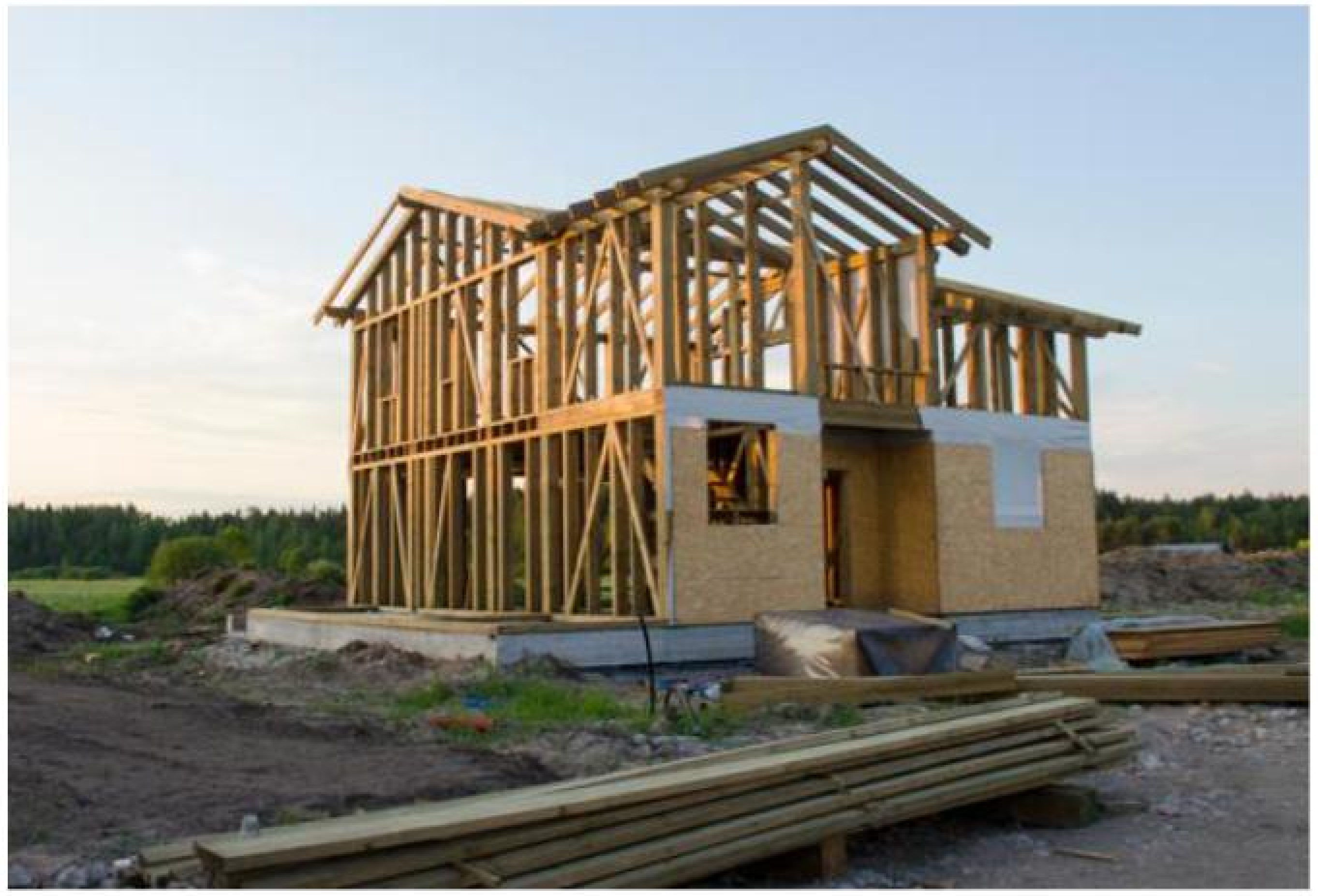
Slide Note

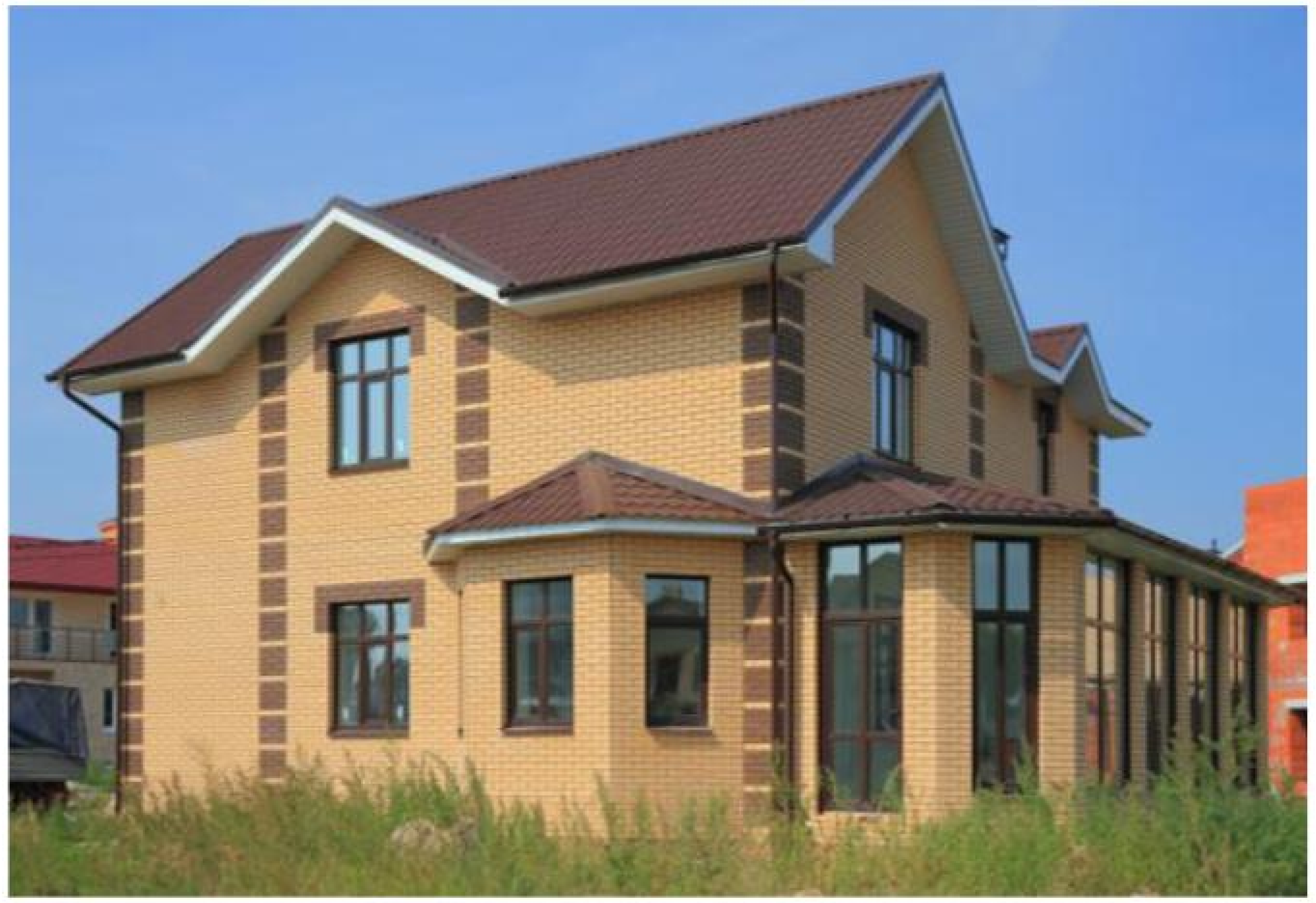
How are these two images different?

On the left we see the structure of the house.

On the right, the structure looks complete with both the interiors and exteriors done. The house is fit for habitation now.

How Are These Two Images Different?



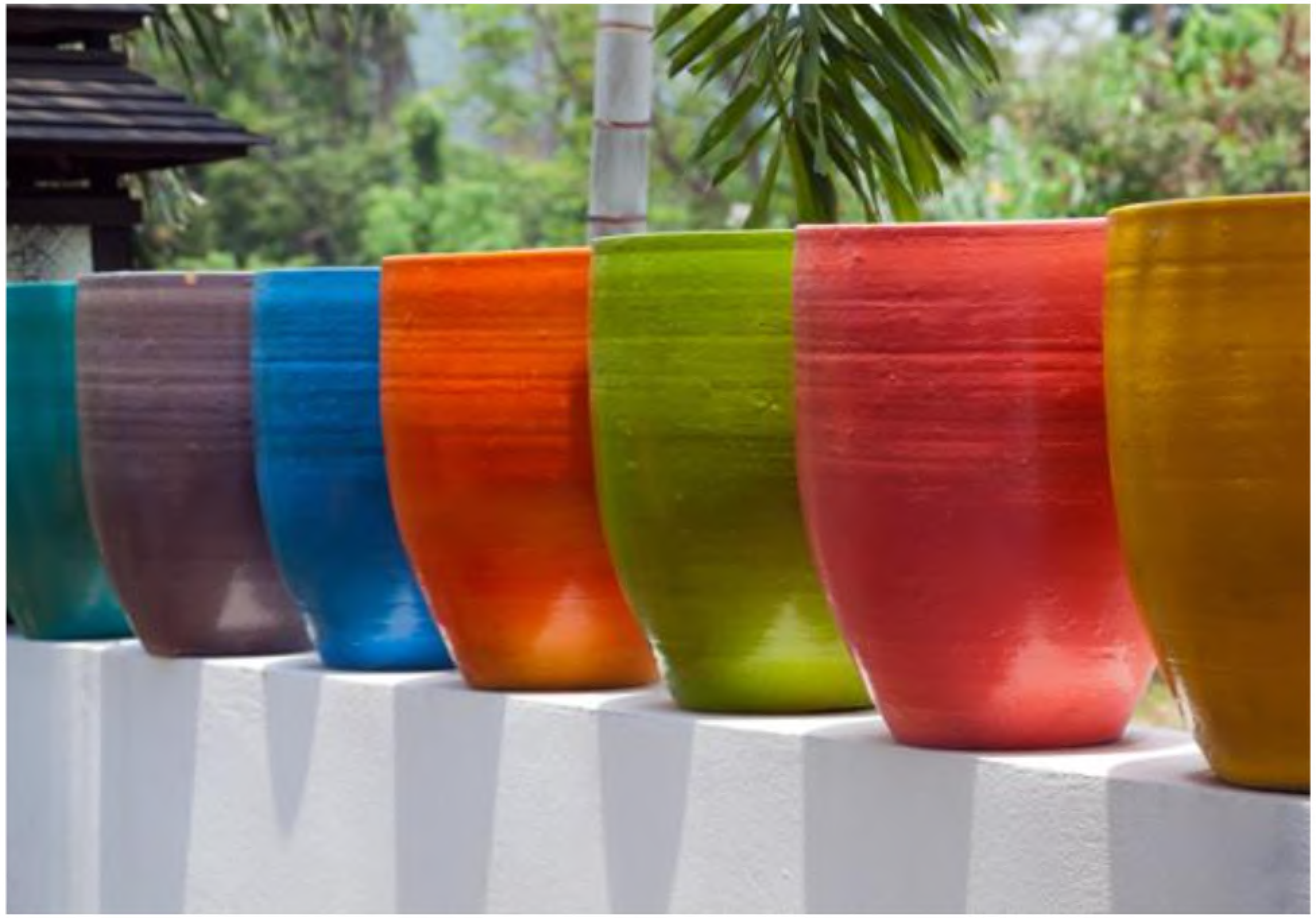


Menu

Slide Note

This image shows several beautifully - colored terracotta pots. Each pot is colored differently.

How Are These Pots Different?



Menu

00:02:00:00:47

02/ 47

Slide Note

Here, you can see a search box of a popular e-commerce shopping website. The bottom image represents the structure whereas the top image has a more aesthetic look when displaying the same search box.

1. If a plain house represents the frame and structure, what is required to get a beautifully-painted house with artwork and fixtures?

2. If a plain flowerpot represents the outer structure, then how can we get a decorated flowerpot?

3. Similarly, if a simple website is created with only HTML elements that represent the frame and structure, then how do we change it to a more appealing website, which has a better user experience and performance?

The answer for all these is STYLING.

Styling in web pages can be achieved through CSS (Cascading Style Sheets).

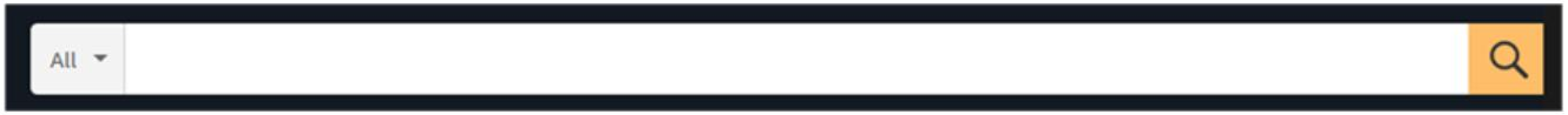
You can understand this concept through following analogies:

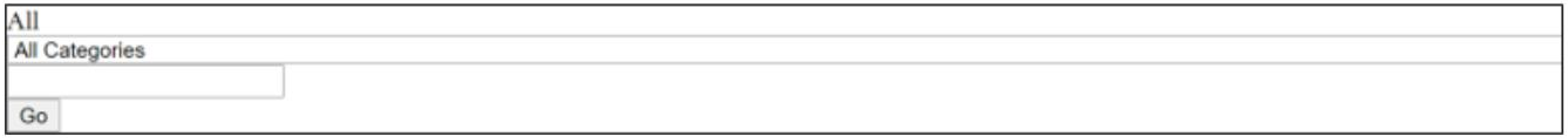
Bricks, cement, sand, wood → HTML5 elements

Doors, windows, nameplate → Semantic elements

Paints, decorative accessories → CSS properties

Which Image Looks More Appealing and Why?



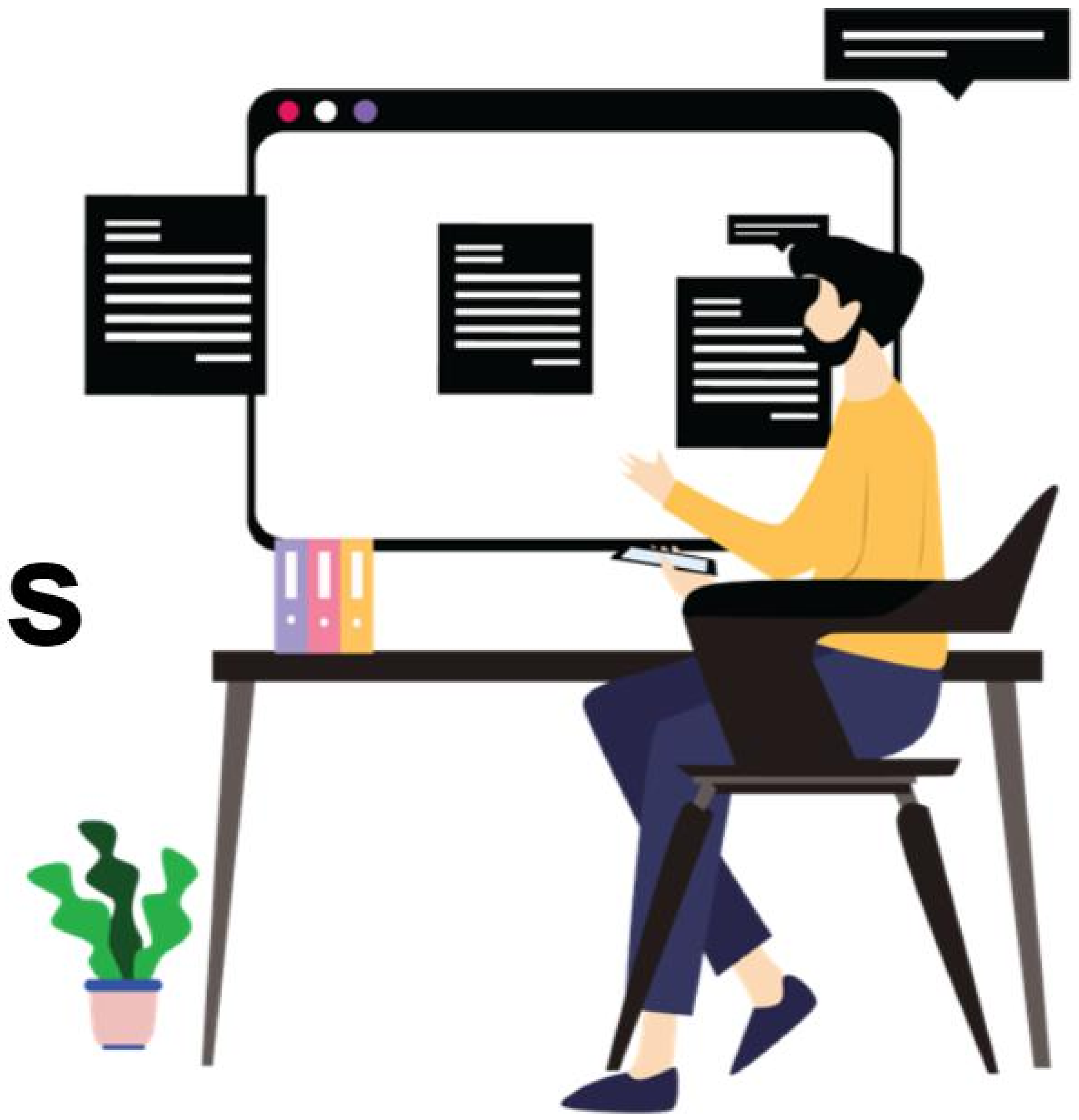


Menu

00:03 00:00:47 03/ 47

Slide Note

Style a Web Page Using CSS Properties and CSS Box Model



Menu

00:04 00:00:47 04/ 47

Slide Note

Learning Objectives

- Use CSS to style an HTML page
- Select HTML elements using CSS selectors
- Apply styles using CSS properties
- Create layouts by using the CSS box model
- Differentiate between block and inline element

Menu

00:05 00:00:47 05/ 47

Slide Note

How can you style web pages to make them look appealing?

Menu

00:06 00:00:47 06/ 47

Slide Note

Cascading Style Sheets(CSS)

Menu

00:07:00:00:47 07/ 47

Slide Note

Cascading in CSS refers to the fact that styling rules “cascade” down from several sources. This means that CSS has an inherent hierarchy and styles of higher precedence can overwrite rules of lower precedence.

• A CSS rule set consists of a selector and a declaration block.

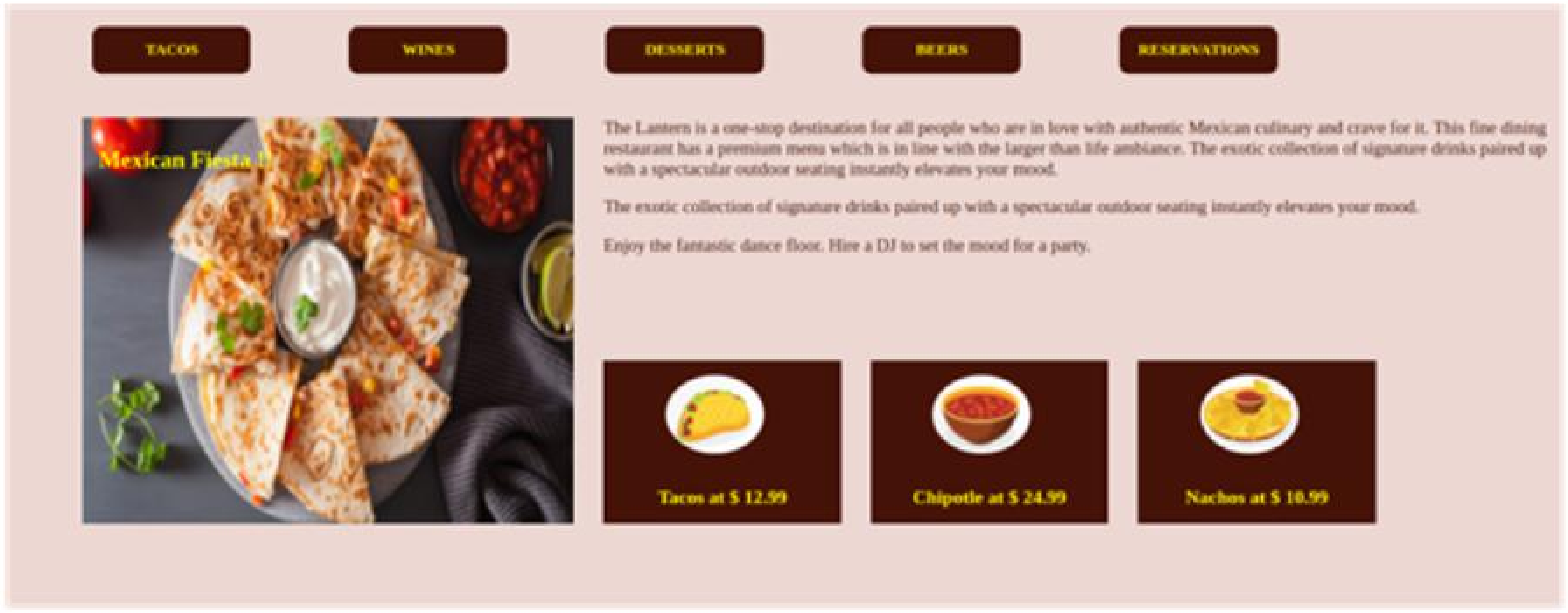
• A selector points to the HTML element you want to style.

• The declaration block contains one or more declarations separated by semicolons.

• Each declaration includes a property name and a value, separated by a colon.

What Is CSS (Cascading Style Sheet)?

CSS defines how HTML elements are to be displayed using certain rules.



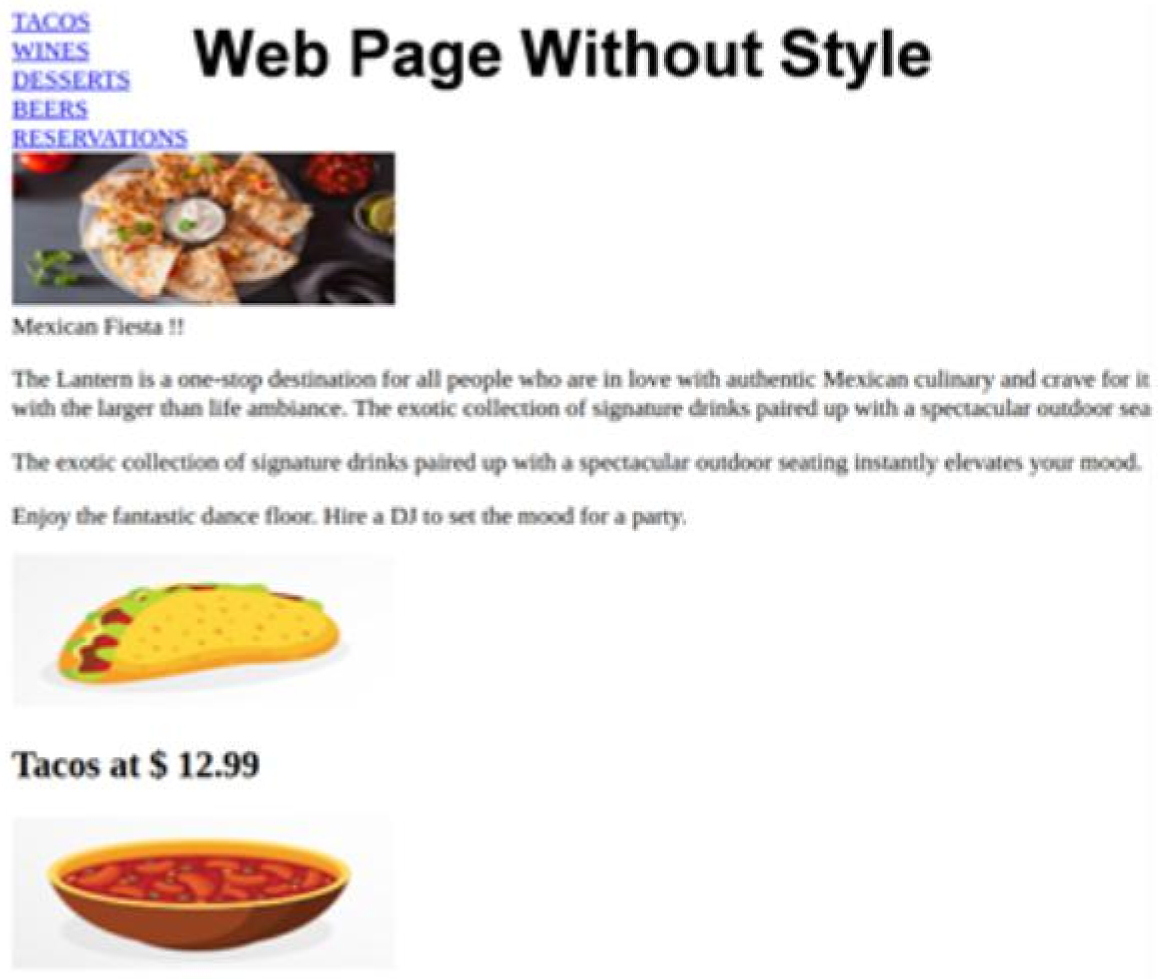
The Lantern is a one-stop destination for all people who are in love with authentic Mexican culinary and crave for it. This fine dining restaurant has a premium menu which is in line with the larger than life ambiance. The exotic collection of signature drinks paired up with a spectacular outdoor seating instantly elevates your mood.

The exotic collection of signature drinks paired up with a spectacular outdoor seating instantly elevates your mood.

Enjoy the fantastic dance floor. Hire a DJ to set the mood for a party.

Web Page With Style

Web Page Without Style



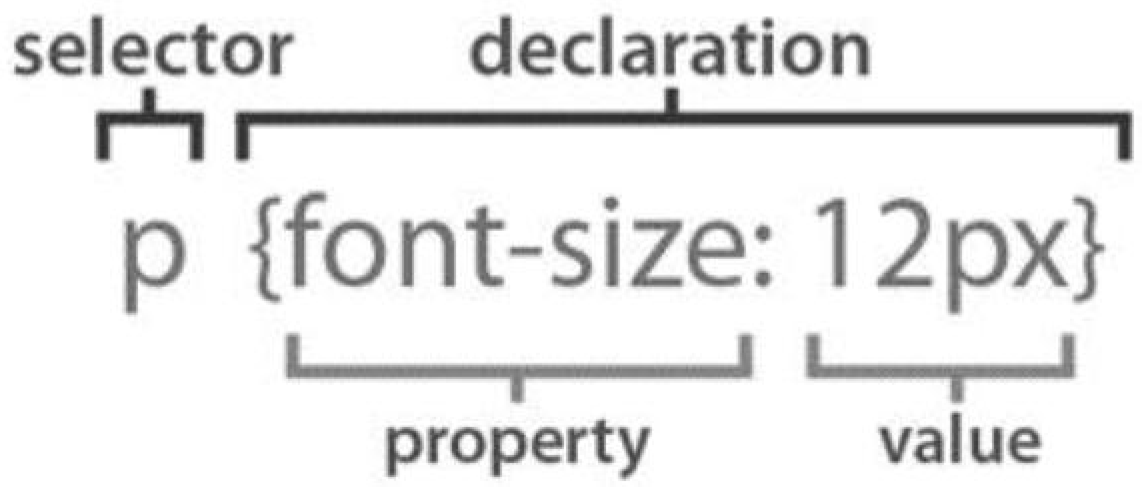
The Lantern is a one-stop destination for all people who are in love with authentic Mexican culinary and crave for it with the larger than life ambiance. The exotic collection of signature drinks paired up with a spectacular outdoor sea

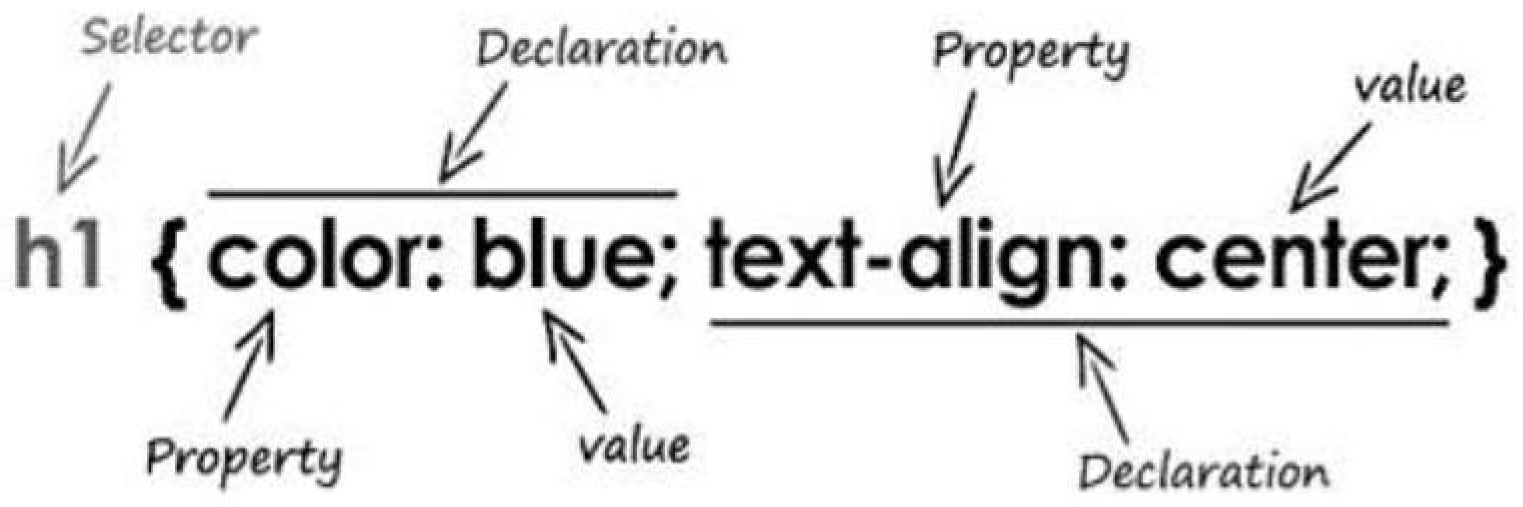
The exotic collection of signature drinks paired up with a spectacular outdoor seating instantly elevates your mood.

Enjoy the fantastic dance floor. Hire a DJ to set the mood for a party.

CSS Syntax

- A CSS rule set consists of a selector and a declaration block.
- A selector points to the HTML element that you want to style.
- A declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.





Slide Note

Applying Styles to an HTML Page

- **Inline Style**

  Inline styles are placed directly inside an HTML element in the code using the "style" attribute.

- **Embedded or Internal Style**

  Internal styles are placed inside the head section of a particular web page via the style tag.

- **External Style**

  An external style sheet is a separate page that is then linked to the web page.

Menu

Slide Note

Apply CSS to an HTML Page

Apply styles to an HTML page in different ways. Click on Types of styling to learn about the different styles.

DEMO

Menu

00:11:00:00:47 11/ 47

Slide Note

Inline styles are applied using the "style" attribute for any element.

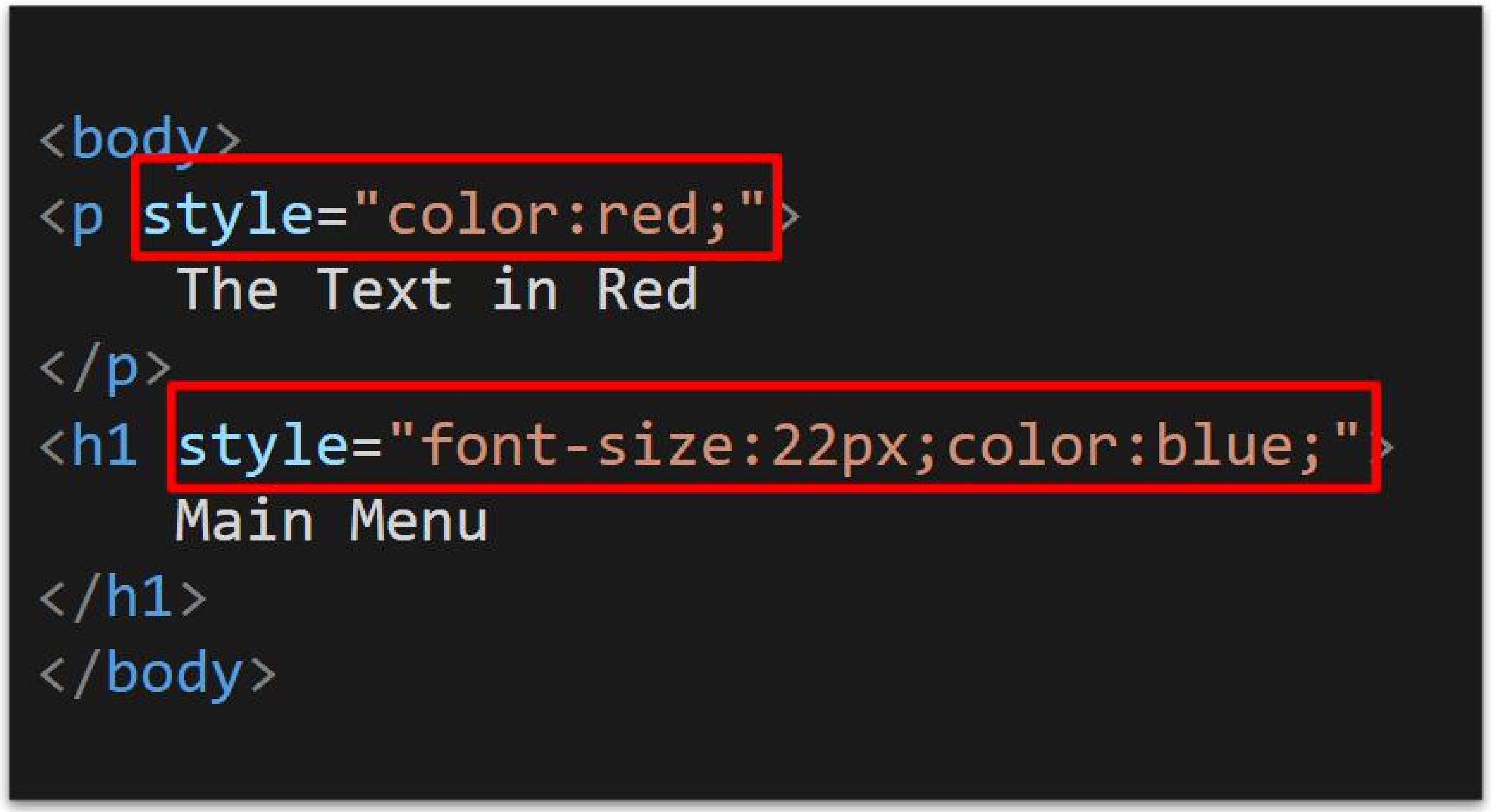
A style attribute is part of the Global Attributes and can be used on any HTML element.

It overrides any style set globally, e.g., styles specified in the

Note: It is recommended for styles to be defined in a separate file or files than using them in inline styles.

Press Esc to exit full screen

Applying Inline Styles



```
<body>
<p style="color:red;">
    The Text in Red
</p>
<h1 style="font-size:22px;color:blue;">
    Main Menu
</h1>
</body>
```

Menu

00:12 00:00:47 12/ 47

Slide Note

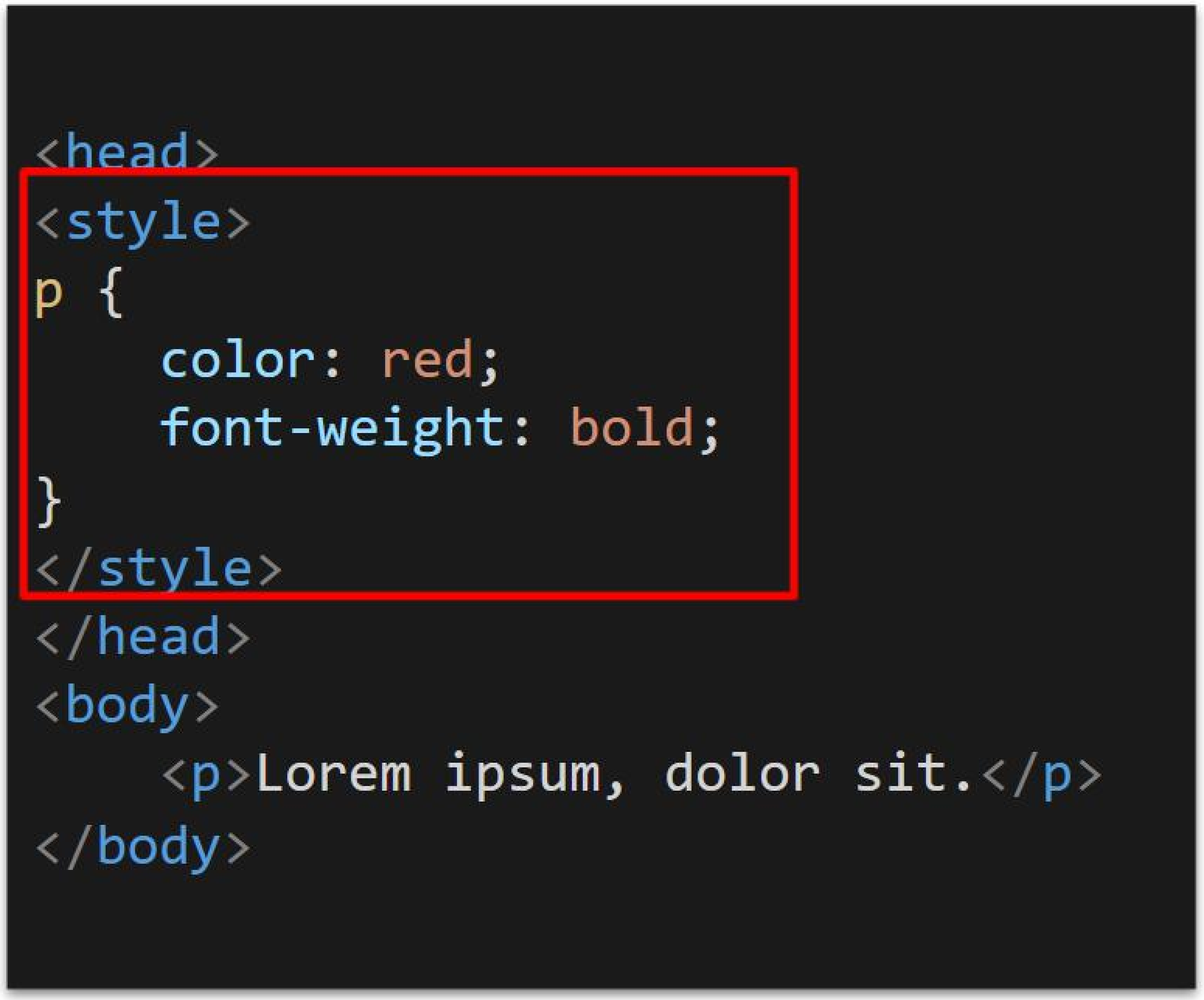
The

Inside the

Styles are getting applied with the given CSS rules to the elements in the document that are selected.

Inline styles get applied only to a particular element, where as embedded styles are more reusable since it can get applied to multiple elements across the web page or document.

Applying Embedded Styles



```
<head>
<style>
p {
    color: red;
    font-weight: bold;
}
</style>
</head>
<body>
    <p>Lorem ipsum, dolor sit.</p>
</body>
```

Menu

00:13 00:00:47 13/ 47

Slide Note

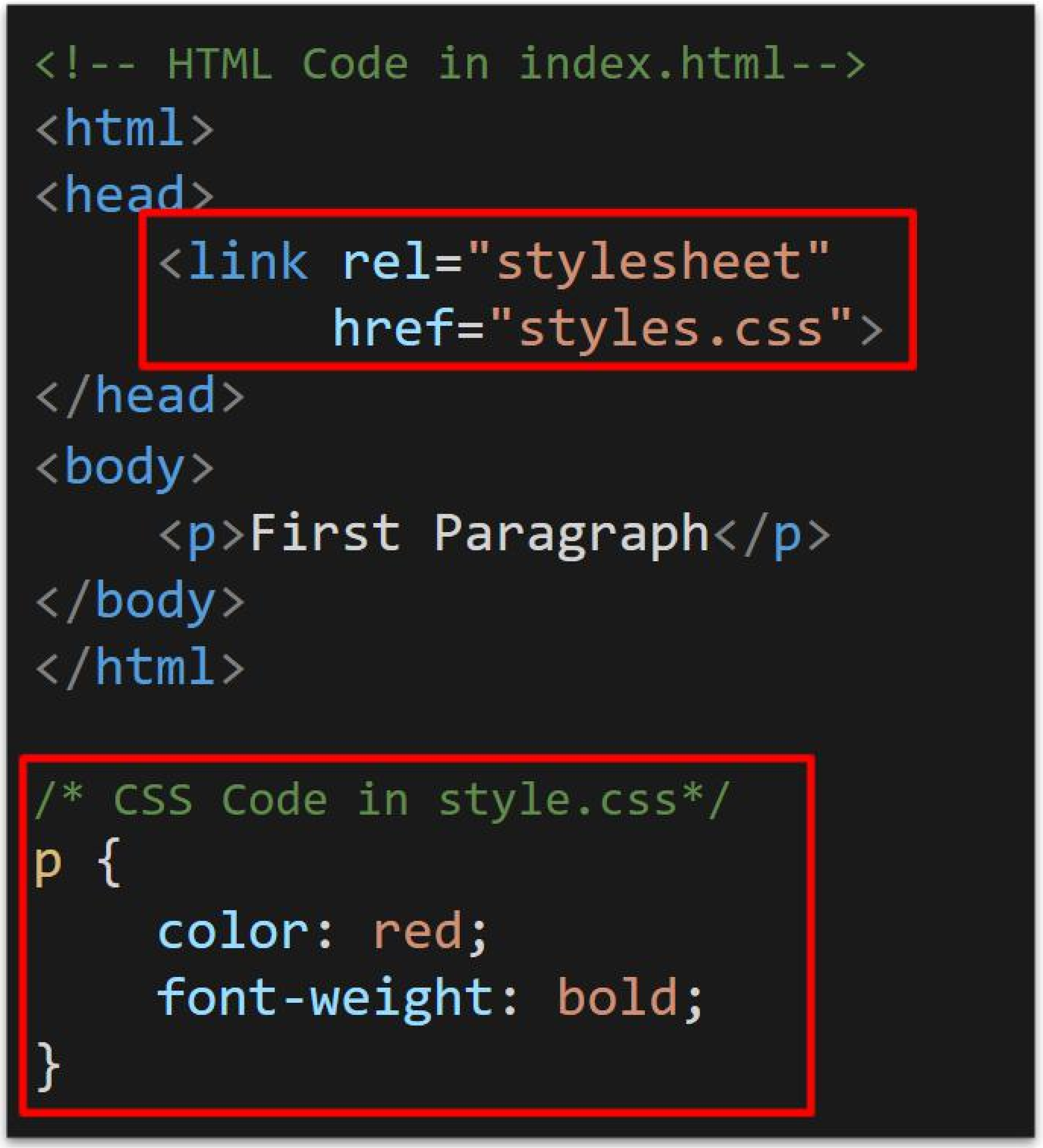
With external styles, same styles can be applied to multiple pages having common styles just by linking the .css file to that HTML document.

A single change in the stylesheet can be applied to all the linked pages if you need to make widespread changes in the web design. This saves a lot of time and effort.

Hence, external styles are more reusable and give a consistent look and feel to a website.

Applying External Styles

1. Create an HTML document with the **.html** extension and add a paragraph element.
2. Create a file in the same folder as the HTML document with **.css extensionand add a CSS rule to** that paragraph element.
3. Link the HTML and CSS document using the **<link>** tag which comes under the **<head>** tag.



```
<!-- HTML Code in index.html-->
<html>
<head>
    <link rel="stylesheet"
          href="styles.css">
</head>
<body>
    <p>First Paragraph</p>
</body>
</html>

/* CSS Code in style.css*/
p {
    color: red;
    font-weight: bold;
}
```

Menu

00:14:00:00:47 14/ 47

Slide Note

External style sheets are recommended since they are reusable across multiple web pages.

Menu

00:15:00:00:47

15/ 47

Slide Note

HTML Elements Using CSS Selector

Menu

00:16:00:00:47 16/ 47

Slide Note

CSS Selectors

- CSS selectors allow you to select and manipulate HTML elements.
- They are used to find HTML elements based on their ID, Class, Type, and Attribute .
- Commonly used CSS selectors are:
  - Type Selectors
  - ID Selectors
  - Class Selectors

Menu

00:17:00:00:47 17/ 47

Slide Note

CSS rules declared under CSS code are applied to all div and span elements.

Press Esc to exit full screen

Type Selectors

A type selector is sometimes referred to as a tag name selector or element selector because it selects an HTML tag/element in your document.

```
<!-- HTML Code -->
<div>
    Intel unveils 11th Gen H series processors forg
    aming laptops: Details here
</div>
<div>
    <span>
        One of the major differences between the
        existing 11th Gen Intel Core processors
        (Tiger Lake U) and the newly unveiled ones(
        Tiger Lake H) is in the integrated
        graphic processing unit
    </span>
</div>
<span>11th Gen Intel Core H series processor</span>
```

```
/* CSS Code */
div { width: 100px; }
span { font-size:  16px; }
```

Menu

Slide Note

CSS rules declared under CSS code are applied only to this element since the ID is unique.
ID selectors are used when you want to apply styles for a particular element.

ID Selectors

- An ID selector uses the ID attribute of an HTML element to select a specific element.
- The value for the ID attribute is unique for the whole HTML document.
- To select an element with a specific ID, write a hash(#) character, followed by the ID of that element.

```html
<!-- HTML Code -->
<div id="headline">
   Intel unveils 11th Gen H series
   processors for gaming laptops: Details here
</div>
```

```css
/* CSS Code */
#headline { color: blue; }
```

Menu

00:20 00:00:47 20/ 47

Slide Note

CSS rules declared under CSS code would be applied to both div and span elements.

The class selector is used for selecting a single or a group of HTML elements with the same class attribute value.

Class Selectors

- The Class selector selects elements with a specific class attribute.

```html
<!-- HTML Code -->
<div class="headline">
   One of the major differences between
   the existing 11th Gen Intel Core
   processors (Tiger Lake U) and the newly
   unveiled ones (Tiger Lake H) is in the
   integrated graphic processing unit
</div>
<span class="headline">
   11th Gen Intel Core H series processor
</span>
```

```css
/* CSS Code */
.headline { color: blue; }
```

- To select elements with a specific class, write a period(.) character, followed by the name of the class.

Menu

00:21 00:00:47 21/ 47

Slide Note

Applying Styles to HTML Elements Using CSS Properties

Menu

00:22 00:00:47 22/ 47

Slide Note

MDN links add

Background-color: With CSS, a color is most often specified by:

a valid color name - like "green"

a HEX value - like "#ff0000"

an RGB value - like "rgb(255,255,0)"

The opacity property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0. The lower the value, the more transparent.it is.

When using the opacity property to add transparency to the background of an element, all its child elements inherit the same transparency. This can make the text inside a fully transparent element hard to read.

Transparency Using RGBA:

An RGBA color value is specified with: rgba(red, green, blue, *alpha*). The *alpha* parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

Background-image:

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

Background-repeat:

The background-image property repeats an image both horizontally and vertically.

To repeat an image vertically, set background-repeat: repeat-y;

Showing the background image only once is also specified by the background-repeat property.

Background-position:

The background-position property is used to specify the position of the background image.

Background-attachment:

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).

CSS Background Property

Background property is shorthand for the following CSS properties:

- background-color
- background-image
- background-position
- background-repeat

Click here for all the background properties and its various values.

```
body {
    background-color: whitesmoke;
    background-image: url('./bg.png');
    background-repeat: no-repeat;
    background-position: 10px;
   }
/*shorthand property*/
P {
   background: gray url('./icon.png')
no-repeat scroll left;
  }
```

Slide Note

Control the appearance of a web page by applying styles using CSS properties.

1. Apply CSS background and font shorthand properties to paragraph text and headings.

2. Apply list style properties to ordered and unordered lists.

Press Esc to exit full screen

Apply Styles Using CSS Properties

Control the appearance of a web page by applying styles using CSS properties. Click on Basic CSS properties for the demo codes.

DEMO

Menu

00:24 00:00:47 24/ 47

Slide Note

Font-family:

In general, there are five font-families:

Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.

Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.

Monospace fonts - here all the letters have the same fixed width. They create a mechanical look.

Cursive fonts imitate human handwriting.

Fantasy fonts are decorative/playful fonts.

The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers and operating systems. Start with the font you want, and end with the generic family.

Font-size:

The font-size property sets the size of the text.

The font-size value can be an absolute, or of relative size.

Setting the text size with pixels or with em (1em=16px)

```
h1 {
  font-size: 40px;
}
h1 {
  font-size: 2.5em; /* 40px/16=2.5em */
}
```

Font-style:

The font-style property is mostly used to specify italic text.

This property has three values:

normal - The text is shown normally.

italic - The text is shown in italics.

oblique - The text is "leaning".

CSS Font Property

Font property is shorthand for the following CSS properties:

- font-family
- font-size
- font-style
- font-weight
- line-height

Refer here for all the font properties and various values

```
p {
    font-family: 'Courier New';
    font-style: italic;
    font-size: large;
    font-weight: 700;
    line-height: 1.15px;
  }
/*Shorthand Property*/
h1 {
    font: bold italic 2em cursive;
  }
```

Slide Note

List style type:

Specifies the type of list-item marker. Default value is "disc."

List style position:

Specifies where to place the list-item marker. Default value is "outside."

List style image:

Specifies the type of list-item marker. Default value is "none."

CSS List-style Property

List-style property is shorthand for the following CSS properties:

- list-style-position
- list-style-type

Click here for all the list-style properties and its various values.

```
ul {
    list-style-type: square;
    list-style-position : outside;
}
ol {
    list-style-type: lower-roman;
    list-style-position : inside;
}
/*Shorthand Property*/
ul{
    list-style: square outside;
}
```

Menu

Slide Note

Quick Check

Which one of the following is the correct (and most specific) CSS selector statement used to select the text inside an HTML element, and to change the text color to red?

<p> Some text content </p>

a) p { color : red; }

b) #p { text-color : red; }

c) $p { color : red; }

d) p { font-color : red }

Menu

00:27 00:00:47 27/ 47

Slide Note

Quick Check: Solution

Which is the correct (and most specific) CSS selector statement used to select the text inside an HTML element, and to change the text color to red in the following example?

<p> Some text content </p>

a) **p { color : red; }**

b) #p { text-color : red; }

c) $p { color : red;}

d) p { font-color : red }

Menu

00:28:00:00:47 28/ 47

Slide Note

What do you see?

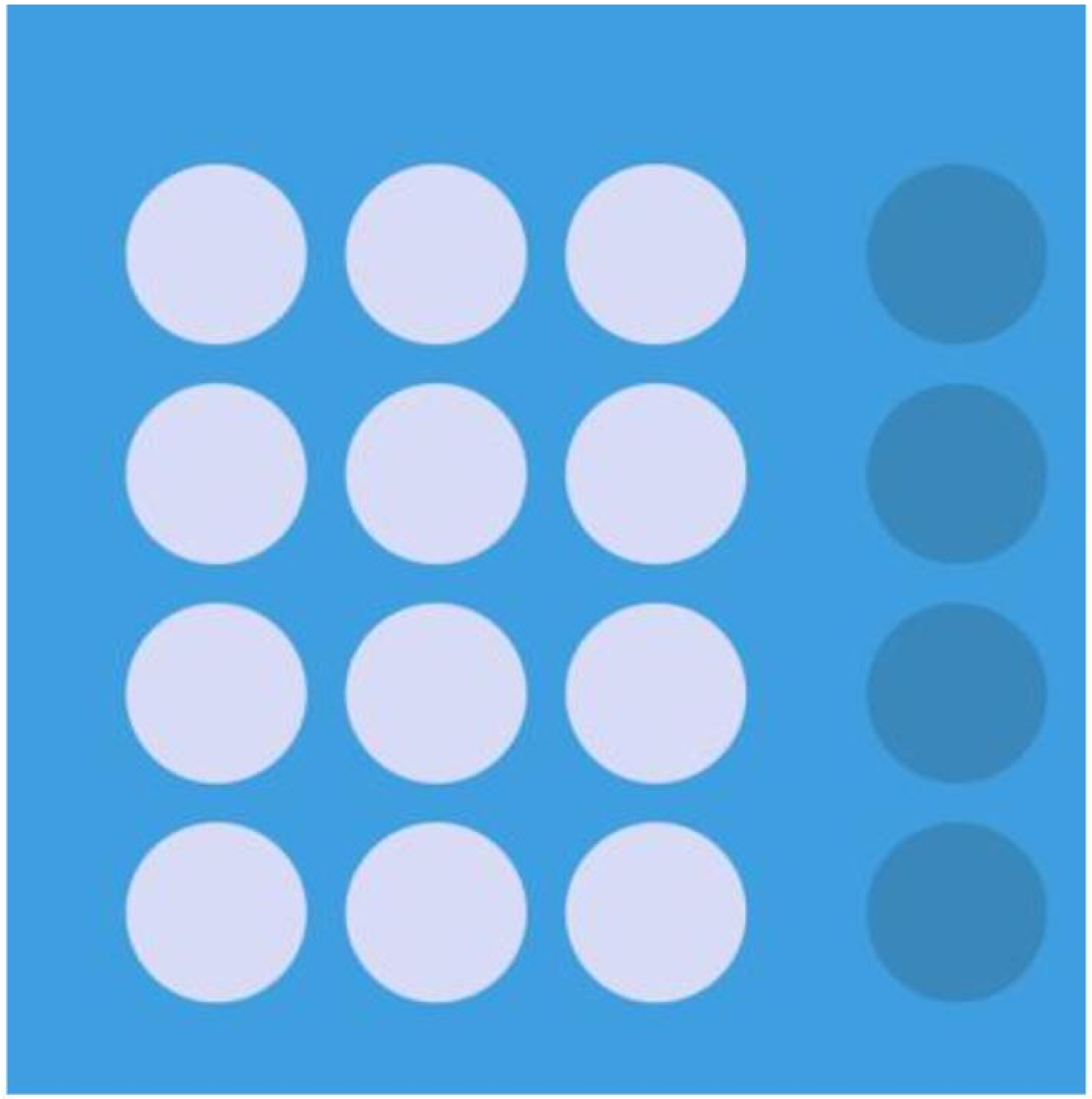
Do you see 16 circles or do you see 12 white circles and 4 other circles?

Why do you think you noticed the 12 circles first and then the 4 other circles, and not all 16 circles altogether? Is it because of the -

difference in their color or

space between the two sets

How Many Circles Do You See in This Picture?



Menu

00:29 00:00:47 29/ 47

Slide Note

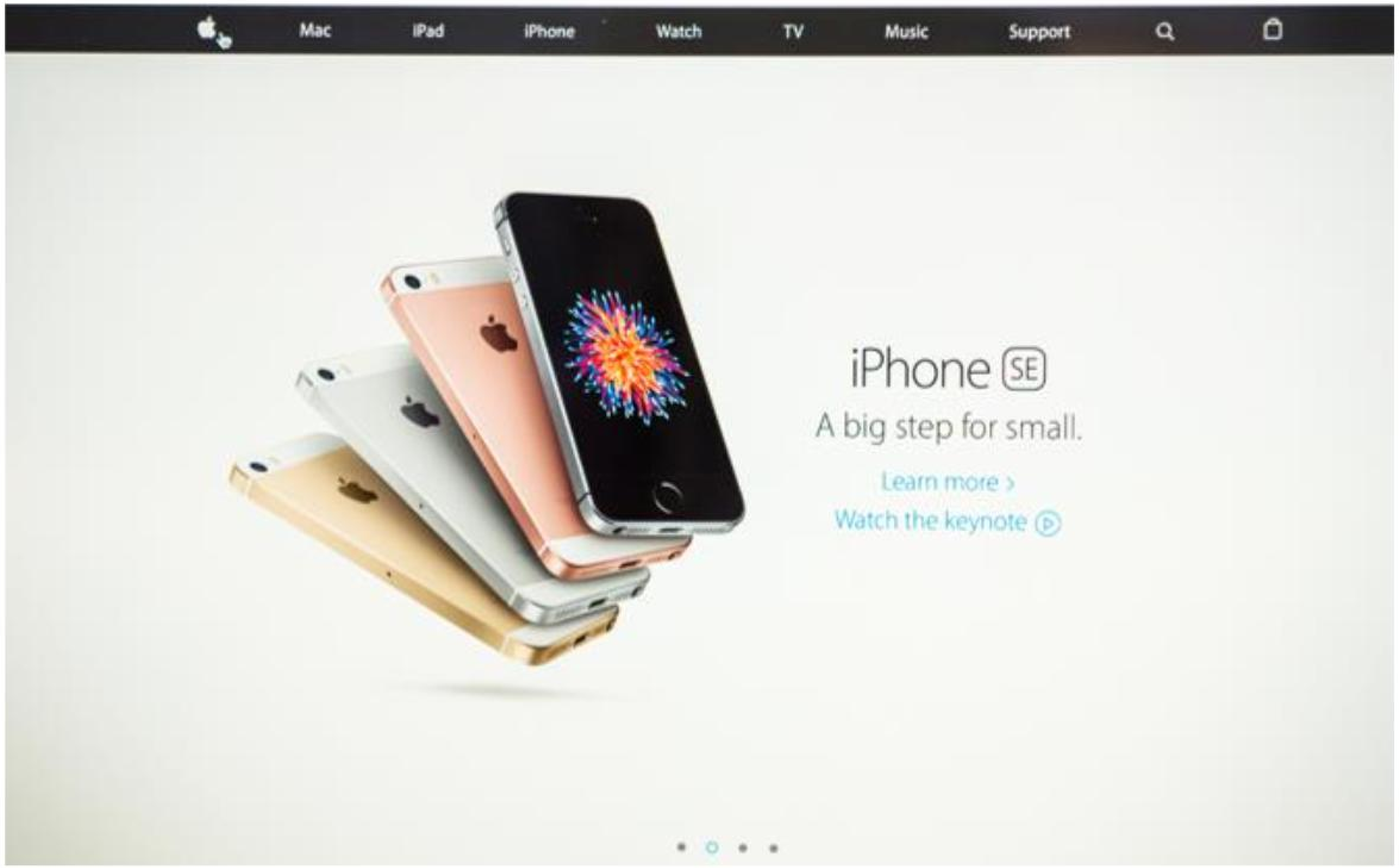
What do you see?

How many logical sections do you see?

The answer is two.

What helps you identify these sections?

The spaces between the iPhone title, description text, and the iPhone image.



What Do You See in This Image?

Menu

00:30 00:00:47 30/ 47

Slide Note

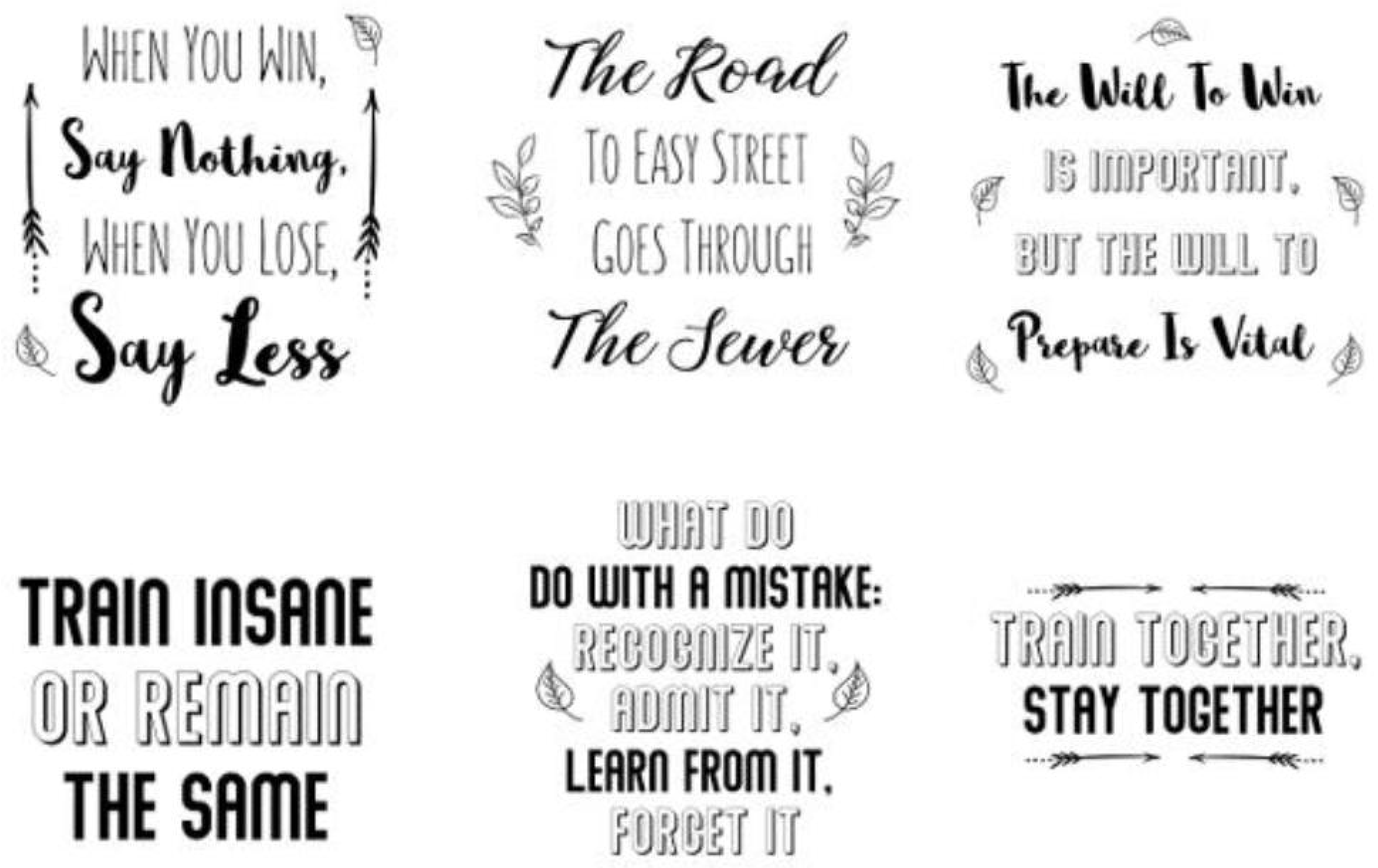
How many logical sections do you see?

6 sections

What helps you identify these sections?

The white spaces

How Many Logical Sections Has This Image Been Divided Into?


When You Win, Say Nothing, When You Lose, Say Less
The Road To Easy Street Goes Through The Sewer
The Will To Win Is Important, But The Will To Prepare Is Vital
Train Insane Or Remain The Same
What Do Do With A Mistake: Recognize It, Admit It, Learn From It, Forget It
Train Together, Stay Together

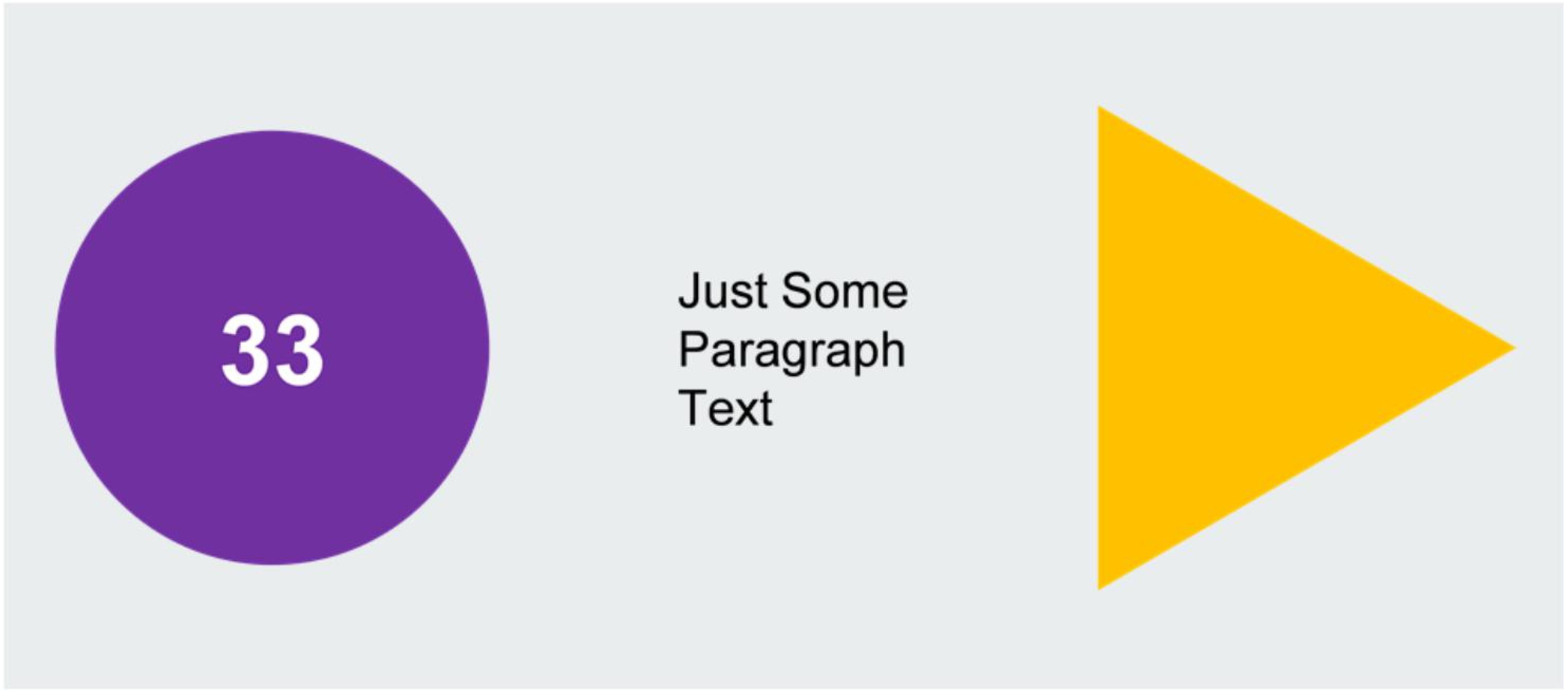
Menu

00:31 00:00:47 31/ 47

Slide Note

This image shows a part of a web page content.

Move to the next slide and see how the same image is perceived by the browser.

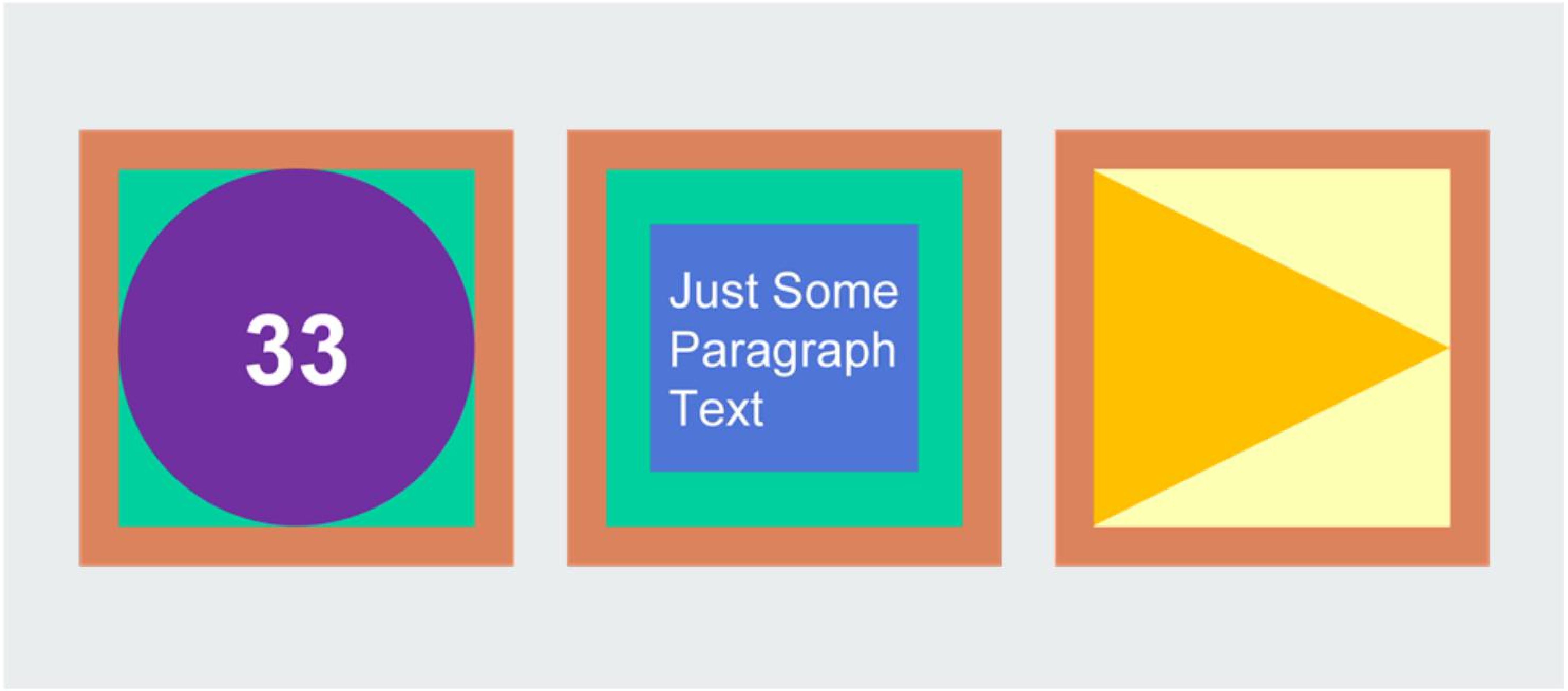


Web Page Content

Slide Note

Everything in CSS has a box around it, and understanding these boxes is key to being able to create layouts with CSS, or to align items with other items.

Press Esc to exit full screen



Web Page Content - Browser Perspective

Menu

Slide Note

Open a website like wikipedia.com.

1. Right click on the "Contents" hyperlink in the browser window

2. Click inspect menu to open chrome dev tools.

3. Select the Computed tab at the top of the rightmost column.

4. Hover over the different properties of the logo's box. The corresponding space on the web page should be highlighted when you do this.

CSS Box Model

The CSS box model is a box that wraps around every HTML element. It consists of margins, borders, padding, and the actual content.

- **Margin:** This is the outermost layer that wraps the content, with the padding and border as the white space between this box and other elements.
- **Border**: Wraps the content and the padding, if any.
- **Padding**: Sits around the content as the white space; its size can be controlled using padding and related properties.
- **Content**: The area where your content is displayed.

Menu

00:34/00:00:47 34/ 47

Slide Note

The CSS box model is essentially a box that wraps around every HTML element. It consists of margins, borders, padding, and the actual content.

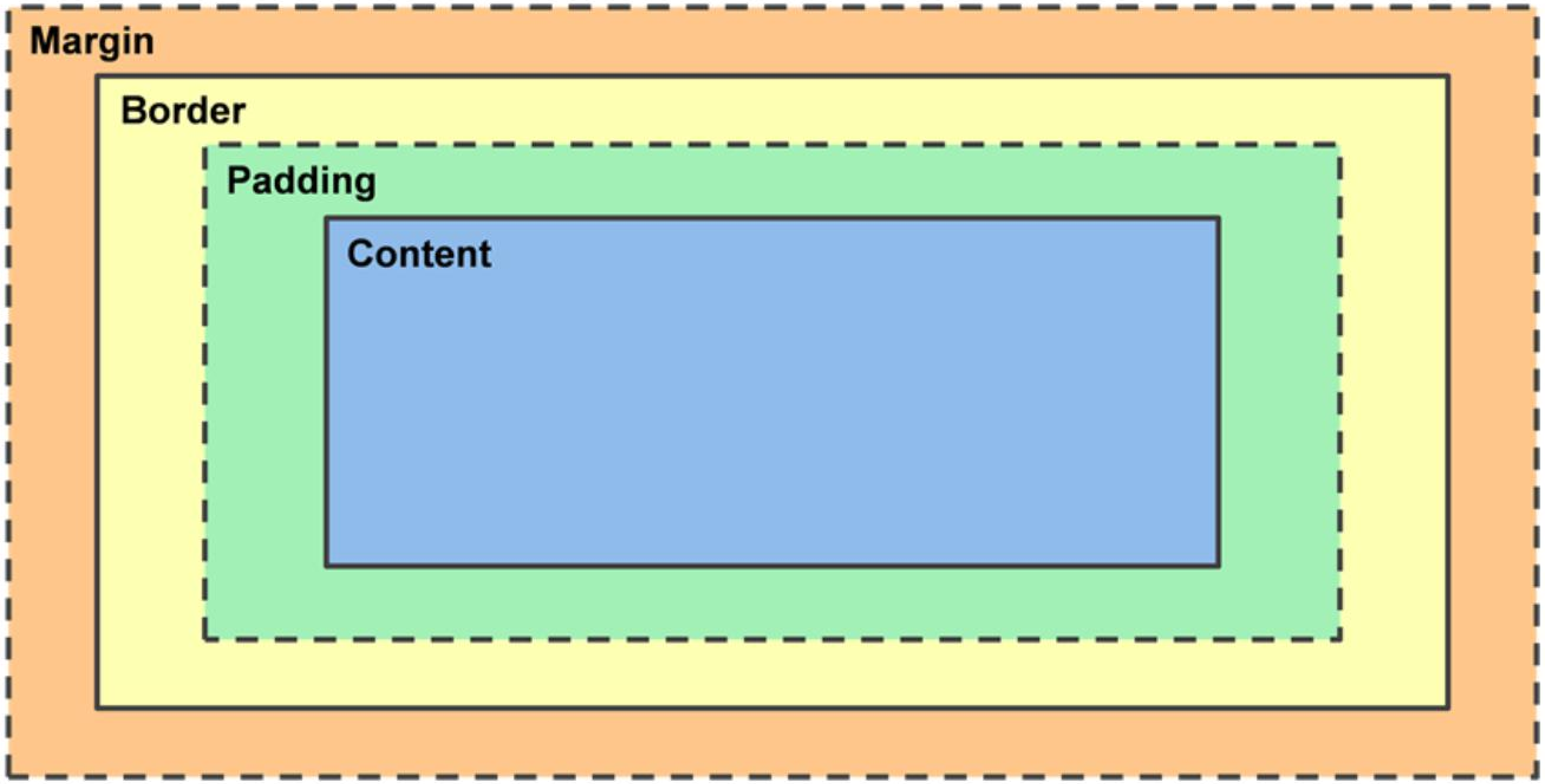
Margin: The outermost layer that wraps the content, with the padding and border as the white space between this box and other elements.

Border: Wraps the content and the padding, if any.

Padding: Sits around the content as the white space; its size can be controlled using padding and related properties.

Content: The area where your content is displayed.

CSS Box Model



Slide Note

Understand the CSS Box model properties

1. Apply different values for margin properties to the div container and content inside the container and observe the appearance.

2. Give negative values for margins to understand margin collapsing.

3. Apply padding properties to observe the appearance of the contents inside the container.

4. Apply different border properties like style, width and color to div elements.

5. Create a web page with some basic content, then add margins, borders and paddings to make the web page look appealing.

CSS Box Model

Create a web page with some basic content, then add margins, borders and paddings to make the web page look appealing.

For this, refer to the Box Model.

DEMO

Menu

00:36 00:00:47 36/ 47

Slide Note

The margin pushes other elements away from the box. Margins can have positive or negative values.

Setting a negative margin on one side of your box can cause it to overlap other things on the page.

We can control all the margins of an element at once using the margin shorthand property, or each side individually using the equivalent longhand properties:

- margin-top
- margin-right
- margin-bottom
- margin-left

Margin shorthand Property:

/* Apply to all four sides */margin: 1em;margin: -3px;/* vertical | horizontal */margin: 5% auto;/* top | horizontal | bottom */margin: 1em auto 2em;/* top | right | bottom | left */margin: 2px 1em 0 auto;/* Global values */margin: inherit;margin: initial;margin: unset;

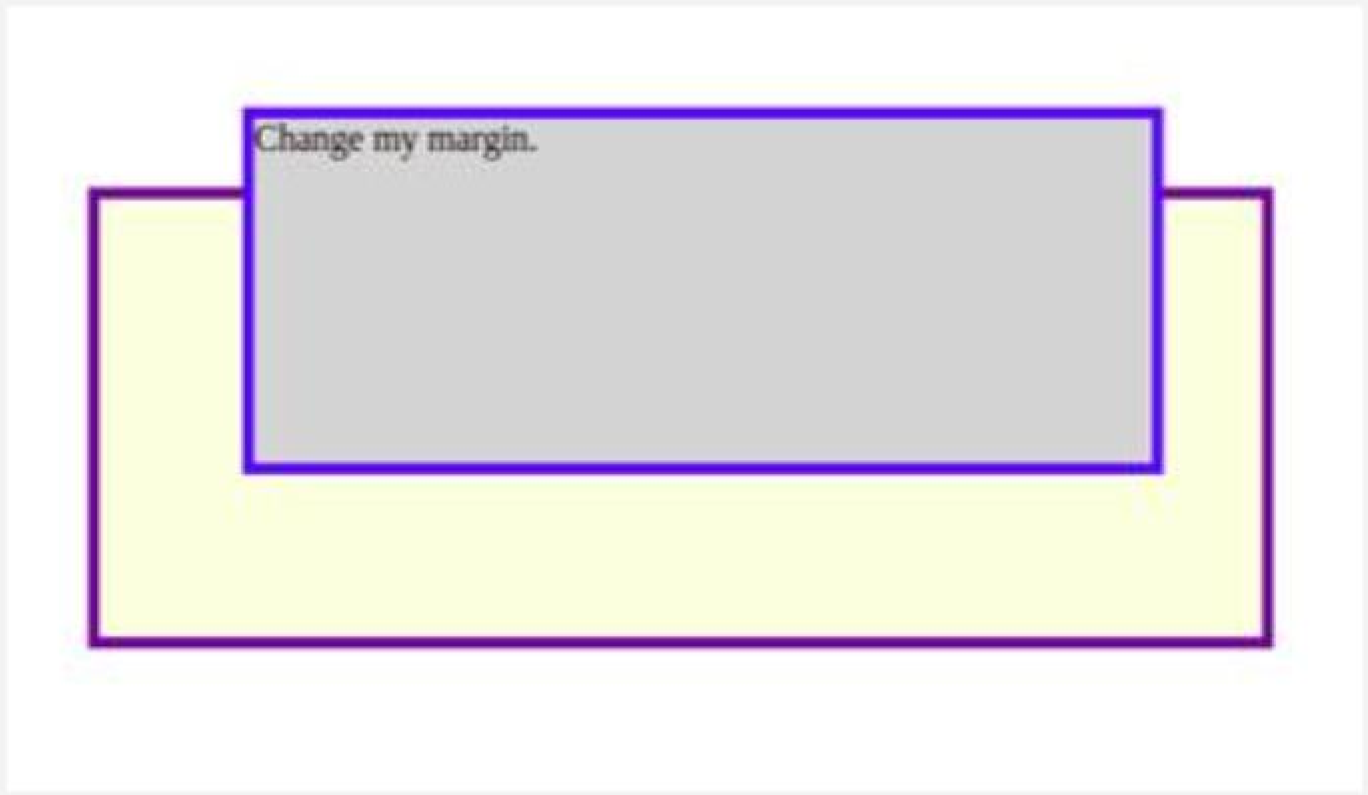
Press Esc to exit full screen

Margin

- The margin is an invisible space around the box.
- It is used to create a space around the elements and outside a defined border.

```css
.box {
  margin-top: -40px;
  margin-right: 30px;
  margin-bottom: 40px;
  margin-left: 4em;
}
```

```html
<div class="container">
    <div class="box">
        Change my margin.
    </div>
</div>
```



Menu

00:37 00:00:47

37/ 47

Slide Note

If there are two elements whose margins touch, and both margins are positive, then those margins will combine to acts as one margin, and is the size of the largest individual margin. If one or both margins are negative, the amount of negative value will subtract from the total.

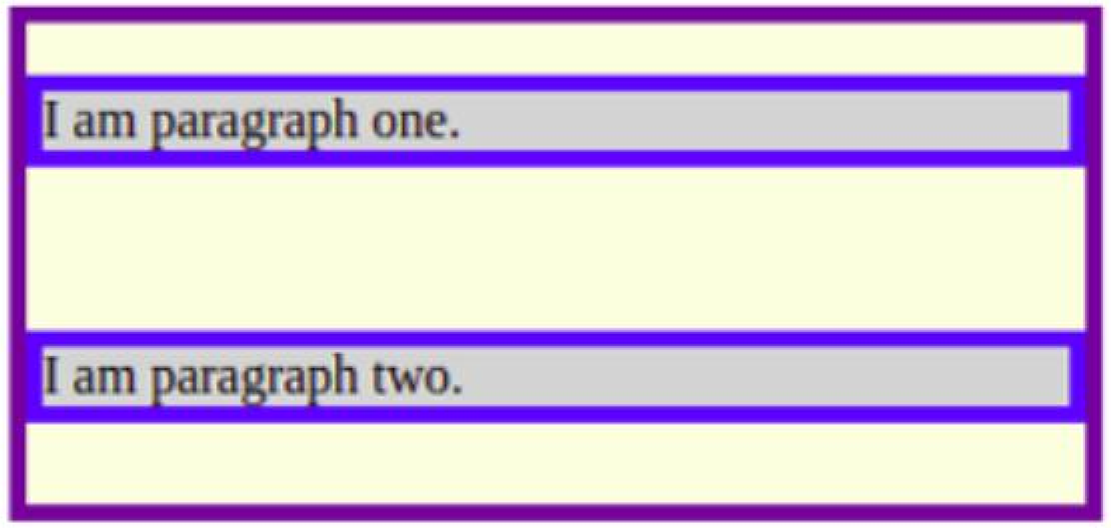
In the example below, we have two paragraphs. The top paragraph has a margin-bottom of 50 pixels. The second paragraph has a margin-top of 30 pixels. The margins have collapsed together, so the actual margin between the boxes is 50 pixels and is not the total of the two margins.

Margin Collapsing

Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.

```css
.one {
    margin-bottom: 50px;
}
.two {
    margin-top: 30px;
}
```

```html
<div class="container">
    <p class="one">
      I am paragraph one.
    </p>
    <p class="two">
      I am paragraph two.
    </p>
</div>
```



Menu

00:38 00:00:47 38/ 47

Slide Note

If you are using the standard box model, the size of the border is added to the width and height of the box.

If you are using the alternative box model, then the size of the border makes the content box smaller as it takes up some of that available width and height.

For styling borders, there are many properties — there are four borders, and each border has a style, width and color that we might want to manipulate.

You can set the width, style, or color of all the four borders at once using the border property.

To set the properties of each side individually, you can use:

border-top

border-right

border-bottom

border-left

To set the width, style, or color of all sides, use the following:

border-width

border-style

border-color

To set the width, style, or color of a single side, you can use one of the most granular longhand properties:

border-top-width

border-top-style

border-top-color

border-right-width

border-right-style

border-right-color

border-bottom-width

border-bottom-style

border-bottom-color

border-left-width

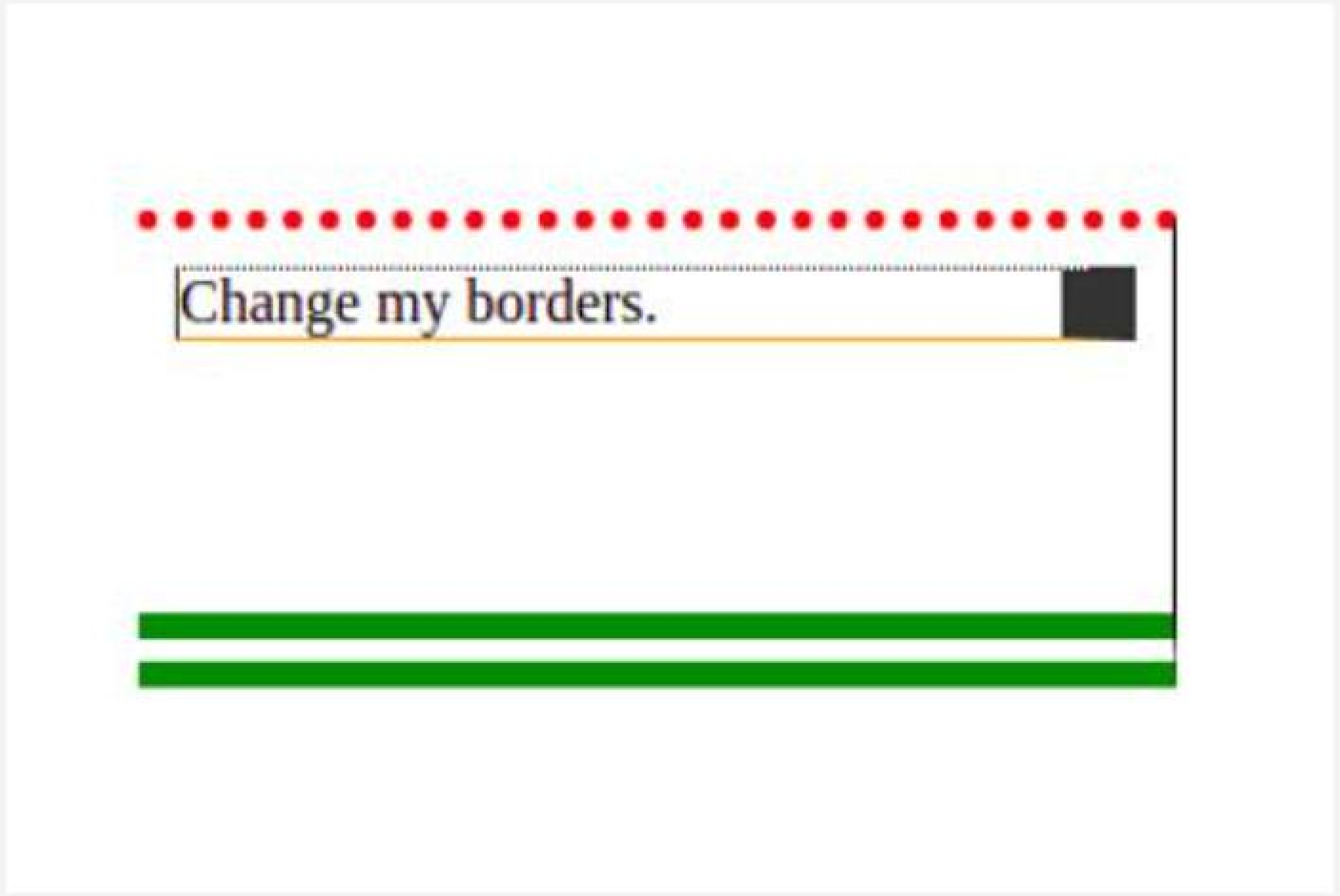
border-left-style

border-left-color

Border

- A border is drawn between the margin and the padding of a box.
- Border property allows you to specify the style, width, and color of an element's border.

```
.container {
    border-top: 5px dotted red;
    border-right: 1px solid black;
    border-bottom: 20px double
green;
}
.box {
    border: 1px solid #333333;
    border-top-style: dotted;
    border-right-width: 20px;
    border-bottom-color: orange;
}
```



Menu

00:39 00:00:47 39/ 47

Slide Note

Unlike margins, you cannot have negative amounts of padding, so the value must be 0 or a positive value.

Any background applied to your element will be displayed behind the padding. It is typically used to push the content away from the border.

We can control the padding on each side of an element individually using the padding property, or on each side individually using the equivalent longhand properties:

padding-top

padding-right

padding-bottom

padding-left

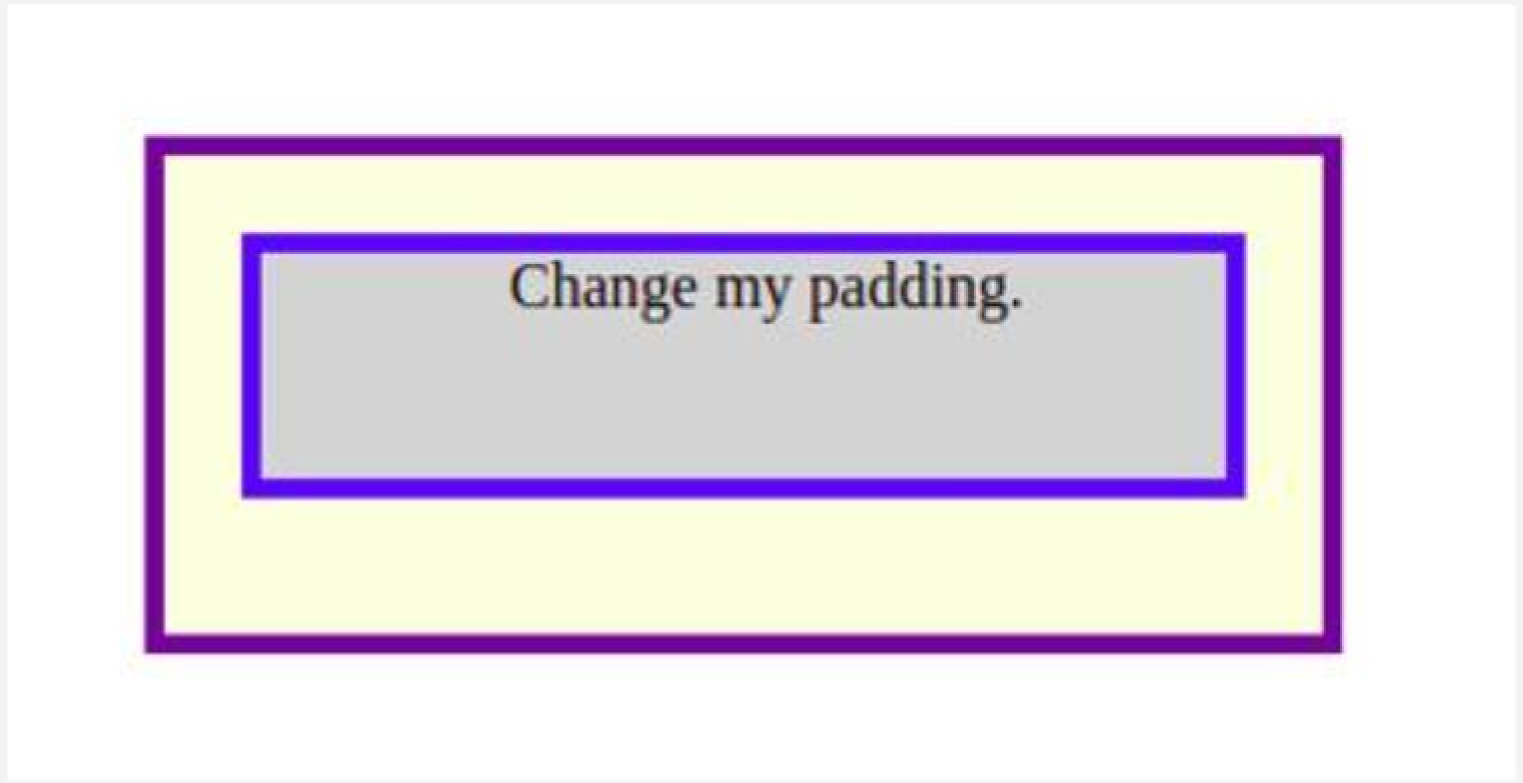
```
padding: 5%;                /* All sides: 5%
padding */padding: 10px;              /* All
sides: 10px padding */padding: 10px
20px;        /* top and bottom: 10px
padding */                                /*
left and right: 20px padding */padding:
10px 3% 20px;     /* top:           10px
padding */                                   /*
left and right: 3% padding   */
                         /* bottom:        20px
padding */padding: 1em 3px 30px 5px;
    /* top:    1em padding  */
                            /* right:  3px
padding  */
     /* bottom: 30px padding */
                               /* left:   5px
padding  */
```

Padding

- Padding sits between the border and the content area.
- It is used to create space around an element's content inside the defined borders.

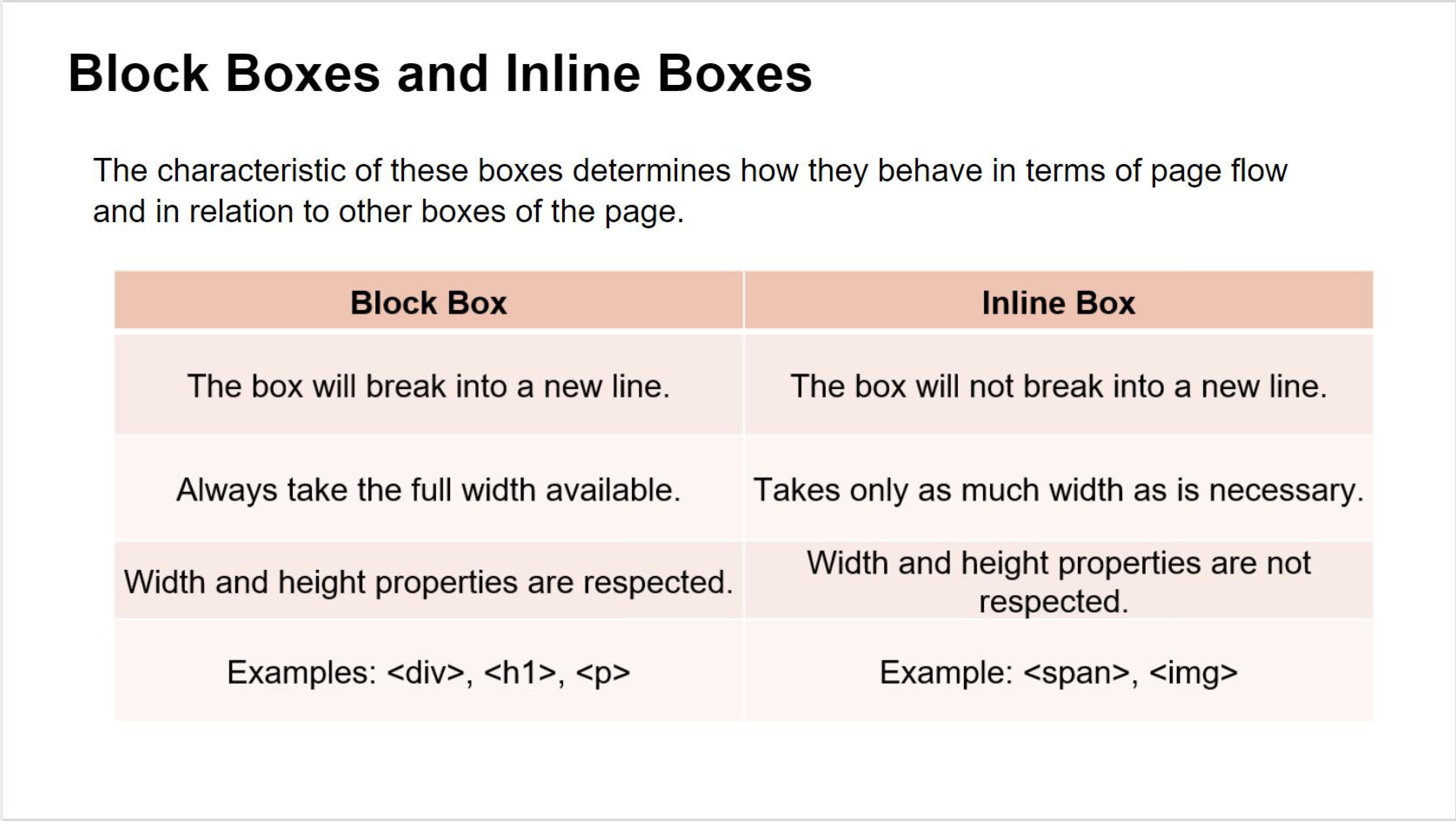
```css
.box {
    padding-top: 0;
    padding-right: 30px;
    padding-bottom: 40px;
    padding-left: 4em;
}
```

```html
<div class="container">
    <div class="box">
        Change my padding.
    </div>
</div>
```



Menu

00:40 00:00:47 40/ 47



Block Boxes and Inline Boxes

The characteristic of these boxes determines how they behave in terms of page flow and in relation to other boxes of the page.

Block Box	**Inline Box**
The box will break into a new line.	The box will not break into a new line.
Always take the full width available.	Takes only as much width as is necessary.
Width and height properties are respected.	Width and height properties are not respected.
Examples: <div>, <h1>, <p>	Example: <span>, <img>

Slide Note

Display with inline-block demo code:

In the code, we have a inside a paragraph and have applied a width, height, margin, border, and padding to it. You can see that the width and height are ignored. The vertical margin, padding, and border are respected but they do not change the relationship of other content to our inline box and so the padding and border overlaps other words in the paragraph. Horizontal padding, margins, and borders are respected and will cause other content to move away from the box.

Try changing display: inline-block to our element. Then change this to display block or remove the line completely to see the difference in the display models.

CSS Display Property

Let's apply different values to the display property to see how the content layout changes. Click on the link Block and Inline elements to do so.

DEMO

Menu

00:42 00:00:47 42/ 47

Slide Note

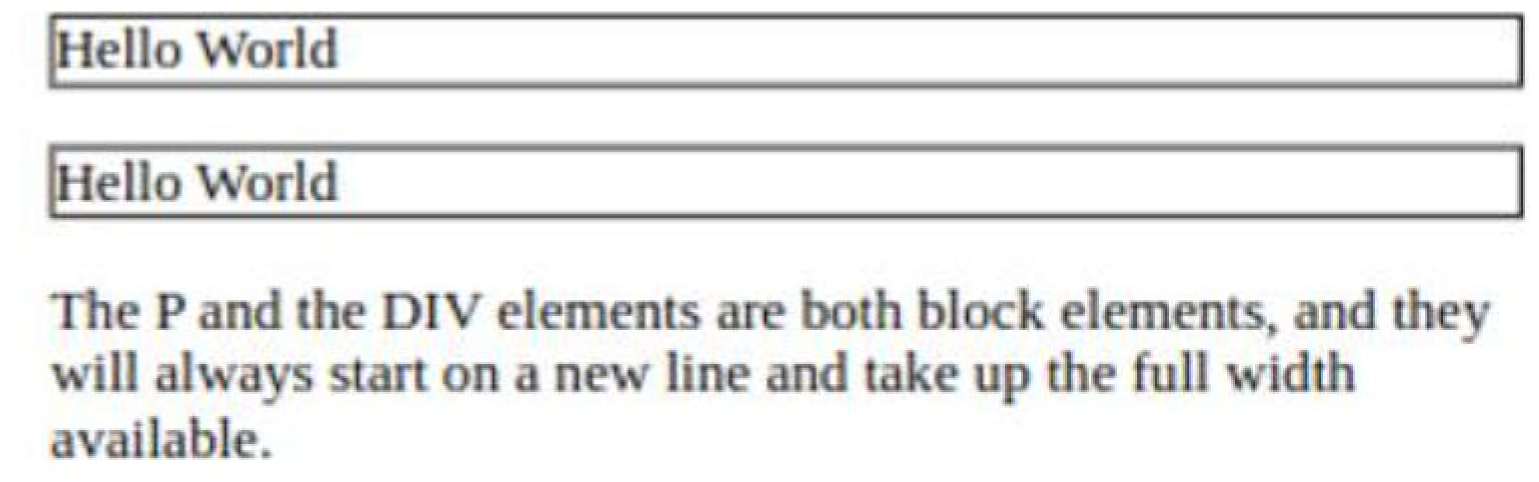
Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

CSS Display Property

The `display` CSS property sets whether an element is treated as a block or an inline element.

Following values are taken by the display property:

- Block
- Inline-block

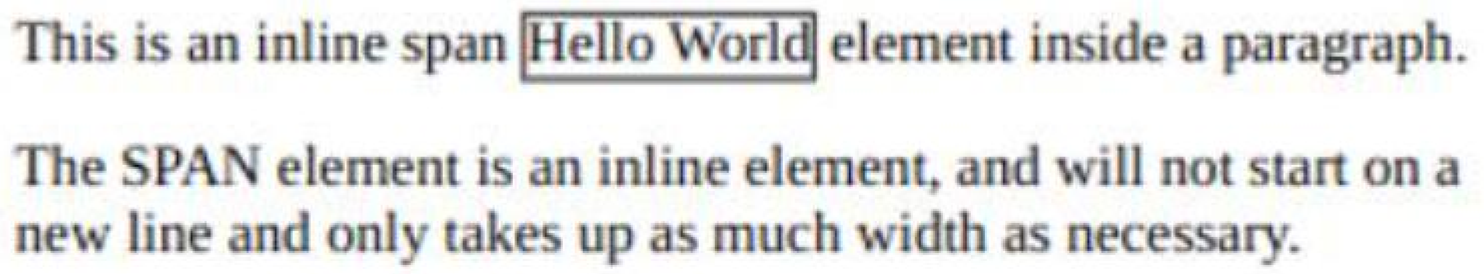


Hello World

Hello World

The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available.

Block Elements



This is an inline span Hello World element inside a paragraph.

The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.

Inline Elements

Slide Note

There is a special value of display, which provides a middle ground between inline and the block. This is useful for situations where you do not want an item to break onto a new line but do want it to respect width and height and avoid overlapping.

An element with display: inline-block does a subset of the block things we already know about:

The width and height properties are respected.

padding, margin, and border will cause other elements to be pushed away from the box.

It does not, however, break onto a new line, and will only become larger than its content if you explicitly add width and height properties.

We have added the display: inline-block to our element.

Display: Inline-Block

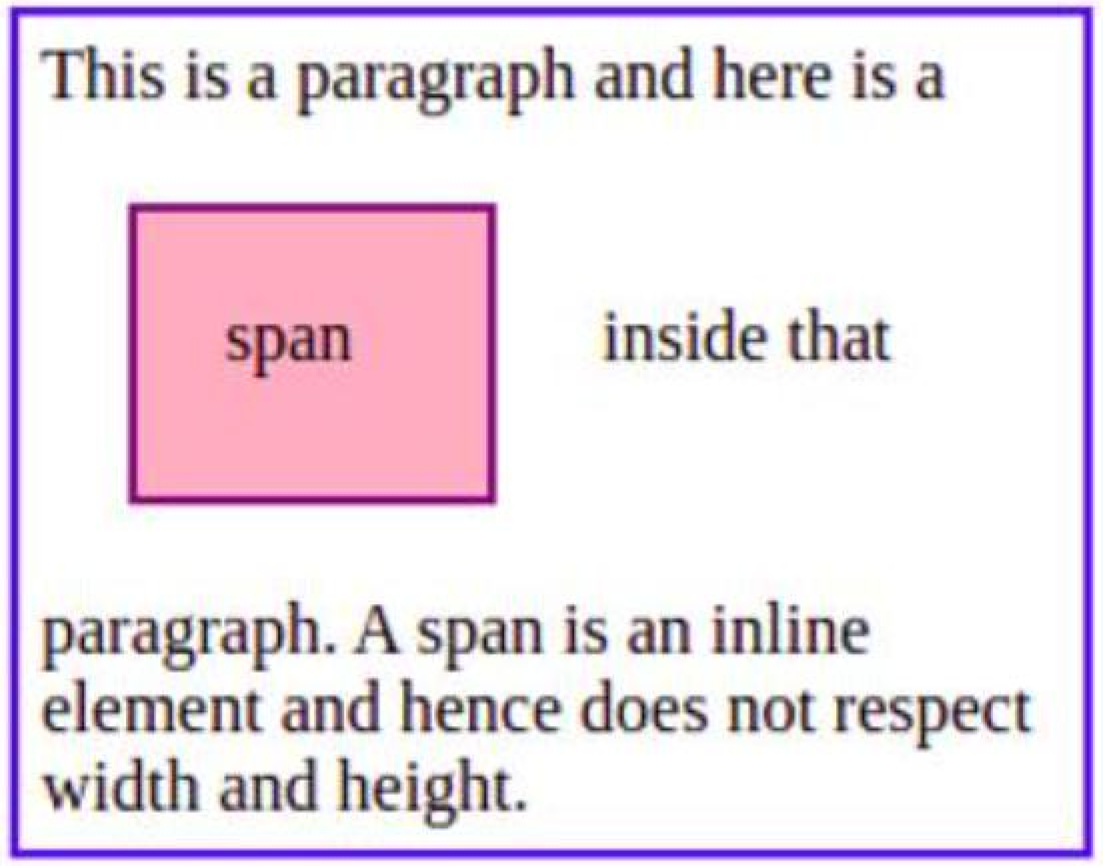
HTML Code:

```
<p>
    This is a paragraph and here
    is a <span>span</span> inside that paragr
    aph. A span is an inline element and
    hence not respect width and height.
</p>
```

CSS Code:

```
span {
    margin: 20px;
    padding: 20px;
    width: 80px;
    height: 50px;
    background-color: lightpink;
    border: 2px solid purple;
    display: inline-block;
}

p {
    width: 18vw;
    border: 1px solid blue;
    padding: 5px;
    font-size: 2em;
}
```



This is a paragraph and here is a span inside that paragraph. A span is an inline element and hence does not respect width and height.

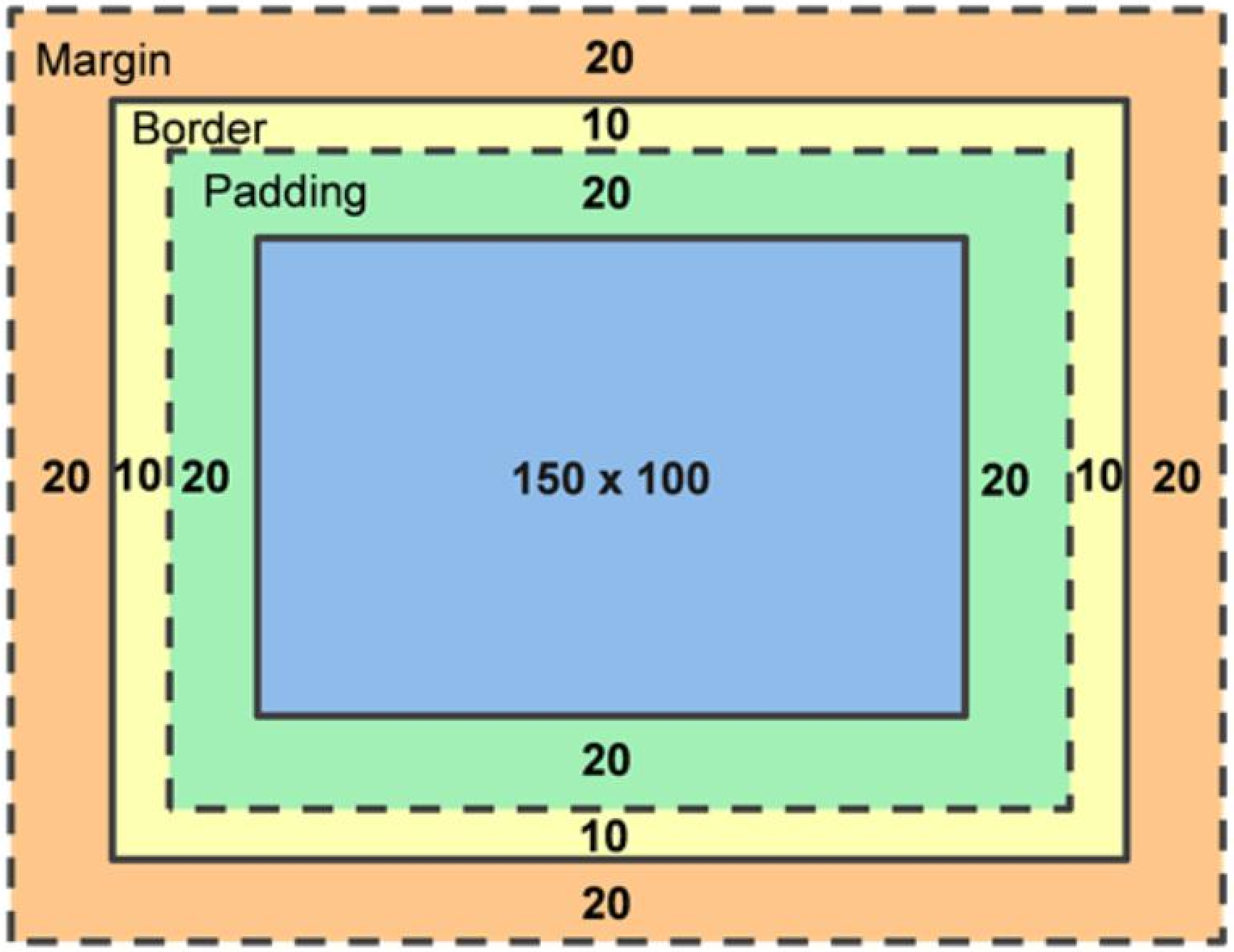
Menu

00:44 00:00:47 44/ 47

Slide Note

Quick Check

Can you notice the width and height of the outermost box?



Menu

00:45 00:00:47

45/ 47

Quick Check: Solution

Width: 250 px
Height:200 px

Explanation:
According to the box model, the total width of an element can be calculated using the following formula:
margin-right + border-right + padding-right + width + padding-left + border-left + margin-left

The total height of an element can be calculated using the following formula:
margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom

Using the formulas, we can find the total height and width of our example code.
Width: 250px = 20px + 10px + 20px + 150px + 20px + 10px + 20px
Height: 200px = 20px + 10px + 20px + 100px + 20px + 10px + 20px