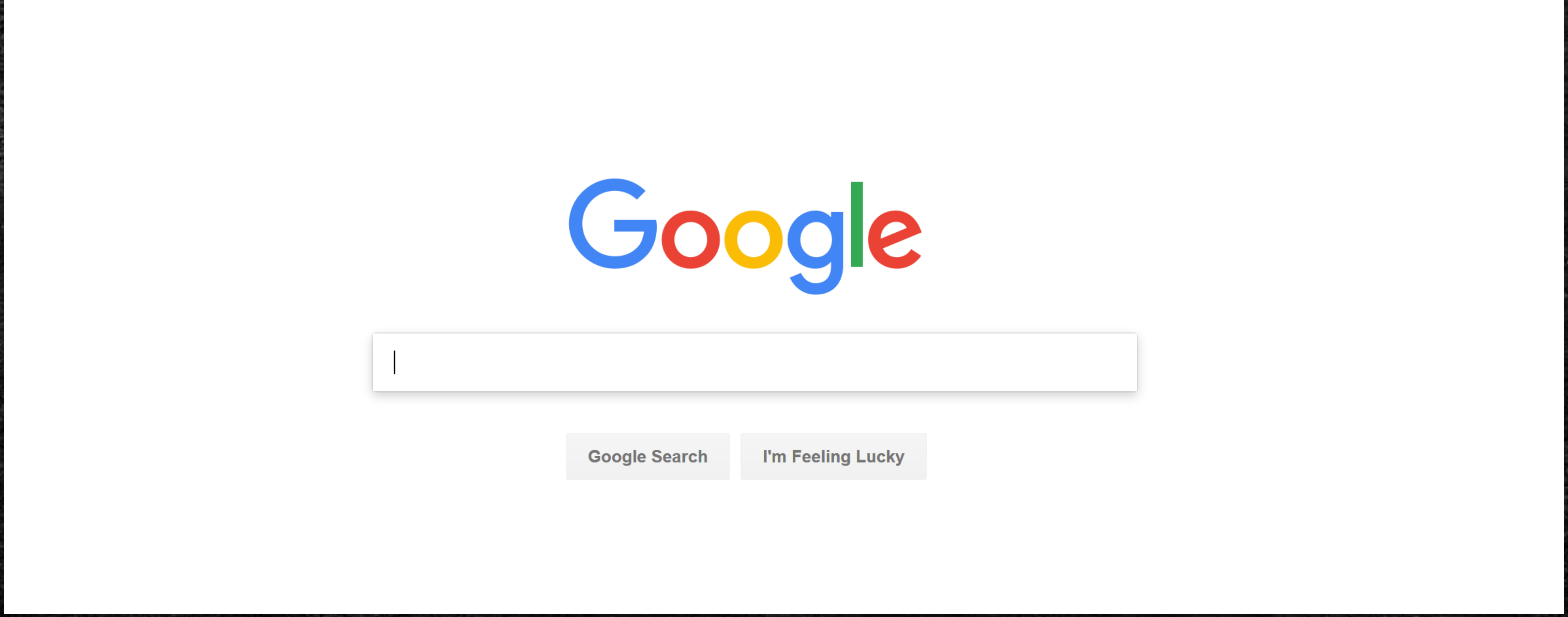
Traditionally, the term 'form' has referred to a printed document that contains spaces for you to fill in information.

HTML borrows the concept of a form to refer to different elements that allow you to collect information from visitors to your site.

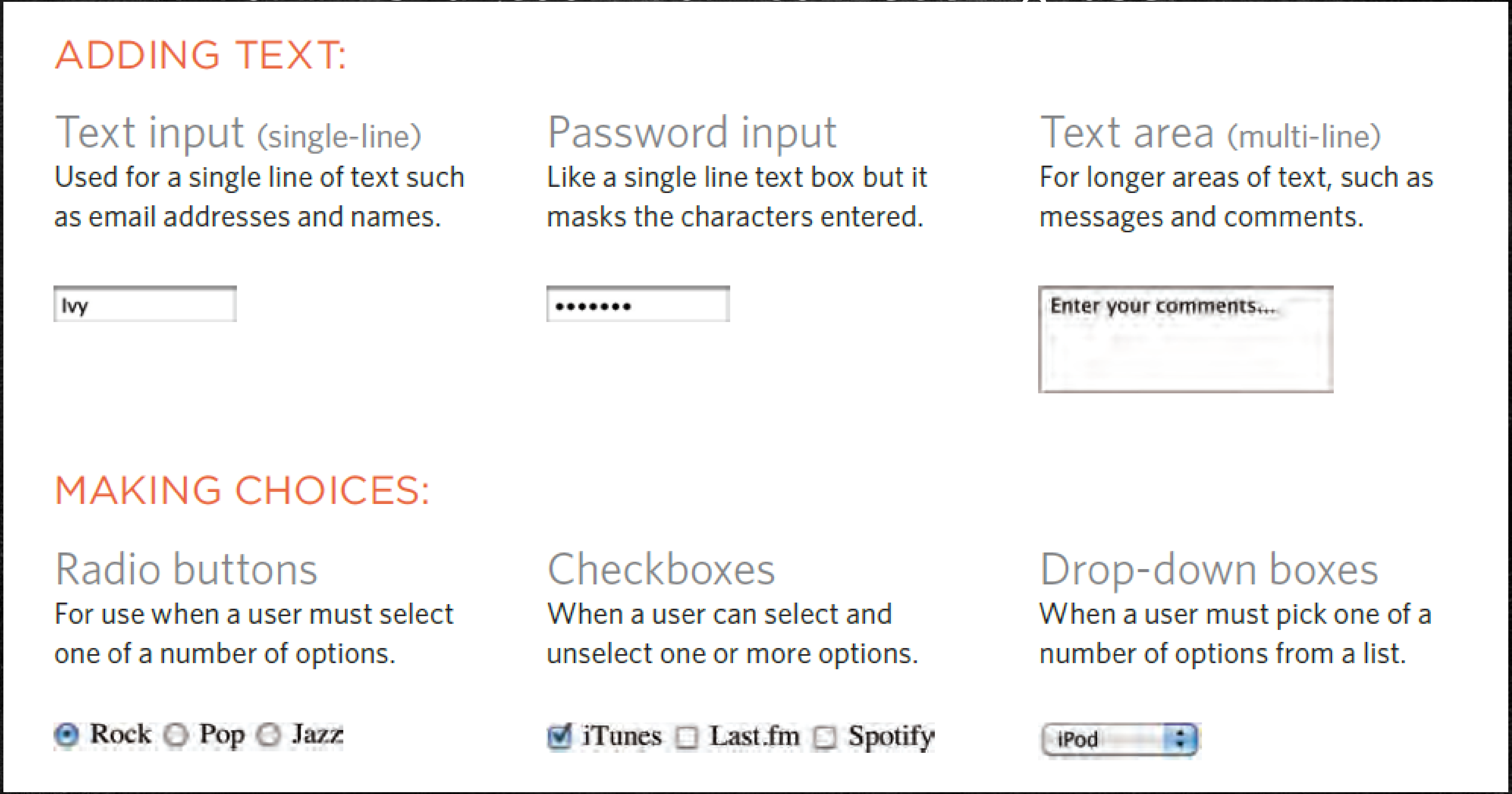
Whether you are adding a simple search box to your website or you need to create more complicated insurance applications, HTML forms give you a set of elements to collect data from your users. In this chapter you will learn:

- How to create a form on your website
- The different tools for collecting data
- New HTML5 form controls

The Most Popular Form on the Internet



A form is a tool for collecting user



ADDING TEXT:

Text input (single-line)

Used for a single line of text such as email addresses and names.

Ivy

Password input

Like a single line text box but it masks the characters entered.

••••••••

Text area (multi-line)

For longer areas of text, such as messages and comments.

Enter your comments...

MAKING CHOICES:

Radio buttons

For use when a user must select one of a number of options.

● Rock ○ Pop ○ Jazz

Checkboxes

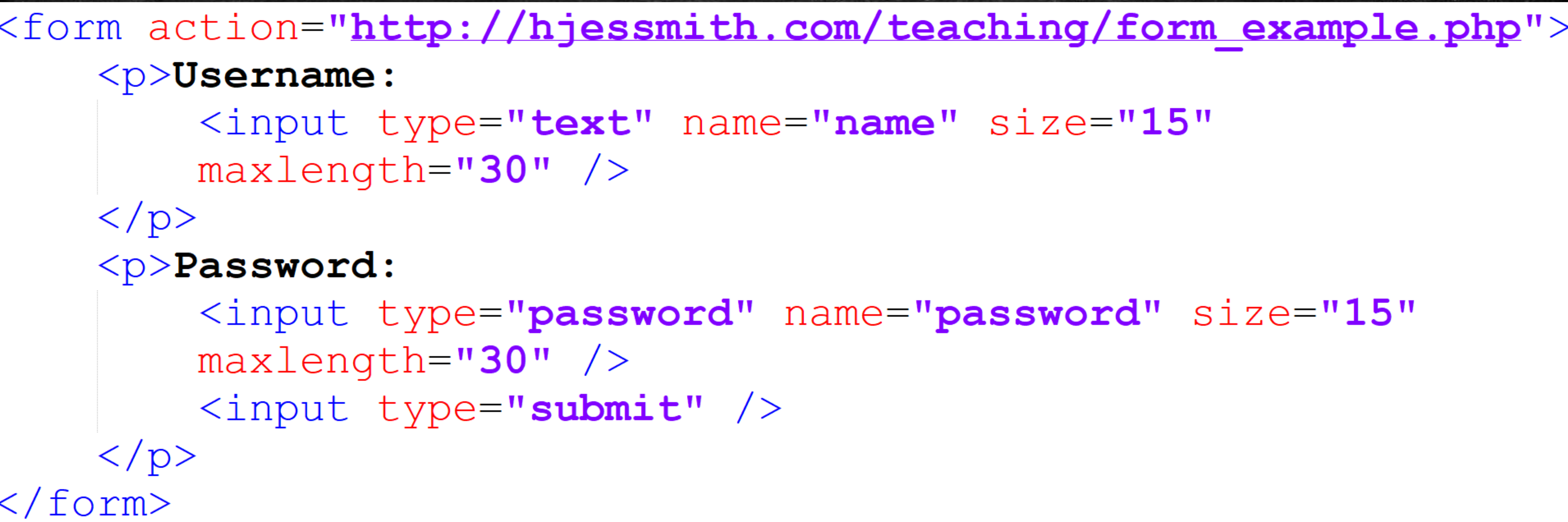
When a user can select and unselect one or more options.

☑ iTunes □ Last.fm □ Spotify

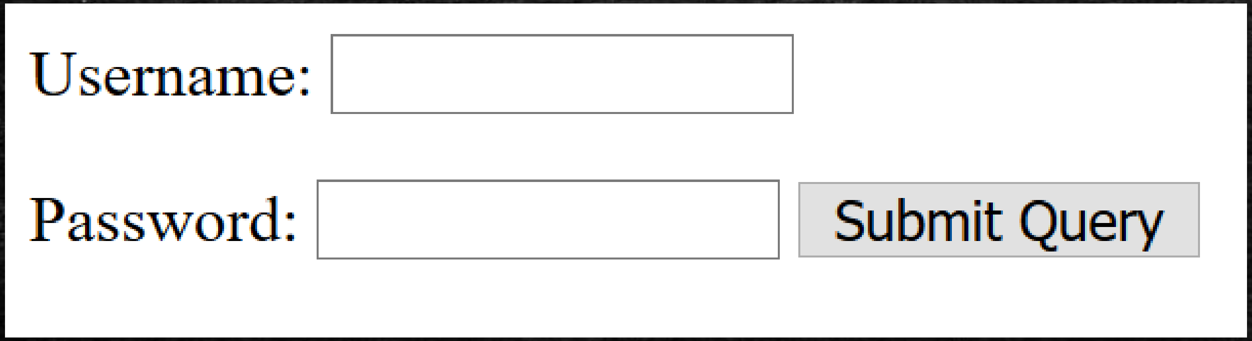
Drop-down boxes

When a user must pick one of a number of options from a list.

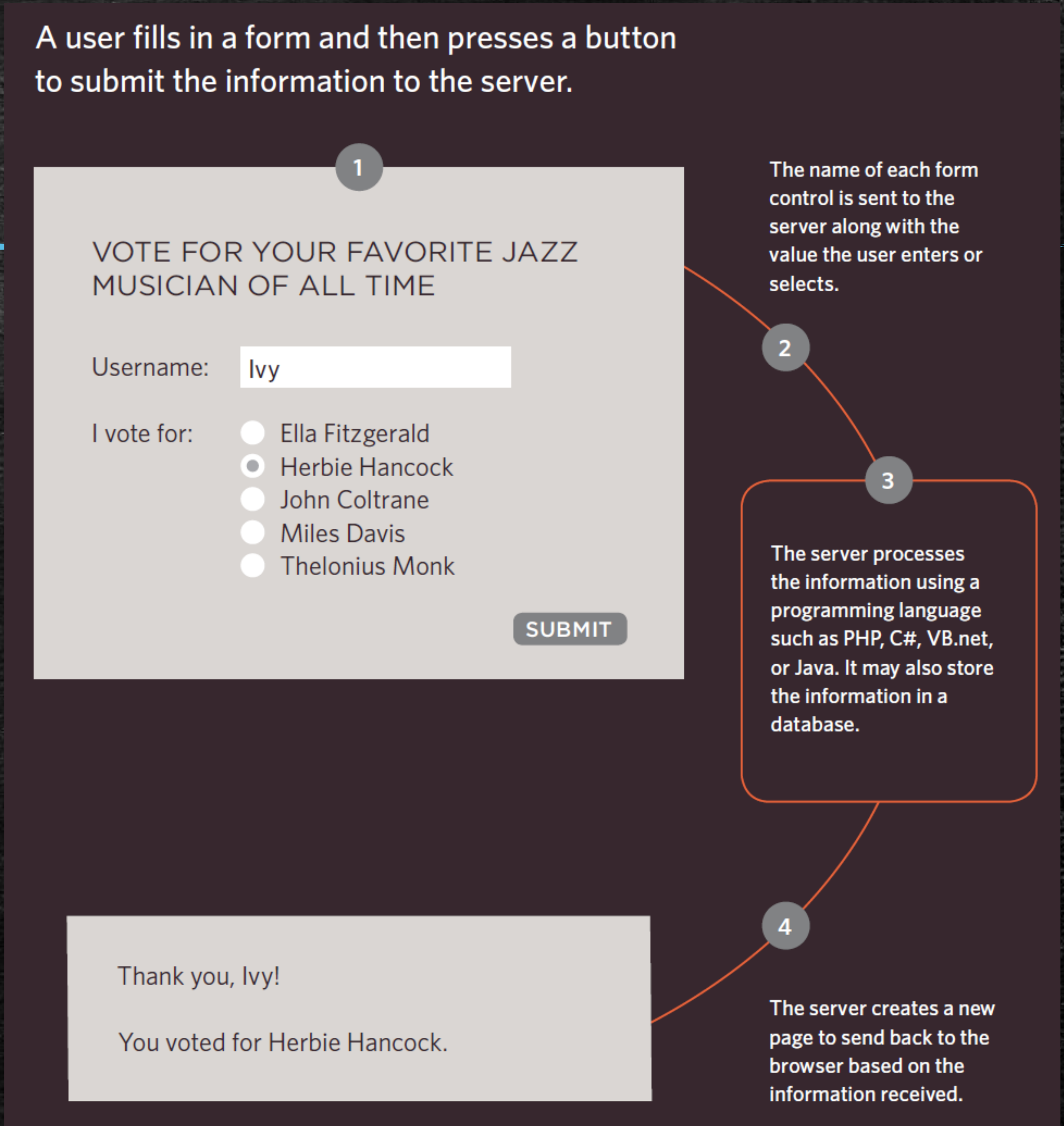
iPod



```
<form action="http://hjessmith.com/teaching/form_example.php">
    <p>Username:
        <input type="text" name="name" size="15"
        maxlength="30" />
    </p>
    <p>Password:
        <input type="password" name="password" size="15"
        maxlength="30" />
        <input type="submit" />
    </p>
</form>
```



Data Collection from Forms



A user fills in a form and then presses a button to submit the information to the server.

VOTE FOR YOUR FAVORITE JAZZ MUSICIAN OF ALL TIME

Username: Ivy

I vote for:
○ Ella Fitzgerald
● Herbie Hancock
○ John Coltrane
○ Miles Davis
○ Thelonius Monk

The name of each form control is sent to the server along with the value the user enters or selects.

The server processes the information using a programming language such as PHP, C#, VB.net, or Java. It may also store the information in a database.

The server creates a new page to send back to the browser based on the information received.

Thank you, Ivy!

You voted for Herbie Hancock.

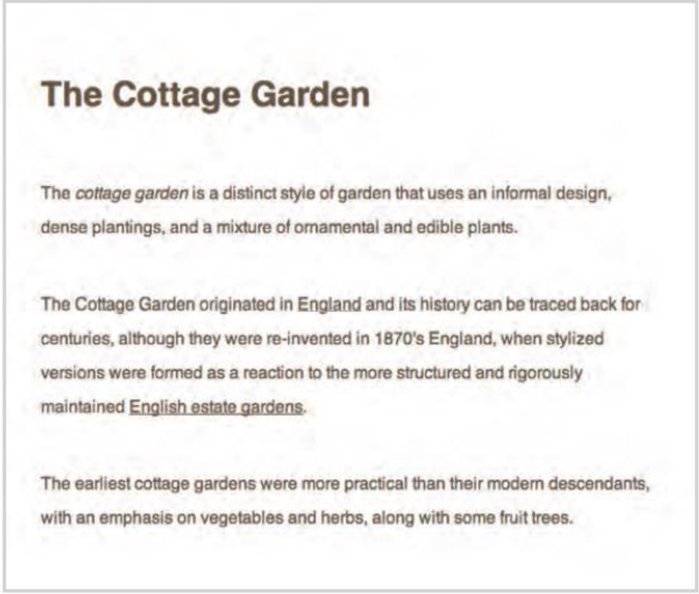
Forms Example

On to CSS (Cascading Style Sheets)

The key to understanding how CSS works is to imagine that there is an invisible box around every HTML element.

On this page, you can see a basic HTML page.

On the right hand page, you can see the same HTML page, but I have added outlines to each of the elements so that you can see how CSS will treat each element as if it lives inside its own box.



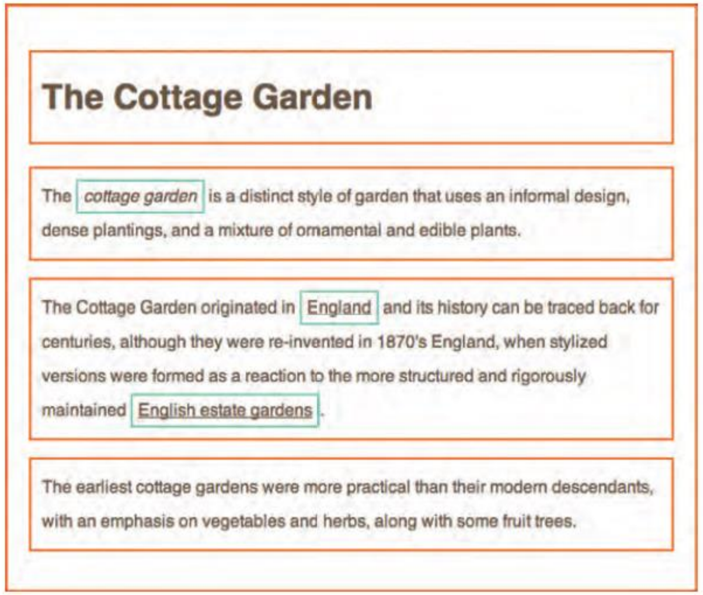
The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in England and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained English estate gardens.

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

CSS allows you to create rules that control the way that each individual box (and the contents of that box) is presented.



The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in England and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained English estate gardens.

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

In this example, block level elements are shown with red borders, and inline elements have green borders.

The `<body>` element creates the first box, then the `<h1>`, `<h2>`, `<p>`, `<i>`, and `<a>` elements each create their own boxes within it.

Using CSS, you could add a border around any of the boxes, specify its width and height, or add a background color. You could also control text inside a box — for example, its color, size, and the typeface used.

BLOCK & INLINE ELEMENTS

You may remember from pages 185-186 that in there is a difference between block level and inline elements and how how browsers display them.

Block level elements look like they start on a new line. Examples include the `<h1>`-`<h6>`, `<p>` and `<div>` elements.

Inline elements flow within the text and do not start on a new line. Examples include `<b>`, `<i>`, `<img>`, `<em>` and `<span>`.

EXAMPLE STYLES

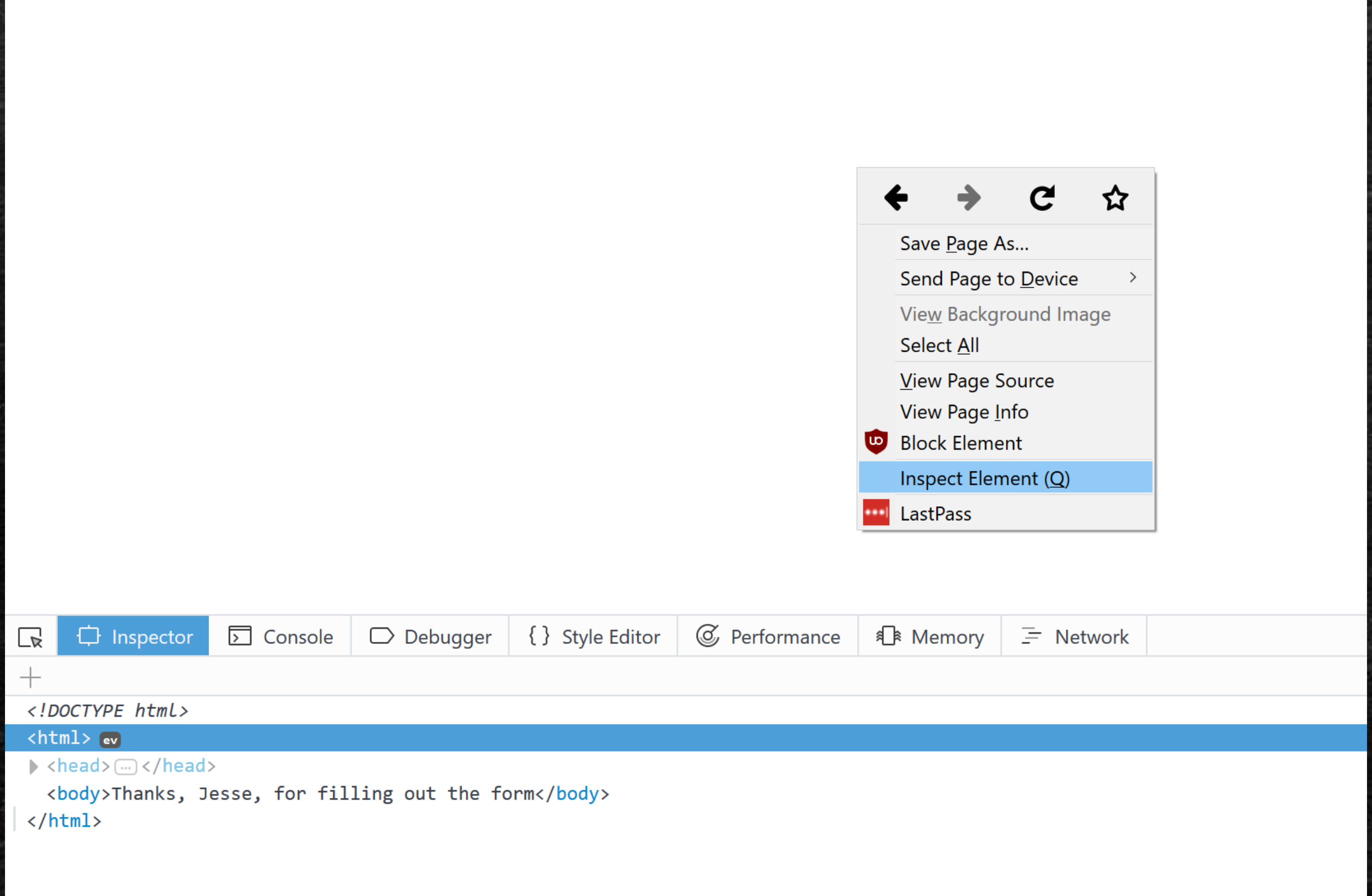
BOXES

Width and height
Borders (color, width, and style)
Background color and images
Position in the browser window.

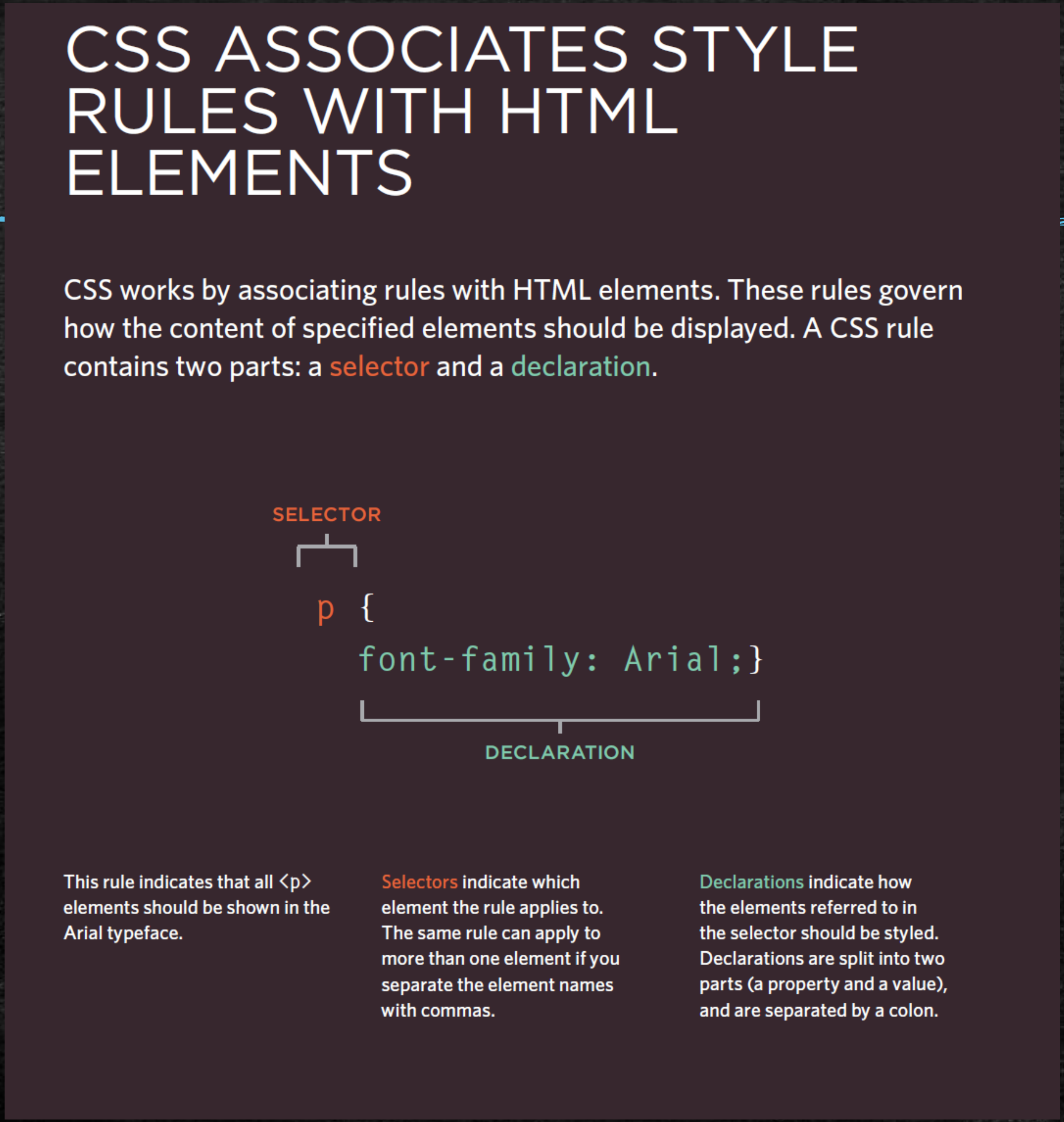
TEXT

Typeface
Size
Color
Italics, bold, uppercase, lowercase, small-caps

Viewing CSS Boxes using Browser Developer Tools



```
<!DOCTYPE html>
<html>
  <head>…</head>
  <body>Thanks, Jesse, for filling out the form</body>
</html>
```



CSS ASSOCIATES STYLE RULES WITH HTML ELEMENTS

CSS works by associating rules with HTML elements. These rules govern how the content of specified elements should be displayed. A CSS rule contains two parts: a selector and a declaration.

SELECTOR

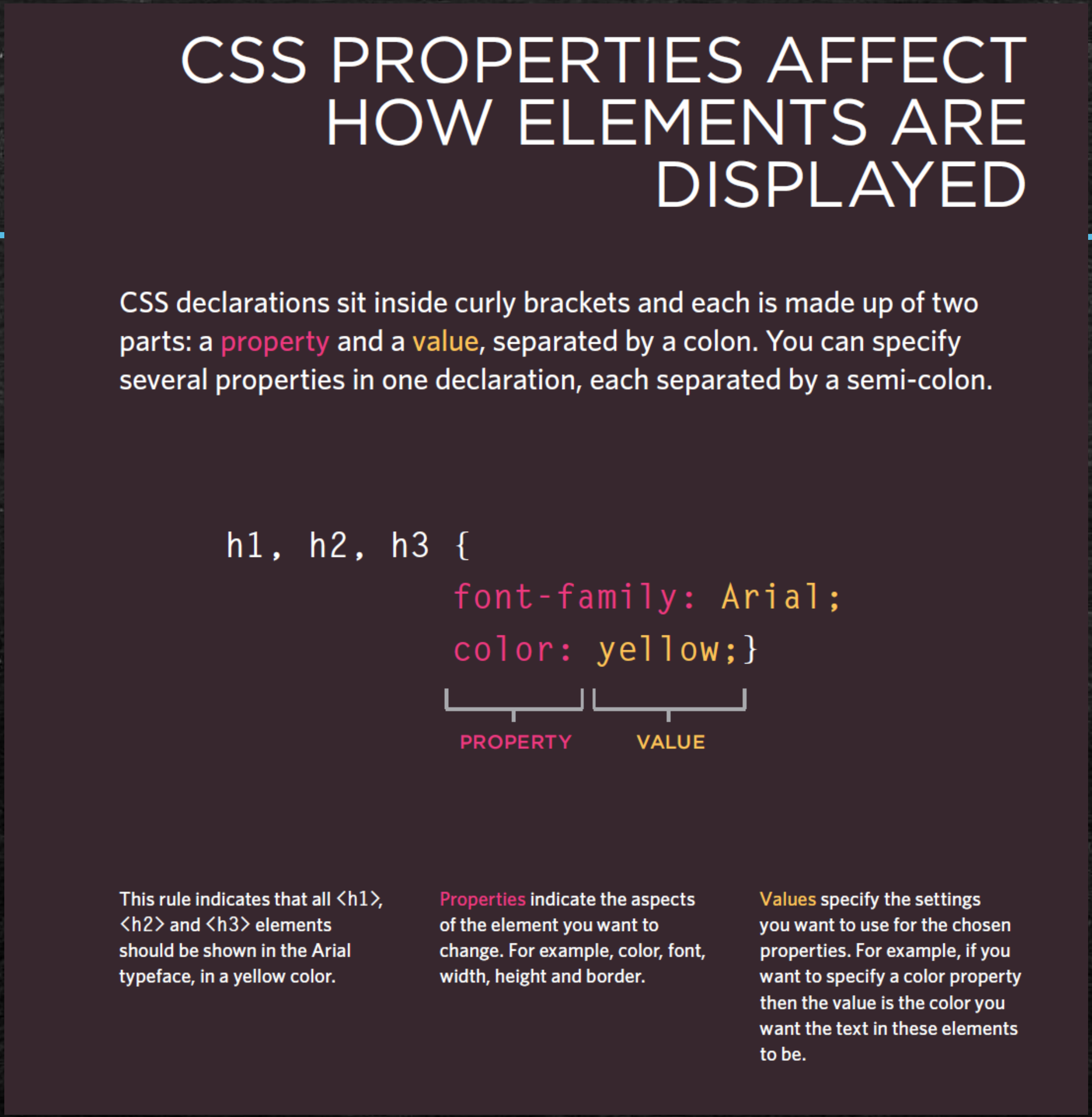
```
p {
  font-family: Arial;}
```

DECLARATION

This rule indicates that all <p> elements should be shown in the Arial typeface.

Selectors indicate which element the rule applies to. The same rule can apply to more than one element if you separate the element names with commas.

Declarations indicate how the elements referred to in the selector should be styled. Declarations are split into two parts (a property and a value), and are separated by a colon.



CSS PROPERTIES AFFECT HOW ELEMENTS ARE DISPLAYED

CSS declarations sit inside curly brackets and each is made up of two parts: a property and a value, separated by a colon. You can specify several properties in one declaration, each separated by a semi-colon.

```
h1, h2, h3 {
          font-family: Arial;
          color: yellow;}
```

PROPERTY

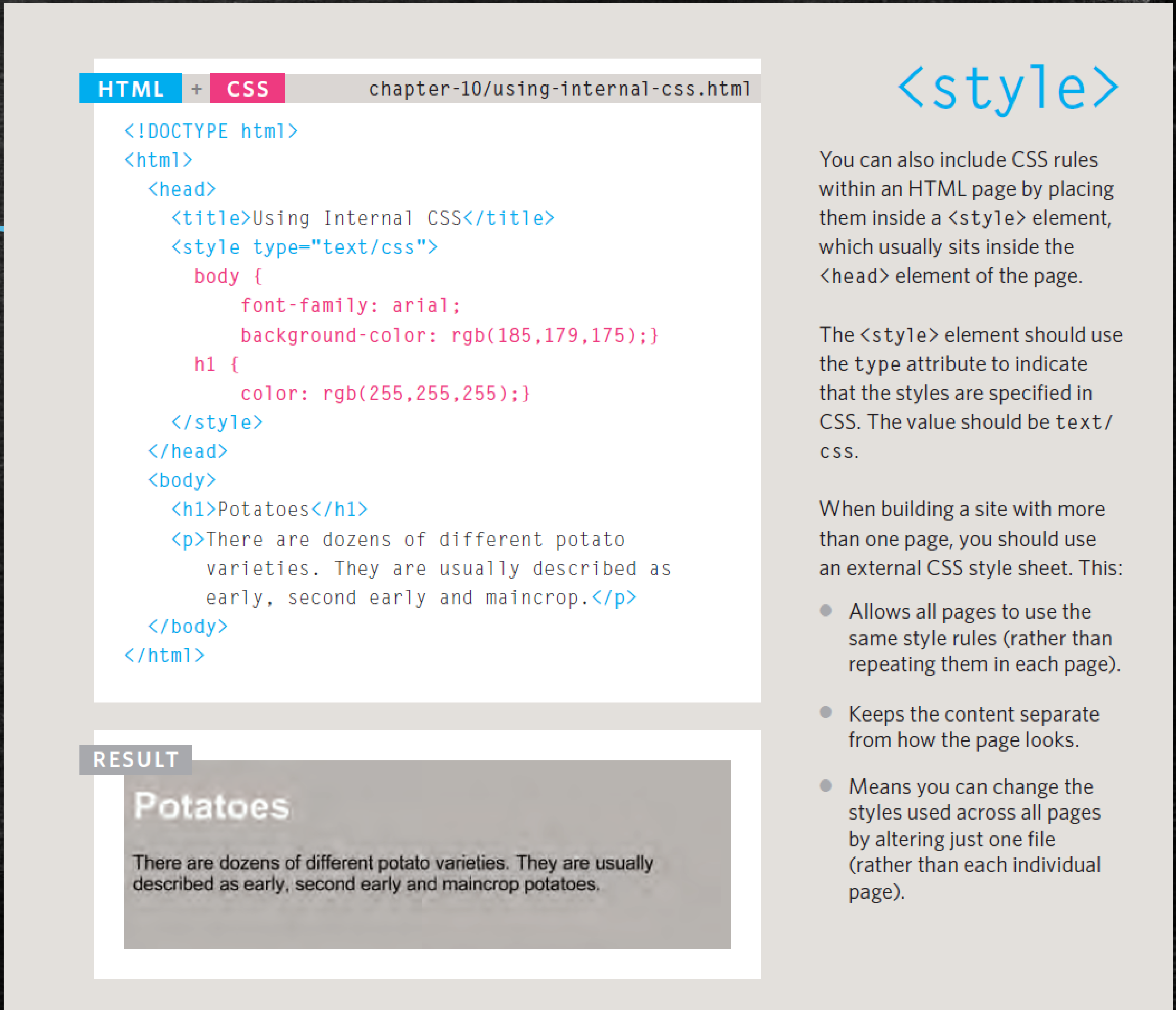
VALUE

This rule indicates that all <h1>, <h2> and <h3> elements should be shown in the Arial typeface, in a yellow color.

Properties indicate the aspects of the element you want to change. For example, color, font, width, height and border.

Values specify the settings you want to use for the chosen properties. For example, if you want to specify a color property then the value is the color you want the text in these elements to be.

Embedding CSS inside a page



HTML + CSS chapter-10/using-internal-css.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using Internal CSS</title>
    <style type="text/css">
      body {
          font-family: arial;
          background-color: rgb(185,179,175);}
      h1 {
          color: rgb(255,255,255);}
    </style>
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
       varieties. They are usually described as
       early, second early and maincrop.</p>
  </body>
</html>
```

RESULT

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.

<style>

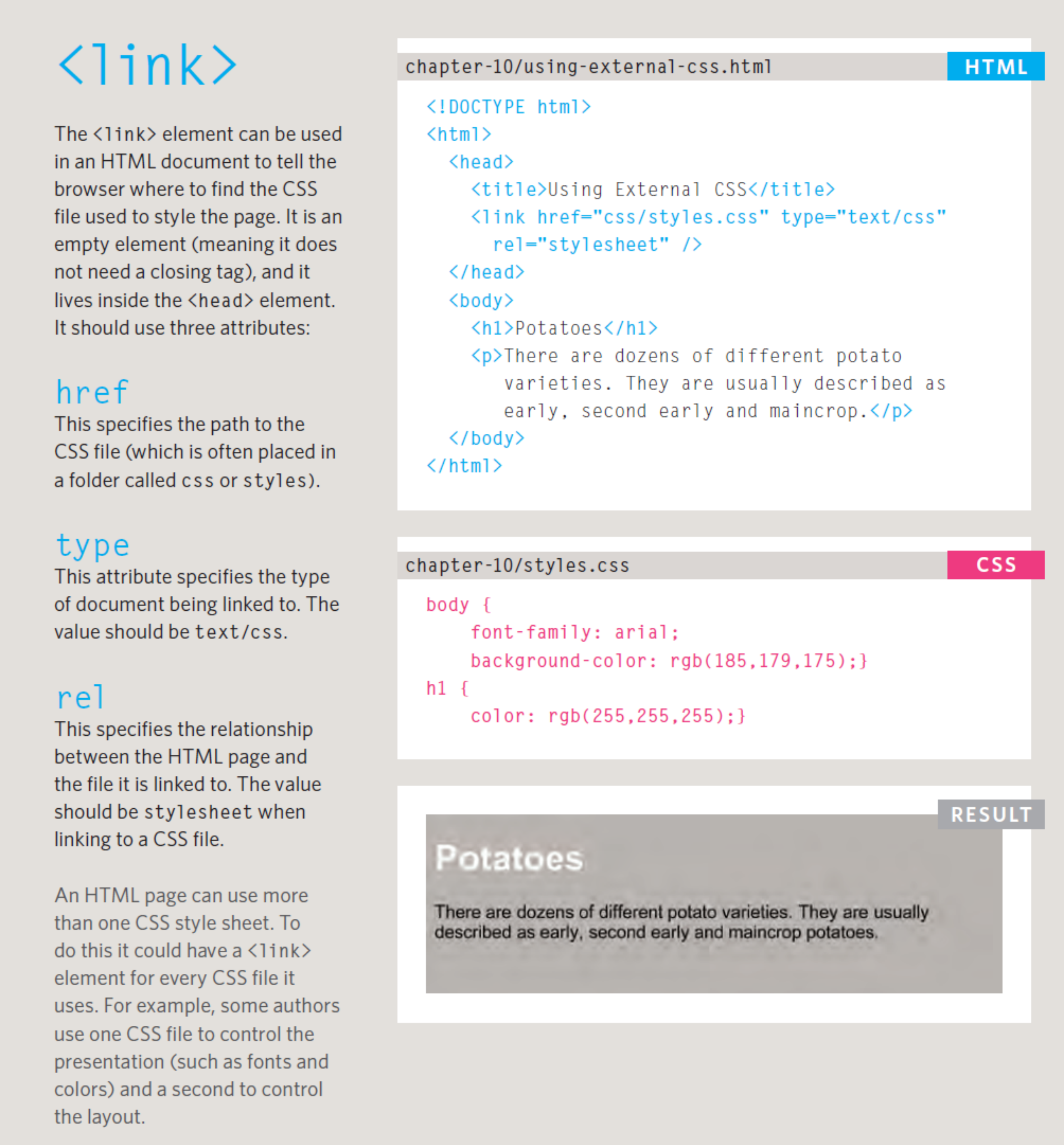
You can also include CSS rules within an HTML page by placing them inside a `<style>` element, which usually sits inside the `<head>` element of the page.

The `<style>` element should use the `type` attribute to indicate that the styles are specified in CSS. The value should be `text/css`.

When building a site with more than one page, you should use an external CSS style sheet. This:

- Allows all pages to use the same style rules (rather than repeating them in each page).
- Keeps the content separate from how the page looks.
- Means you can change the styles used across all pages by altering just one file (rather than each individual page).

Linking an External Stylesheet



<link>

The <link> element can be used in an HTML document to tell the browser where to find the CSS file used to style the page. It is an empty element (meaning it does not need a closing tag), and it lives inside the <head> element. It should use three attributes:

href

This specifies the path to the CSS file (which is often placed in a folder called css or styles).

type

This attribute specifies the type of document being linked to. The value should be text/css.

rel

This specifies the relationship between the HTML page and the file it is linked to. The value should be stylesheet when linking to a CSS file.

An HTML page can use more than one CSS style sheet. To do this it could have a <link> element for every CSS file it uses. For example, some authors use one CSS file to control the presentation (such as fonts and colors) and a second to control the layout.

chapter-10/using-external-css.html HTML

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" type="text/css"
      rel="stylesheet" />
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
       varieties. They are usually described as
       early, second early and maincrop.</p>
  </body>
</html>
```

chapter-10/styles.css CSS

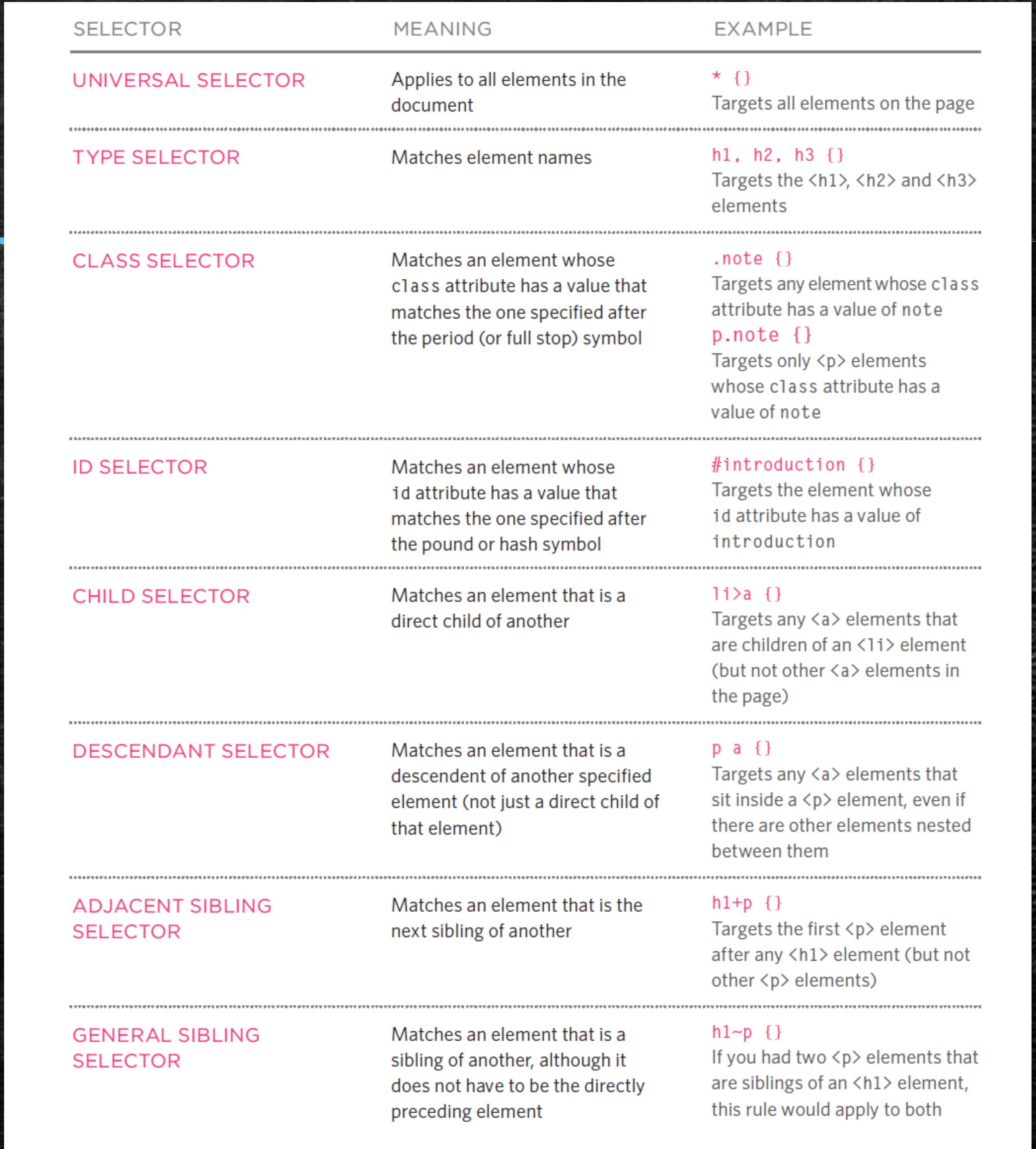
```css
body {
    font-family: arial;
    background-color: rgb(185,179,175);}
h1 {
    color: rgb(255,255,255);}
```

RESULT

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.

CSS Selectors

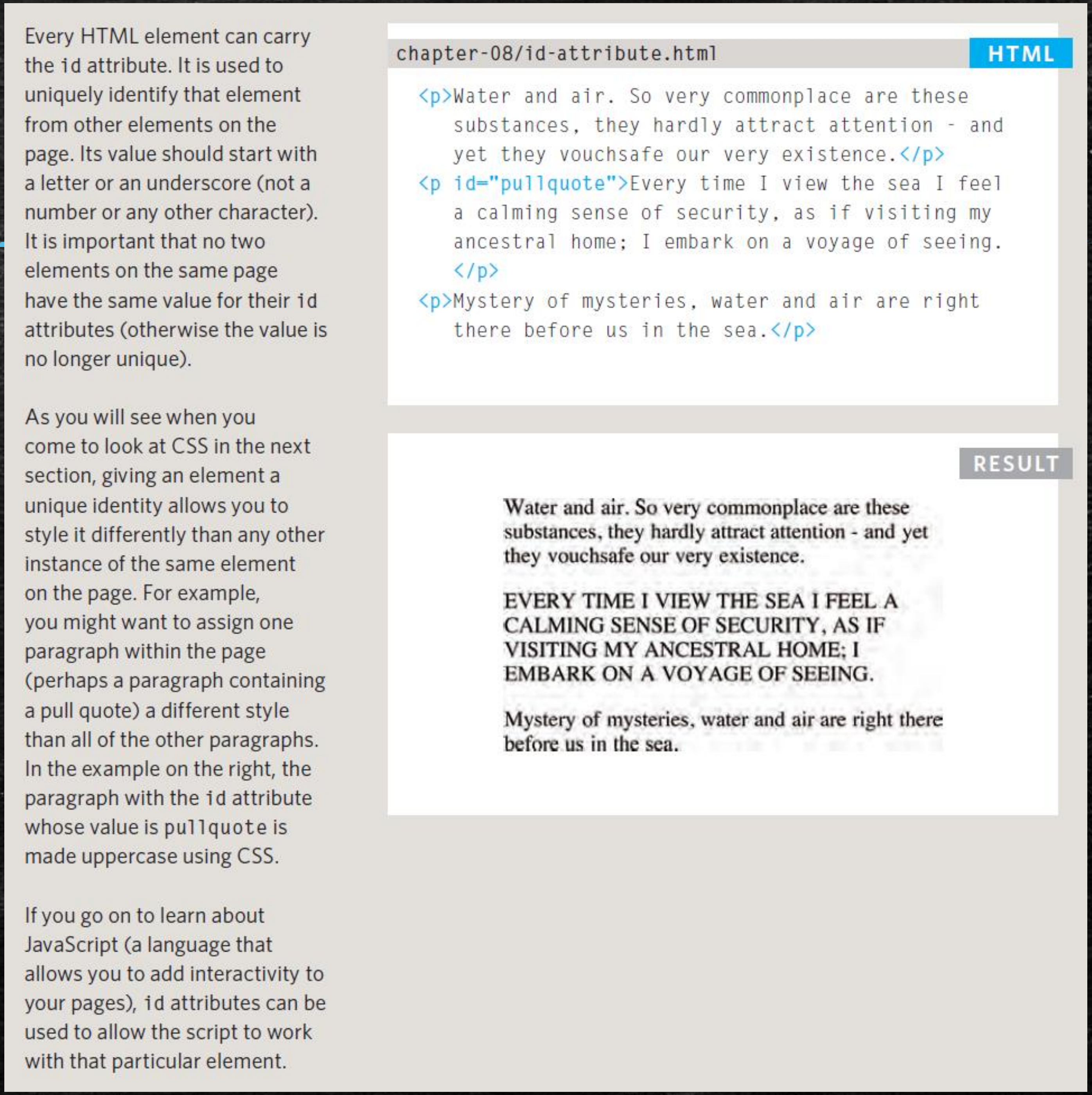


SELECTOR	MEANING	EXAMPLE
UNIVERSAL SELECTOR	Applies to all elements in the document	`* {}` Targets all elements on the page
TYPE SELECTOR	Matches element names	`h1, h2, h3 {}` Targets the `<h1>`, `<h2>` and `<h3>` elements

CLASS SELECTOR	Matches an element whose `class` attribute has a value that matches the one specified after the period (or full stop) symbol	`.note {}`
		Targets any element whose `class` attribute has a value of `note`
		`p.note {}`
		Targets only `<p>` elements whose `class` attribute has a value of `note`

ID SELECTOR	Matches an element whose `id` attribute has a value that matches the one specified after the pound or hash symbol	`#introduction {}` Targets the element whose `id` attribute has a value of `introduction`
CHILD SELECTOR	Matches an element that is a direct child of another	`li>a {}` Targets any `<a>` elements that are children of an `<li>` element (but not other `<a>` elements in the page)
DESCENDANT SELECTOR	Matches an element that is a descendent of another specified element (not just a direct child of that element)	`p a {}` Targets any `<a>` elements that sit inside a `<p>` element, even if there are other elements nested between them
ADJACENT SIBLING SELECTOR	Matches an element that is the next sibling of another	`h1+p {}` Targets the first `<p>` element after any `<h1>` element (but not other `<p>` elements)
GENERAL SIBLING SELECTOR	Matches an element that is a sibling of another, although it does not have to be the directly preceding element	`h1~p {}` If you had two `<p>` elements that are siblings of an `<h1>` element, this rule would apply to both

ID Attributes



Every HTML element can carry the id attribute. It is used to uniquely identify that element from other elements on the page. Its value should start with a letter or an underscore (not a number or any other character). It is important that no two elements on the same page have the same value for their id attributes (otherwise the value is no longer unique).

As you will see when you come to look at CSS in the next section, giving an element a unique identity allows you to style it differently than any other instance of the same element on the page. For example, you might want to assign one paragraph within the page (perhaps a paragraph containing a pull quote) a different style than all of the other paragraphs. In the example on the right, the paragraph with the id attribute whose value is pullquote is made uppercase using CSS.

If you go on to learn about JavaScript (a language that allows you to add interactivity to your pages), id attributes can be used to allow the script to work with that particular element.

chapter-08/id-attribute.html

HTML

```html
<p>Water and air. So very commonplace are these
   substances, they hardly attract attention - and
   yet they vouchsafe our very existence.</p>
<p id="pullquote">Every time I view the sea I feel
   a calming sense of security, as if visiting my
   ancestral home; I embark on a voyage of seeing.
   </p>
<p>Mystery of mysteries, water and air are right
   there before us in the sea.</p>
```

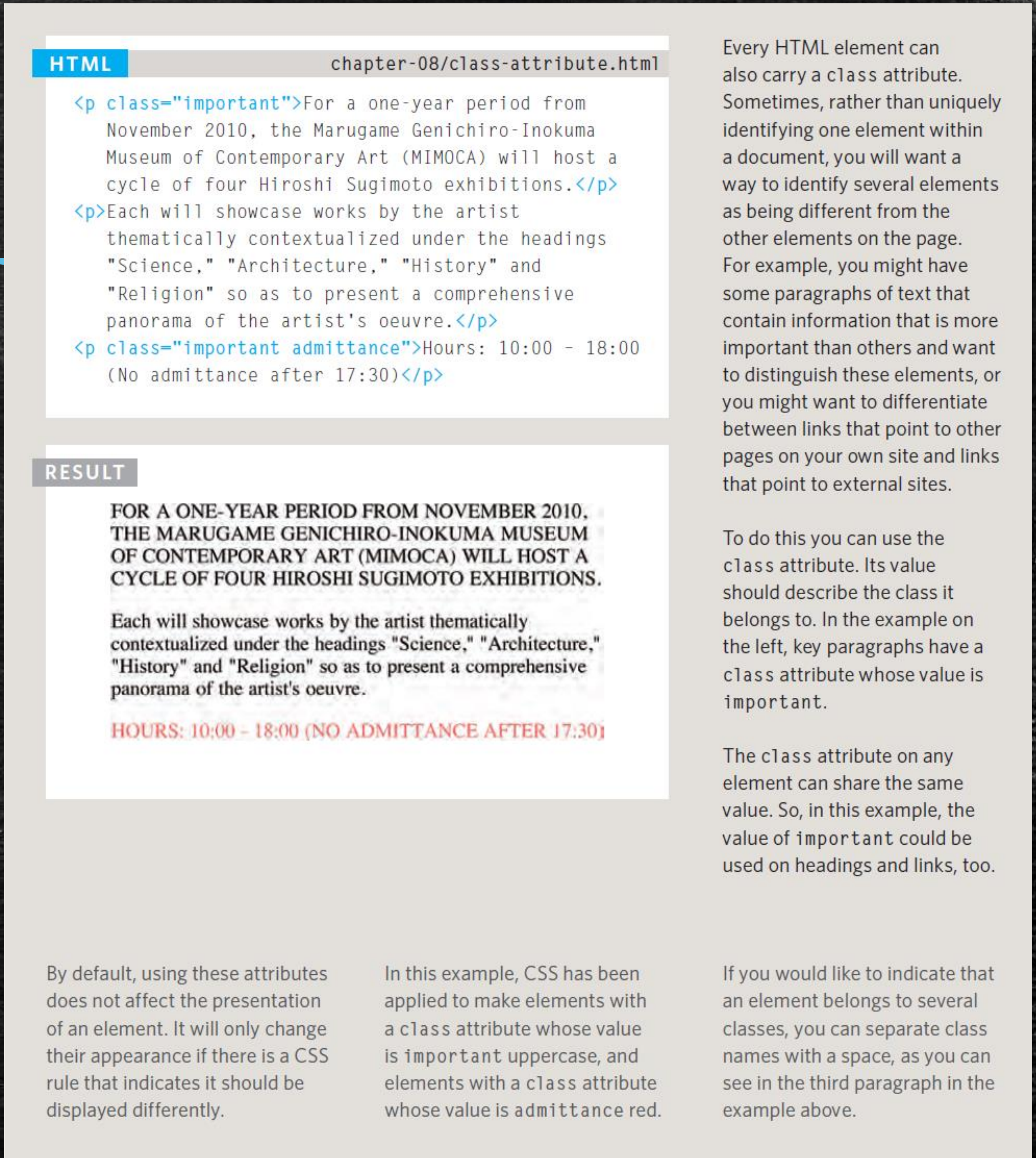
RESULT

Water and air. So very commonplace are these substances, they hardly attract attention - and yet they vouchsafe our very existence.

EVERY TIME I VIEW THE SEA I FEEL A CALMING SENSE OF SECURITY, AS IF VISITING MY ANCESTRAL HOME; I EMBARK ON A VOYAGE OF SEEING.

Mystery of mysteries, water and air are right there before us in the sea.

Class Attributes



Every HTML element can also carry a class attribute. Sometimes, rather than uniquely identifying one element within a document, you will want a way to identify several elements as being different from the other elements on the page. For example, you might have some paragraphs of text that contain information that is more important than others and want to distinguish these elements, or you might want to differentiate between links that point to other pages on your own site and links that point to external sites.

To do this you can use the class attribute. Its value should describe the class it belongs to. In the example on the left, key paragraphs have a class attribute whose value is important.

The class attribute on any element can share the same value. So, in this example, the value of important could be used on headings and links, too.

HTML chapter-08/class-attribute.html

```
<p class="important">For a one-year period from
  November 2010, the Marugame Genichiro-Inokuma
  Museum of Contemporary Art (MIMOCA) will host a
  cycle of four Hiroshi Sugimoto exhibitions.</p>
<p>Each will showcase works by the artist
  thematically contextualized under the headings
  "Science," "Architecture," "History" and
  "Religion" so as to present a comprehensive
  panorama of the artist's oeuvre.</p>
<p class="important admittance">Hours: 10:00 - 18:00
  (No admittance after 17:30)</p>
```

RESULT

FOR A ONE-YEAR PERIOD FROM NOVEMBER 2010, THE MARUGAME GENICHIRO-INOKUMA MUSEUM OF CONTEMPORARY ART (MIMOCA) WILL HOST A CYCLE OF FOUR HIROSHI SUGIMOTO EXHIBITIONS.

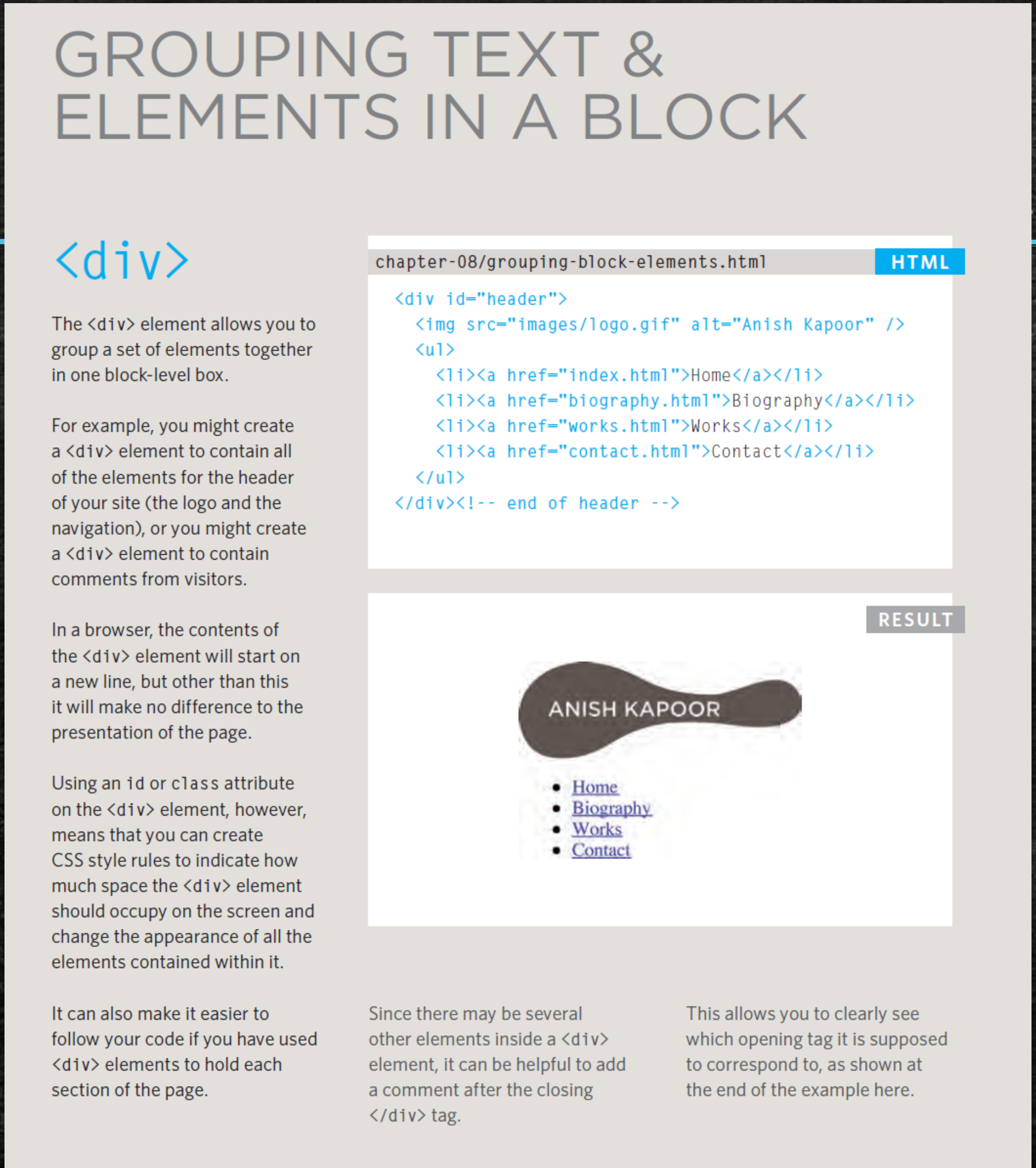
Each will showcase works by the artist thematically contextualized under the headings "Science," "Architecture," "History" and "Religion" so as to present a comprehensive panorama of the artist's oeuvre.

HOURS: 10:00 - 18:00 (NO ADMITTANCE AFTER 17:30)

By default, using these attributes does not affect the presentation of an element. It will only change their appearance if there is a CSS rule that indicates it should be displayed differently.

In this example, CSS has been applied to make elements with a class attribute whose value is important uppercase, and elements with a class attribute whose value is admittance red.

If you would like to indicate that an element belongs to several classes, you can separate class names with a space, as you can see in the third paragraph in the example above.



GROUPING TEXT & ELEMENTS IN A BLOCK

<div>

The <div> element allows you to group a set of elements together in one block-level box.

For example, you might create a <div> element to contain all of the elements for the header of your site (the logo and the navigation), or you might create a <div> element to contain comments from visitors.

In a browser, the contents of the <div> element will start on a new line, but other than this it will make no difference to the presentation of the page.

Using an id or class attribute on the <div> element, however, means that you can create CSS style rules to indicate how much space the <div> element should occupy on the screen and change the appearance of all the elements contained within it.

It can also make it easier to follow your code if you have used <div> elements to hold each section of the page.

chapter-08/grouping-block-elements.html HTML

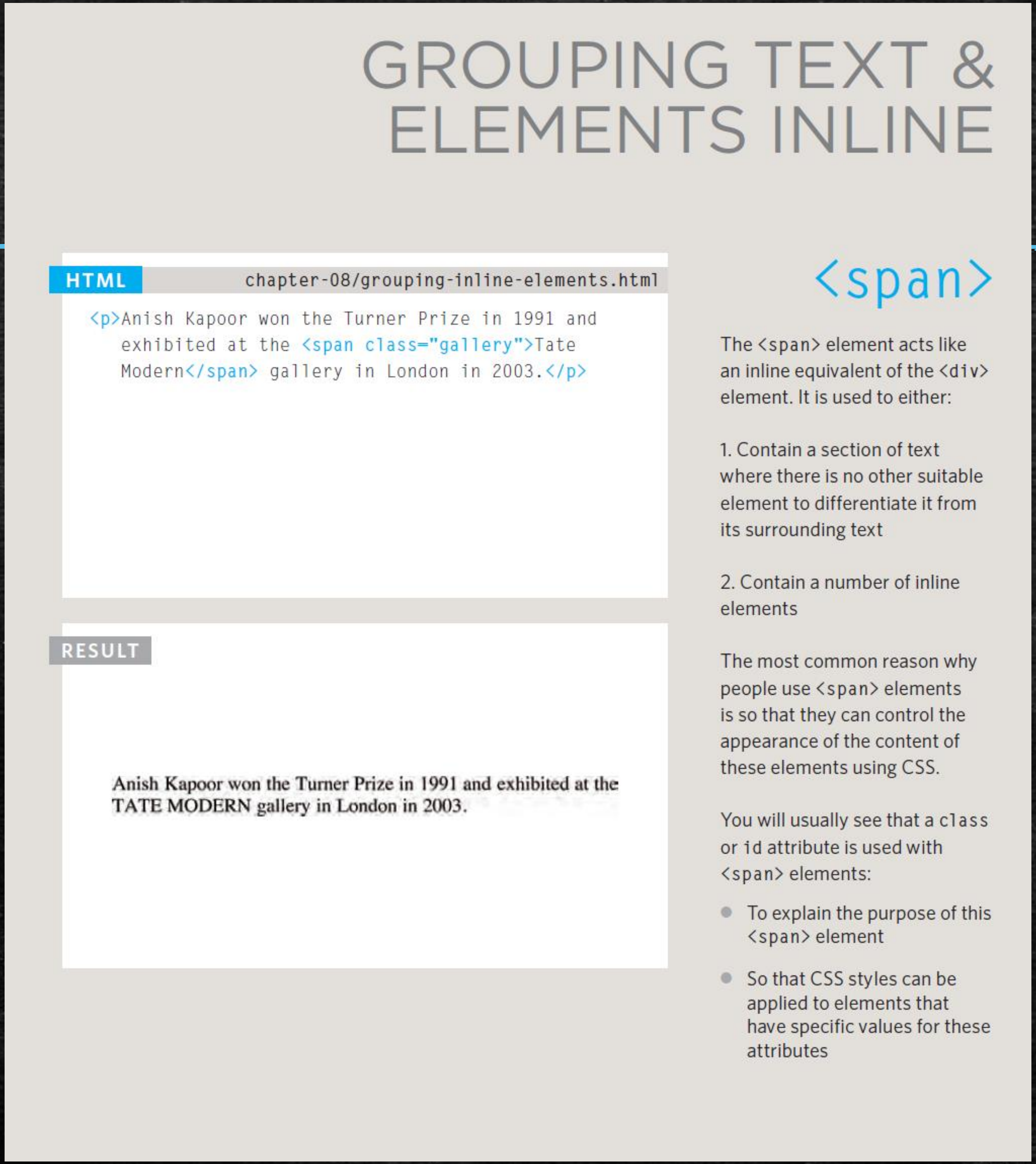
```
<div id="header">
  <img src="images/logo.gif" alt="Anish Kapoor" />
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="biography.html">Biography</a></li>
    <li><a href="works.html">Works</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</div><!-- end of header -->
```

RESULT

- Home
- Biography
- Works
- Contact

Since there may be several other elements inside a <div> element, it can be helpful to add a comment after the closing </div> tag.

This allows you to clearly see which opening tag it is supposed to correspond to, as shown at the end of the example here.



GROUPING TEXT & ELEMENTS INLINE

HTML chapter-08/grouping-inline-elements.html

```
<p>Anish Kapoor won the Turner Prize in 1991 and
  exhibited at the <span class="gallery">Tate
  Modern</span> gallery in London in 2003.</p>
```

RESULT

Anish Kapoor won the Turner Prize in 1991 and exhibited at the TATE MODERN gallery in London in 2003.

<span>

The <span> element acts like an inline equivalent of the <div> element. It is used to either:

1. Contain a section of text where there is no other suitable element to differentiate it from its surrounding text

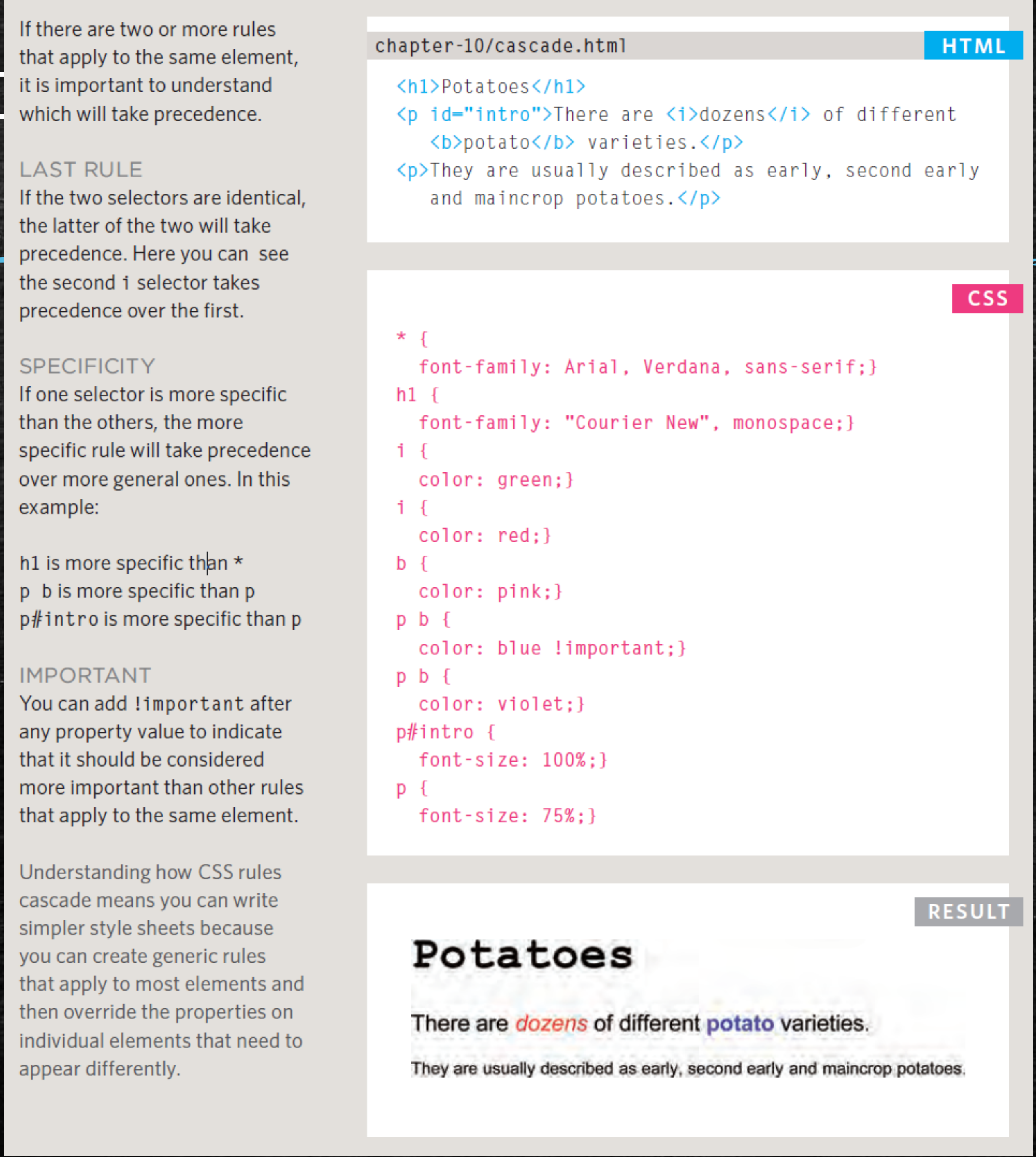
2. Contain a number of inline elements

The most common reason why people use <span> elements is so that they can control the appearance of the content of these elements using CSS.

You will usually see that a class or id attribute is used with <span> elements:

- To explain the purpose of this <span> element
- So that CSS styles can be applied to elements that have specific values for these attributes

How rul



If there are two or more rules that apply to the same element, it is important to understand which will take precedence.

LAST RULE

If the two selectors are identical, the latter of the two will take precedence. Here you can see the second i selector takes precedence over the first.

SPECIFICITY

If one selector is more specific than the others, the more specific rule will take precedence over more general ones. In this example:

h1 is more specific than *
p b is more specific than p
p#intro is more specific than p

IMPORTANT

You can add !important after any property value to indicate that it should be considered more important than other rules that apply to the same element.

Understanding how CSS rules cascade means you can write simpler style sheets because you can create generic rules that apply to most elements and then override the properties on individual elements that need to appear differently.

chapter-10/cascade.html HTML

```
<h1>Potatoes</h1>
<p id="intro">There are <i>dozens</i> of different
   <b>potato</b> varieties.</p>
<p>They are usually described as early, second early
   and maincrop potatoes.</p>
```

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;}
h1 {
  font-family: "Courier New", monospace;}
i {
  color: green;}
i {
  color: red;}
b {
  color: pink;}
p b {
  color: blue !important;}
p b {
  color: violet;}
p#intro {
  font-size: 100%;}
p {
  font-size: 75%;}
```

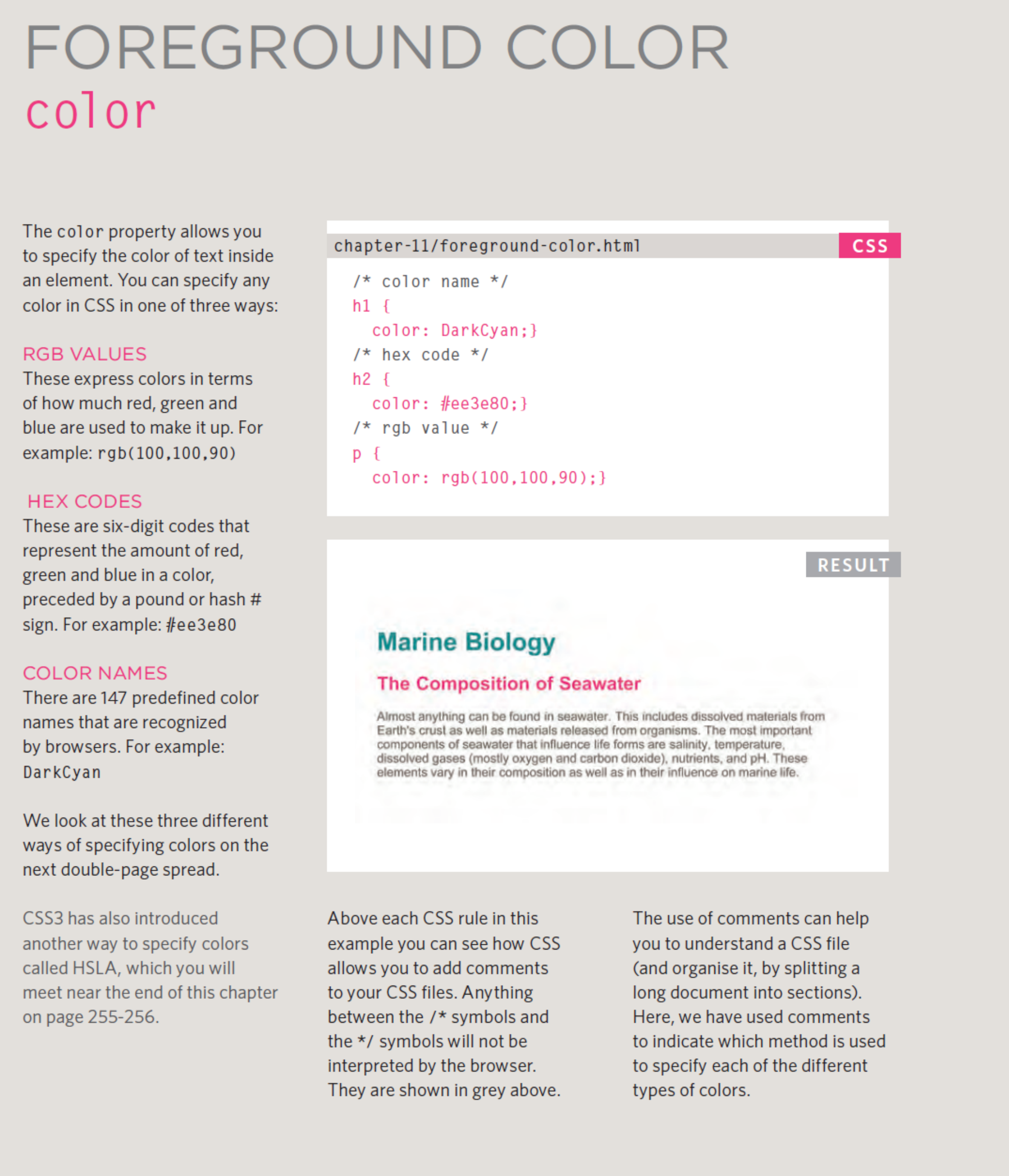
RESULT

Potatoes

There are *dozens* of different **potato** varieties.

They are usually described as early, second early and maincrop potatoes.

Text Color



FOREGROUND COLOR

color

The color property allows you to specify the color of text inside an element. You can specify any color in CSS in one of three ways:

RGB VALUES

These express colors in terms of how much red, green and blue are used to make it up. For example: rgb(100,100,90)

HEX CODES

These are six-digit codes that represent the amount of red, green and blue in a color, preceded by a pound or hash # sign. For example: #ee3e80

COLOR NAMES

There are 147 predefined color names that are recognized by browsers. For example: DarkCyan

We look at these three different ways of specifying colors on the next double-page spread.

CSS3 has also introduced another way to specify colors called HSLA, which you will meet near the end of this chapter on page 255-256.

chapter-11/foreground-color.html CSS

```
/* color name */
h1 {
  color: DarkCyan;}
/* hex code */
h2 {
  color: #ee3e80;}
/* rgb value */
p {
  color: rgb(100,100,90);}
```

RESULT

Marine Biology

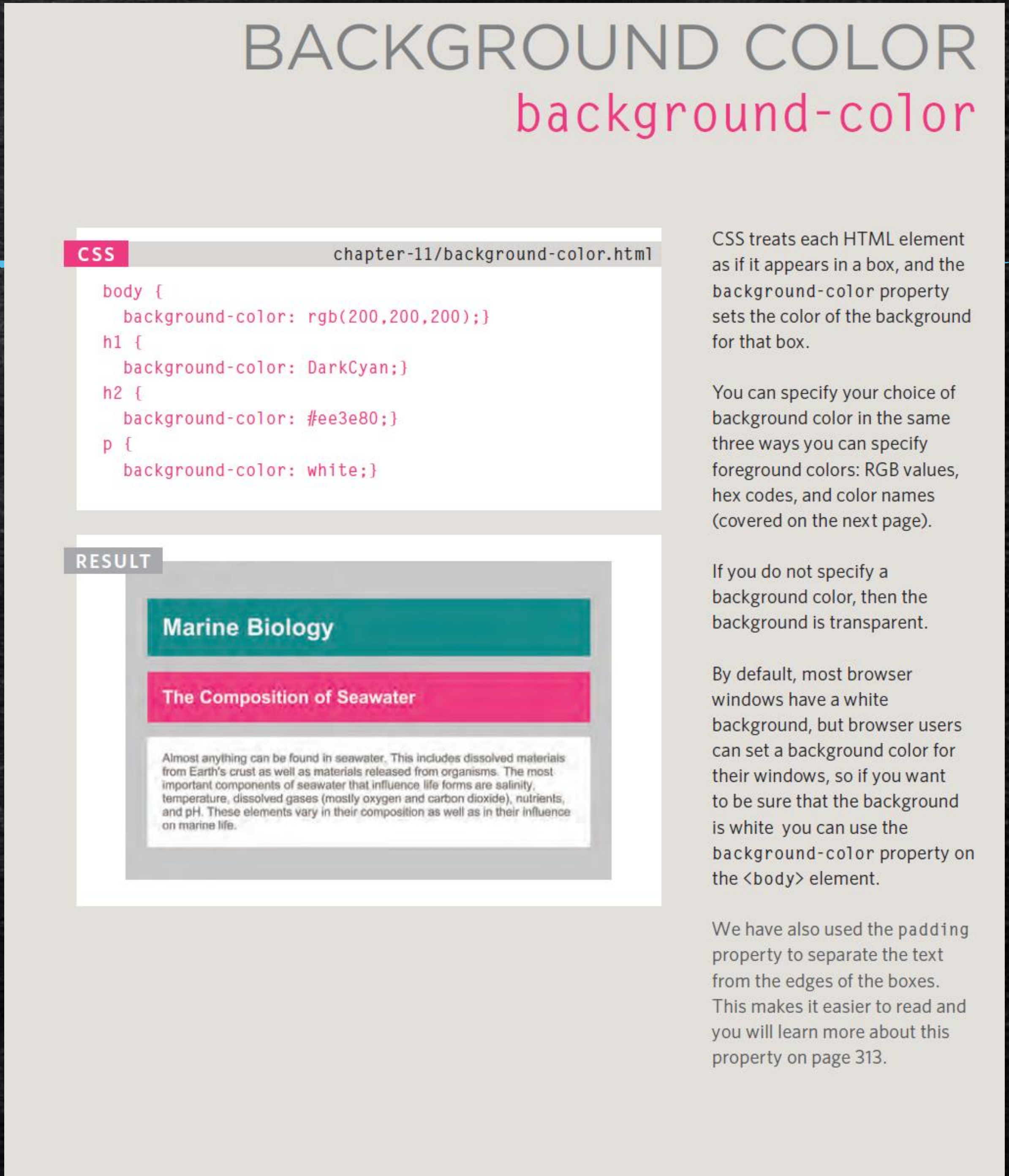
The Composition of Seawater

Almost anything can be found in seawater. This includes dissolved materials from Earth's crust as well as materials released from organisms. The most important components of seawater that influence life forms are salinity, temperature, dissolved gases (mostly oxygen and carbon dioxide), nutrients, and pH. These elements vary in their composition as well as in their influence on marine life.

Above each CSS rule in this example you can see how CSS allows you to add comments to your CSS files. Anything between the /* symbols and the */ symbols will not be interpreted by the browser. They are shown in grey above.

The use of comments can help you to understand a CSS file (and organise it, by splitting a long document into sections). Here, we have used comments to indicate which method is used to specify each of the different types of colors.

Background Color



BACKGROUND COLOR
background-color

CSS chapter-11/background-color.html

```
body {
  background-color: rgb(200,200,200);}
h1 {
  background-color: DarkCyan;}
h2 {
  background-color: #ee3e80;}
p {
  background-color: white;}
```

RESULT

Marine Biology

The Composition of Seawater

Almost anything can be found in seawater. This includes dissolved materials from Earth's crust as well as materials released from organisms. The most important components of seawater that influence life forms are salinity, temperature, dissolved gases (mostly oxygen and carbon dioxide), nutrients, and pH. These elements vary in their composition as well as in their influence on marine life.

CSS treats each HTML element as if it appears in a box, and the `background-color` property sets the color of the background for that box.

You can specify your choice of background color in the same three ways you can specify foreground colors: RGB values, hex codes, and color names (covered on the next page).

If you do not specify a background color, then the background is transparent.

By default, most browser windows have a white background, but browser users can set a background color for their windows, so if you want to be sure that the background is white you can use the `background-color` property on the `<body>` element.

We have also used the `padding` property to separate the text from the edges of the boxes. This makes it easier to read and you will learn more about this property on page 313.

Online Hex Color Picker

- http://htmlcolorcodes.com/color-picker/

Coding Exercise