

CST8285

INTRODUCTION TO CSS

Block vs Inline Elements

- ⦿ A block element will take the full width of its parent element, and contains a line break before and after the element
- ⦿ Examples of block elements: `<p>`, `<h1>`, `<h2>`
- ⦿ An inline element takes up only as much space as it needs, and does not contain line breaks before and after.
- ⦿ Examples of inline elements: `<a>`, ``, ``
- ⦿ Inline elements can not have a block element nested inside. For example, you could not put an `<h1>` tag inside an `<a>` tag.

<div> and

- ⦿ A <div> element is a block element
- ⦿ <div> elements are used to wrap sections of a document
- ⦿ A element is an inline element
- ⦿ elements are used to wrap small sections of text, images, etc.

What is CSS?

- CSS is an acronym for Cascading Style Sheets.
- Defines typographical style, colors, and layout of a web page.
- Separates the style from the structure.
- Can apply CSS styles to visible HTML elements.

How to apply CSS

◎ Inline

- Apply styles to an HTML element with the `style` attribute

◎ Embedded

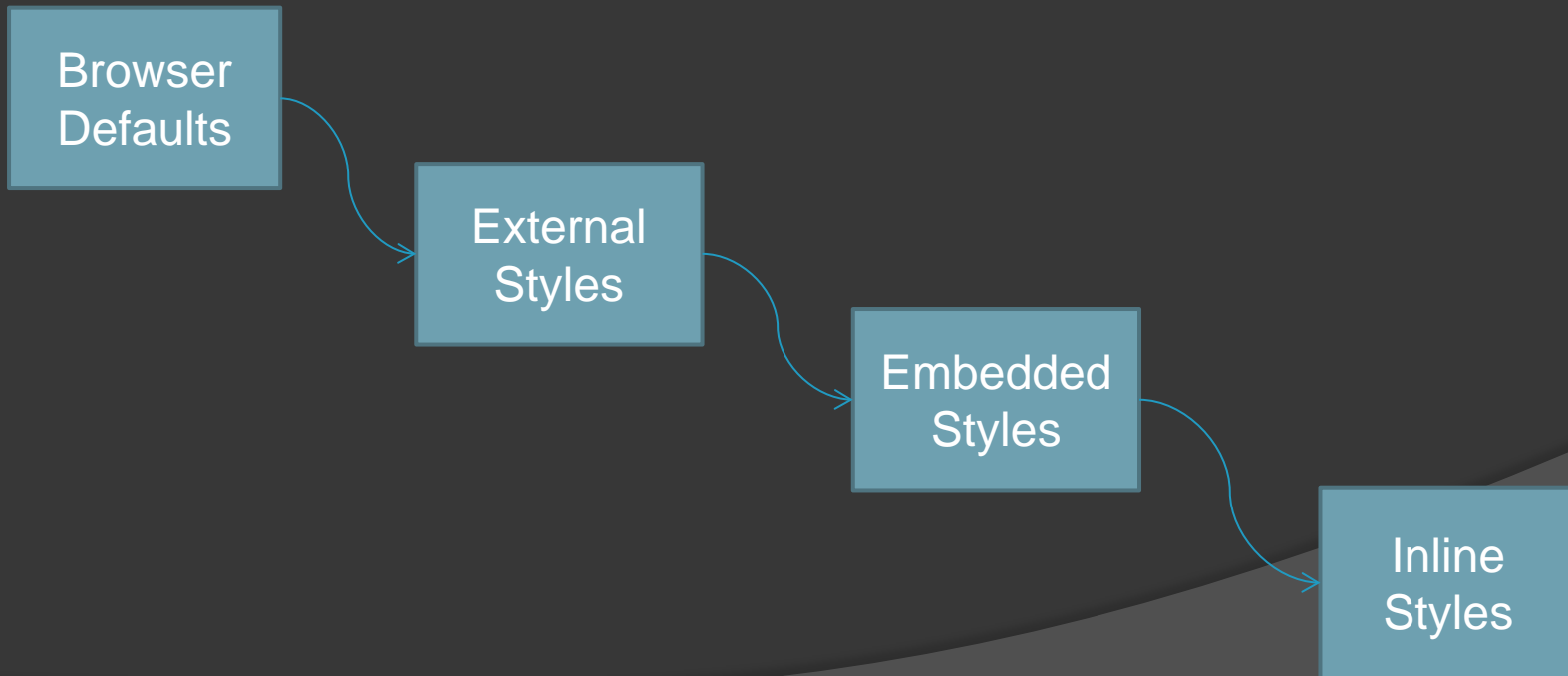
- Define styles in a `<style>` element in the `<head>` section

◎ External

- Coded in an external style sheet

Cascading Styles

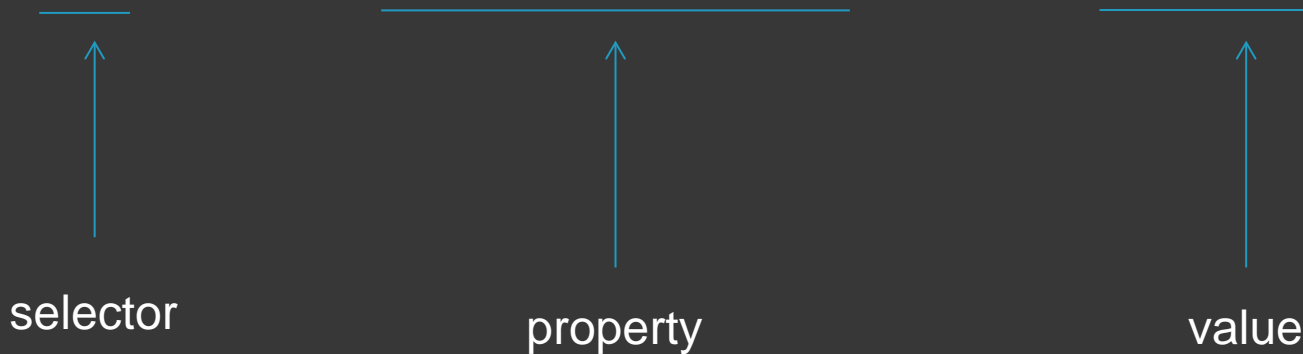
- CSS uses “rules of precedence” to apply styles, with the items on the right overriding items on the left.



Defining a CSS style rule

Here is a basic style rule:

```
td { background-color: #FF0000; }
```



- The curly brackets contain all of the style rules for the chosen element
- The property and value(s) of a rule are separated by a colon (:)
- The property-value pair is called a rule

CSS colors

- ⦿ Two main properties for applying color with CSS
 - `color` and `background-color`
- ⦿ `color` configures the text color of an element
- ⦿ `background-color` configures the background color

CSS colors

- ⦿ The value of `color` and `background-color` can be of the following types:
 - Hexadecimal (ex. `#FF0000`)
 - Decimal (ex. `rgb(255, 0, 0)`)
 - Hue, Saturation and Lightness (ex. `hsl(0, 100% 50%)`)
 - Color words (ex. `red`)
- ⦿ In this course, we will focus on the hex color codes.

Inline CSS styles

- Unlike embedded and external style rules, inline styles do not use a selector
- Styles are passed in the `property: value;` format as the value of the `style` attribute
 - Style attribute can be applied to any visible HTML element
- Ex: `<p style="text-align:center;">`

Inline CSS styles

- Element level inline styles will override all other styles.

- Ex:

```
<body style="color:#FF0000;">  
  <p>This text will appear red</p>  
  <p style="color: #0000FF;">This  
  text will appear in blue</p>  
</body>
```

Embedded CSS

- Applies to the entire document in which it is defined.
- Placed in a `<style>` element in the `<head>` section.
- Uses the same syntax as an external stylesheet.

● Ex:

```
<style>
  p { color: #FF0000; }
  h1 {color: #0000FF; }
</style>
```

External Stylesheets

- Styles are saved in an outside file with .css extension
- Useful for defining styles across multiple pages
- Added to a page in the <head> section using a <link> element
 - Ex. `<link href="css/style.css" rel="stylesheet">`

CSS Selectors

- ⦿ Selectors are used to identify which element a CSS rule applies to
- ⦿ Selectors include:
 - HTML element
 - class
 - id

CSS selector – HTML Element

- Uses the element itself as the selector
- Ex: `p { text-align: center; }`
- Will apply to every occurrence of that element within the scope of the rule

CSS selector – class

- In embedded and external stylesheets, you can define a style class
- Class is identified by a dot, then the class name
 - Ex: `.red { color: #FF0000; }`
- To apply the class to an HTML element, use the `class` attribute
 - Ex. `<p class="red">This is red text.</p>`
- Can apply a CSS class to multiple elements.

CSS selector – id

- In embedded and external stylesheets, you can define a style for an id
- Id is identified by a # then the id name
 - Ex. `#header { text-align: center; }`
- To apply the id style, give the HTML element the corresponding id
 - Ex: `<div id="header">`

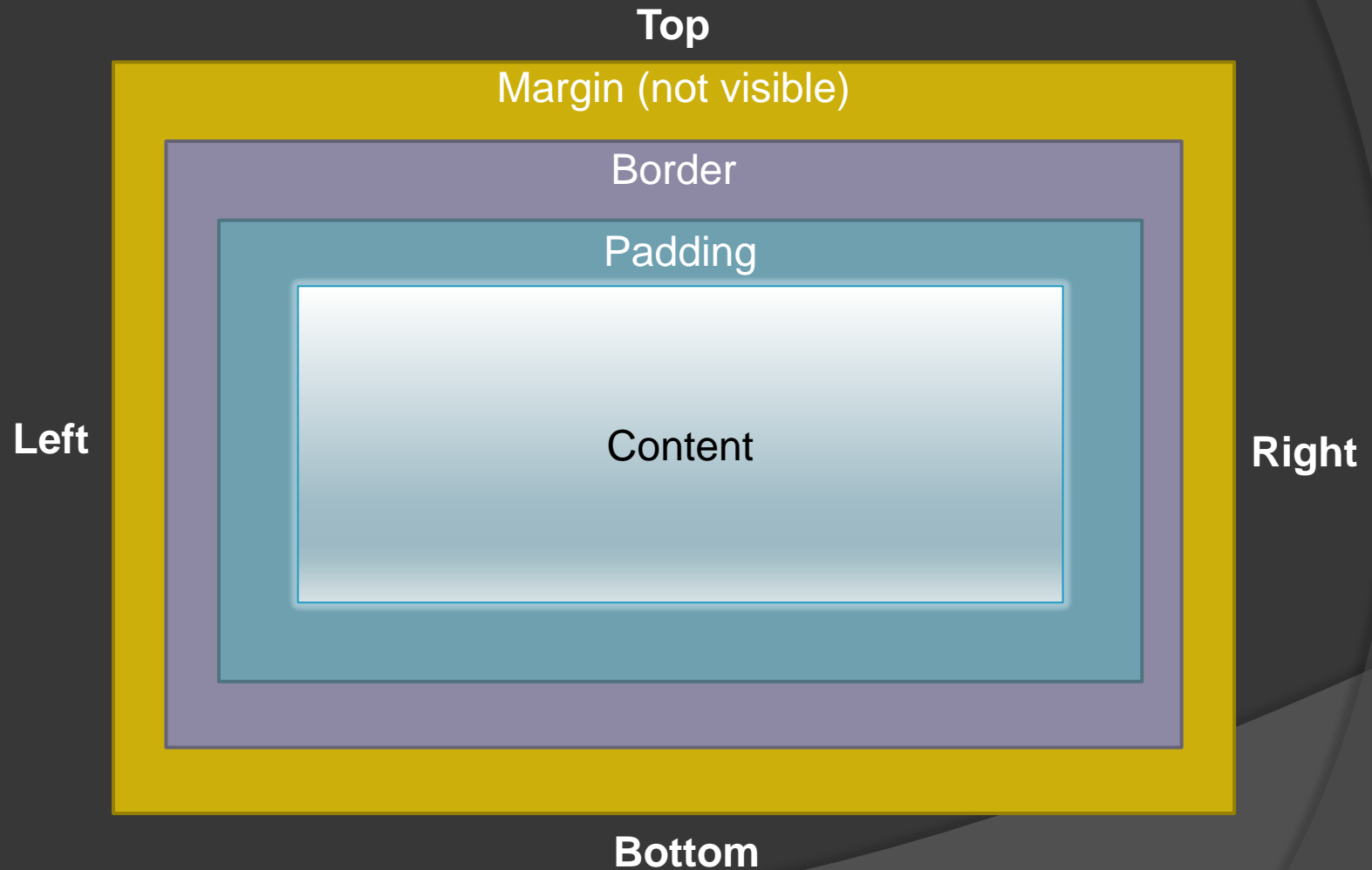
CSS selectors – id vs class

- An id applies to a single element, where a class applies to all elements of that class.

Descendent selector

- You can define a style for an element in the context of a container element
 - Ex: `#header h1 { color: #00FF00; }`
 - The above style would apply to all h1 elements in the header context.
- A selector can have multiple descendents
 - Ex: `#header .nav li { color: #0000FF; }`

The CSS box model



Margins and Padding

- ⦿ Margins are the non-visible space outside of the element
 - Ex. `margin: 5px` would place a non-visible space of 5px around the element.
- ⦿ Padding is the space between the border and the content of an element
 - Ex. `padding: 5px;` would place a 5px space between the border and the element's content.
- ⦿ Can specify margins and padding at the top, left, right, or bottom of an element
 - Ex. `padding-top: 5px;` would put 5px of padding only on the top of the element.

Positioning with CSS

- ⦿ Positioning can be static, fixed, relative, or absolute
 - static – default positioning.
 - fixed – relative to browser window
 - relative – relative to its static position
 - absolute – relative to first non-static parent element
- ⦿ Position with top, bottom, left and right
 - Value is a pixel value (px) of how many pixels between the property and the element
 - Ex. `top: 50px` would be 50px from the top of the element

Modifying Text with CSS

- Common CSS properties for modifying text:
 - `text-align`: positions the text
 - Values include left, center, right, or justified.
 - `font-weight`: changes thickness of letters:
 - Values include lighter, normal, bold, bolder
 - `color`: sets the color of the text
 - `text-decoration`: sets decoration to the text
 - Values include underline, overline, line-through

Modifying Tables with CSS

- ⦿ Common CSS properties for modifying `<table>` elements:
 - `width` – specifies width in pixels or percent of screen
 - `border-spacing` – configures space between cell borders
 - `border` – configures the border of the table (not the cells)
 - `border-collapse`: `collapse` (no space between borders), `separate` (space between borders, default value)
- ⦿ Styles that modify text and color applied to the table will apply to all cells in that table, unless they are overridden.

Modifying table cells with CSS

- Common CSS properties for modifying `<td>` elements:
 - `color`: defines the text color
 - `background-color` – defines the background color
 - `padding` – defines the amount of space between the cell contents and the border

Float property

- ⦿ Elements floated horizontally
- ⦿ Can float an image to the right or left, not up or down
- ⦿ Text elements found after a floated element will flow around it.
- ⦿ Text elements before a floated object will not be affected.
- ⦿ Elements that are absolutely positioned ignore the float property.