

Web Design

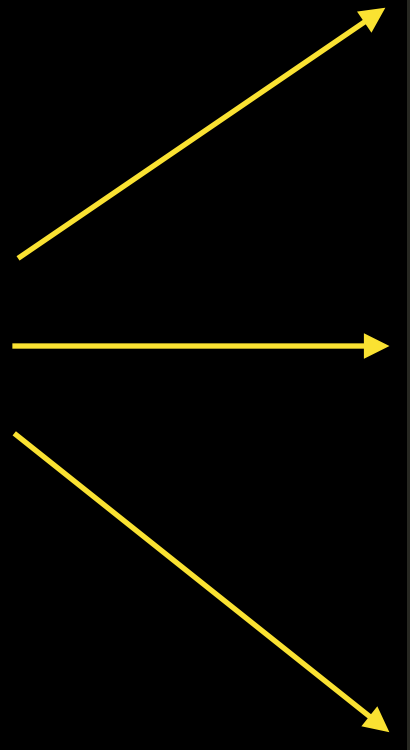
CSS Selectors & Positioning

CSS Selectors

CSS Selectors

- A rule is applied to **all** elements that meet the selector's criteria

```
p {  
  font-size: 1.1rem;  
}
```



```
<p>  
  Lorem ipsum dolor sit amet consectetur adipisicing elit.  
</p>  
  
<p>  
  Lorem, ipsum dolor sit amet consectetur adipisicing elit.  
</p>  
  
<p>  
  Lorem ipsum dolor sit, amet consectetur adipisicing elit.  
</p>
```

The diagram illustrates the application of a CSS rule. On the left, a CSS selector `p` is shown within a rule block, with a yellow box highlighting the selector. Three yellow arrows originate from the right side of this rule block and point to three separate paragraph elements on the right. Each paragraph is enclosed in `<p>` and `</p>` tags and contains a line of Lorem Ipsum text. This visualizes that the CSS rule is applied to every element that matches the `p` selector.

CSS Selectors

Combinators

- Combinators can be used in selectors to add more specificity
- Examples:
 - “ ” (space): descendant
 - “>”: child
 - “~”: sibling
 - “+”: adjacent sibling

CSS Selectors

Universal Selector

- The universal selector “*” (asterisk) can be used to style all elements at once

CSS Classes

- CSS classes allow us to create specific style cases that can be applied to specific elements

```
<p class="example">
  Lorem ipsum dolor sit amet consectetur adipisicing elit.
</p>
```

```
.example {
  font-size: 1.5rem;
}
```

or

```
p.example {
  font-size: 1.5rem;
}
```


Cascading

Cascading

- CSS rules are overwritten by rules with more specific selectors
- Selector specificity (decreasing order):
 - 1: inline (`style="property: value;"`)
 - 2: id (`#example`)
 - 3: class (`.example`)
 - 4: type (`h1`)
 - 5: definition order (later is stronger)

Cascading

- What color is the following paragraph?

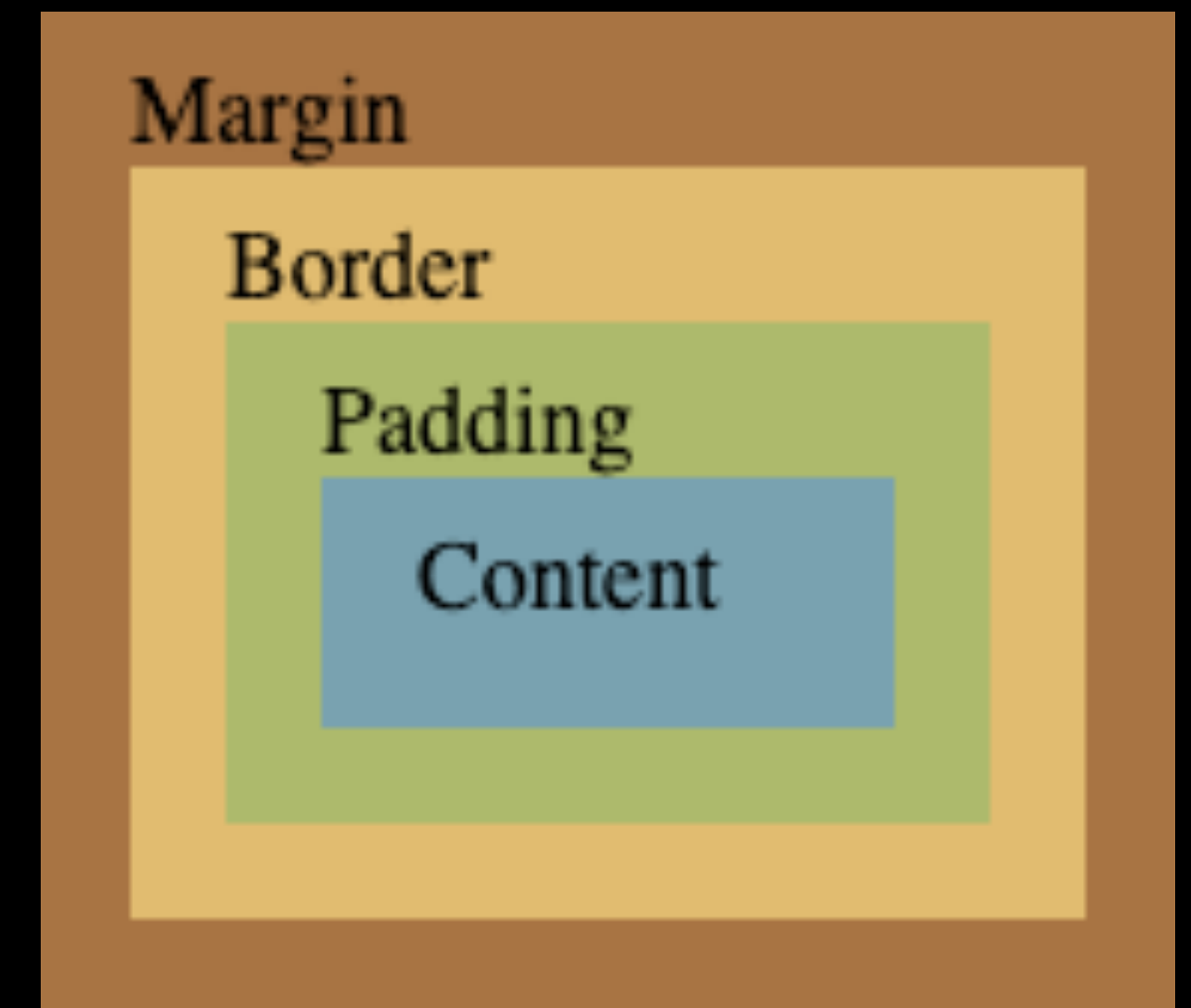
```
#element-id {  
  color:  red;  
}  
  
.element-class {  
  color:  blue;  
}  
  
p {  
  color:  yellow;  
}  
  
p {  
  color:  green;  
}
```

```
<p id="element-id" class="element-class" style="color:  turquoise;">  
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo accusamus  
</p>
```

CSS Box Model

CSS Box Model

- HTML elements can be thought of as boxes. In CSS the box model refers to design and layout of HTML elements
- The amount of space an HTML element takes up is composed of four pieces:
 - **Content:** the text and sub-elements of the element
 - **Padding:** spacing between content and the border/edge (set with the padding attribute)
 - **Border:** line around the element (set with the border property)
 - **Margin:** spacing around the outside of the element (set with the margin property)



CSS Box Model (cont.)

- Check your understanding:
 - How much space horizontally does an element take up with the following CSS properties:
 - content width of 50px
 - padding of 5px
 - border of 1px
 - margin of 10px

Simplifying Box Sizing

- By default, HTML elements have the CSS box-sizing property set to content-box, meaning that the width and height properties affect only the content box and any margin and borders are added on afterwards
- To make it easier to reason about the size of HTML elements we can set the box-sizing property to the border-box
 - This means that the width and height properties account for padding and border widths as well

```
* {  
  box-sizing: border-box;  
}
```

CSS Position

Position

- CSS uses position to decide where to place an element
- Default position is static, meaning after the element before and before the element after (in the HTML code)
- Other values:
 - absolute (stuck to a specified place in the document)
 - fixed (stuck to a specified place in the window)
 - sticky (static then fixed when scrolled to)
 - relative (descendants will use this element when deciding where to go with absolute/fixed positioning)