



2024-02-06 09:58

# CSCI 165

## Introduction to the Internet and the World Wide Web

### Lecture 3: CSS 2



Jetic GU  
2024 Spring Semester (S1)

# Overview

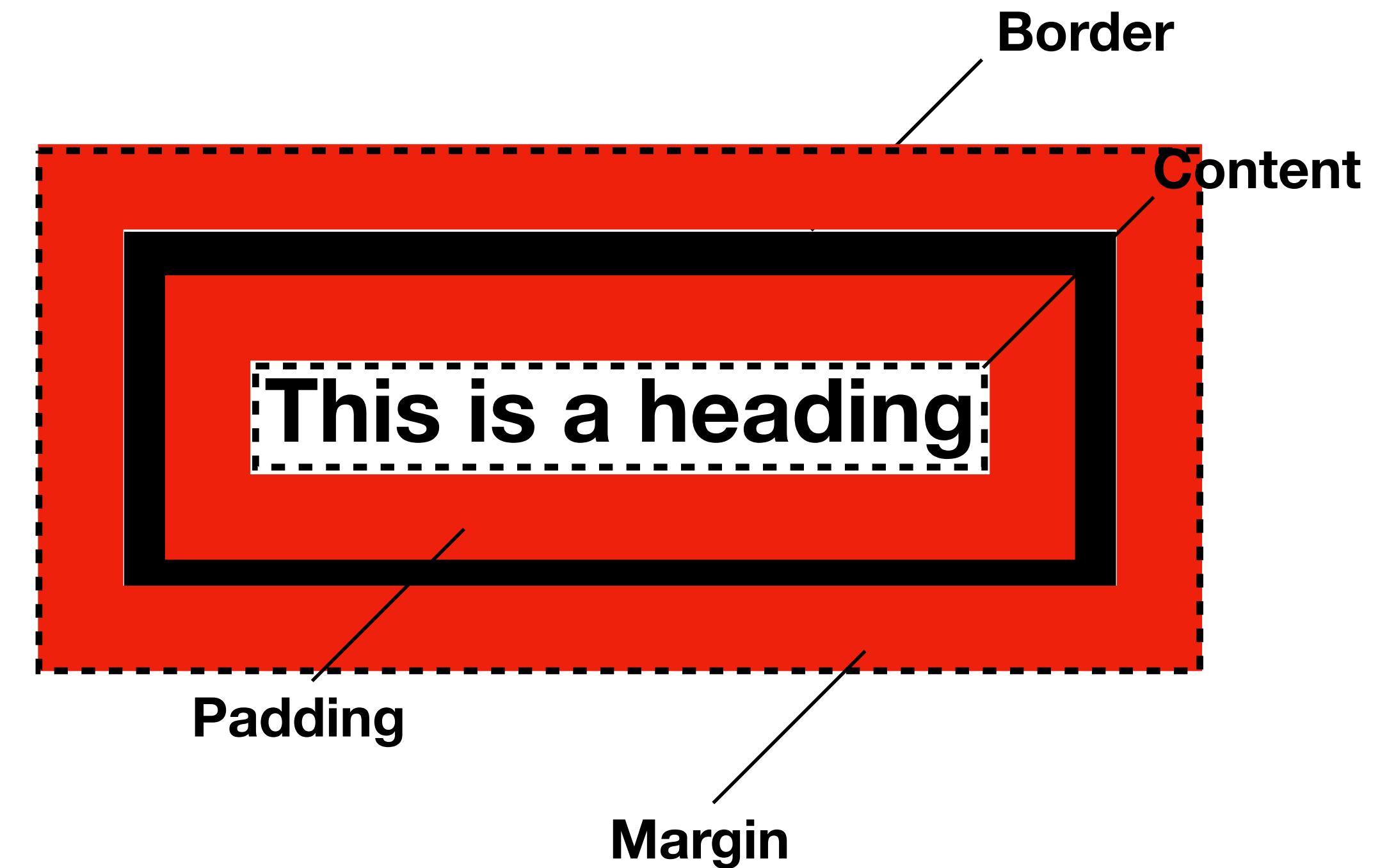
- Focus: Course Introduction
- Architecture: WWW
- Core Ideas:
  1. CSS Boxes
  2. More Selectors
  3. Positioning

# HTML Validator

- In quiz1, I asked you to find a usable HTML validator
  - Use it please, for your submissions
  - Some warnings are less important, especially those associated with metadata
  - Others must be fixed from Lab 4 onwards

# CSS Boxes

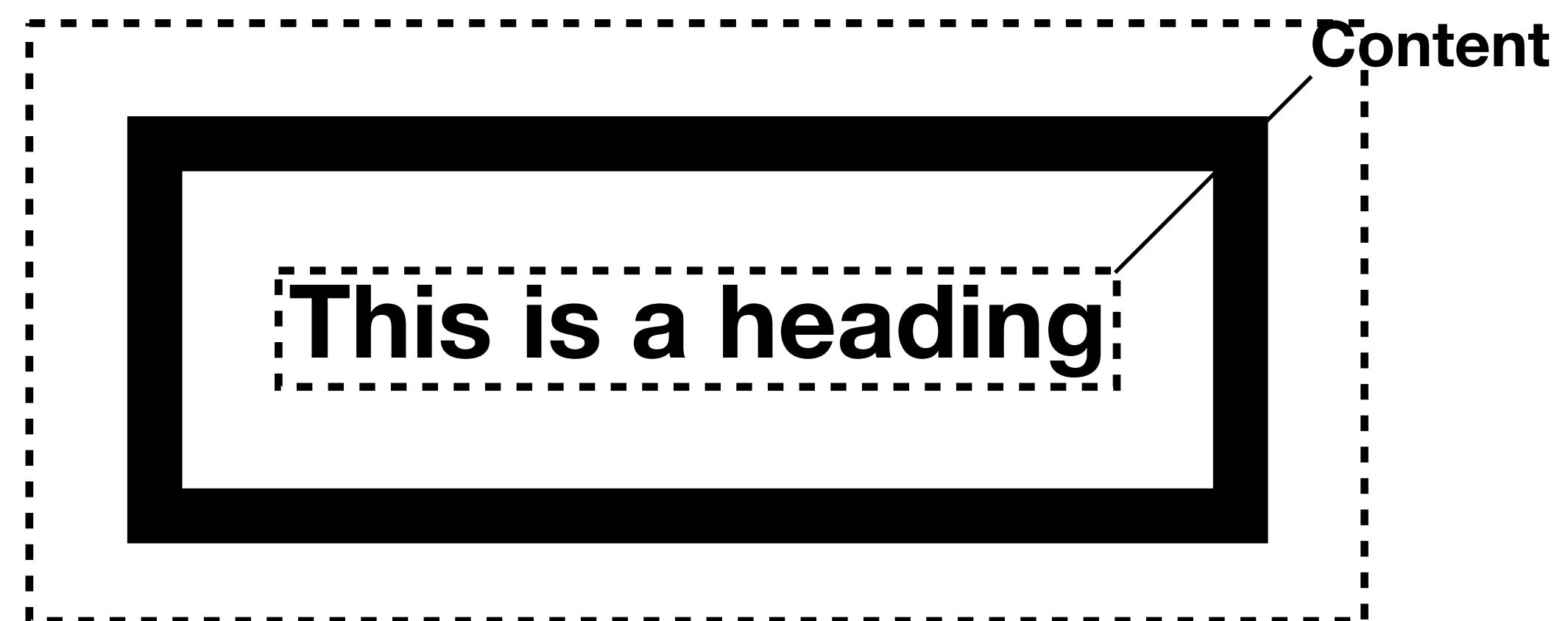
- Everything is boxed
- For example, an `<h1>` tag
  - **Content**  
Where text and images are
  - **Padding**  
Gap between content and border, transparent
  - **Border**
  - **Margin**  
Gap between boxes, transparent



# CSS Boxes

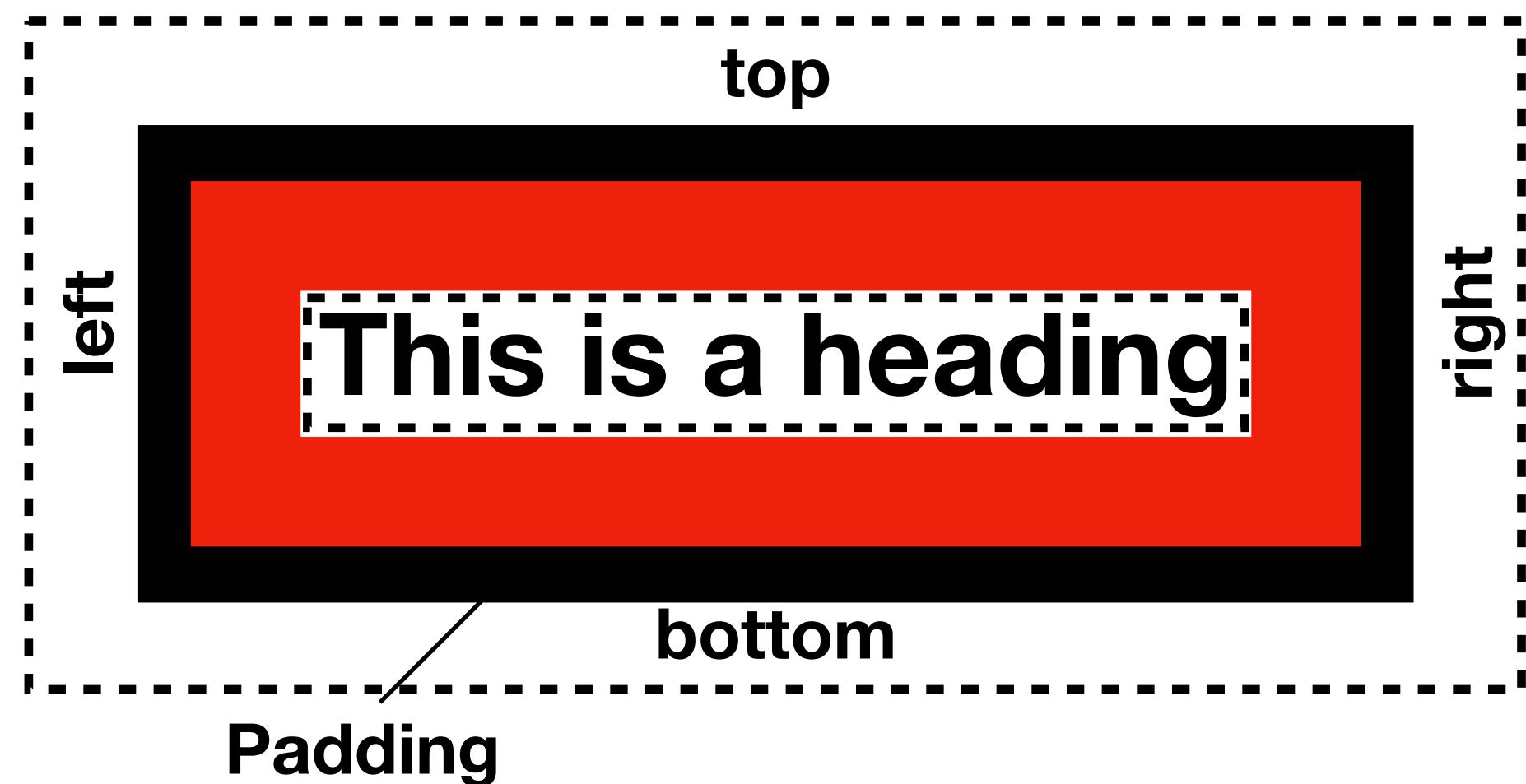
- Content has
  - height and width
  - color and background-color
  - e.g.

```
div {  
    height: 200px;  
    width: 50%;  
}
```
- The 50% here means 50% of its containing box<sup>1</sup>'s total width



# CSS Boxes

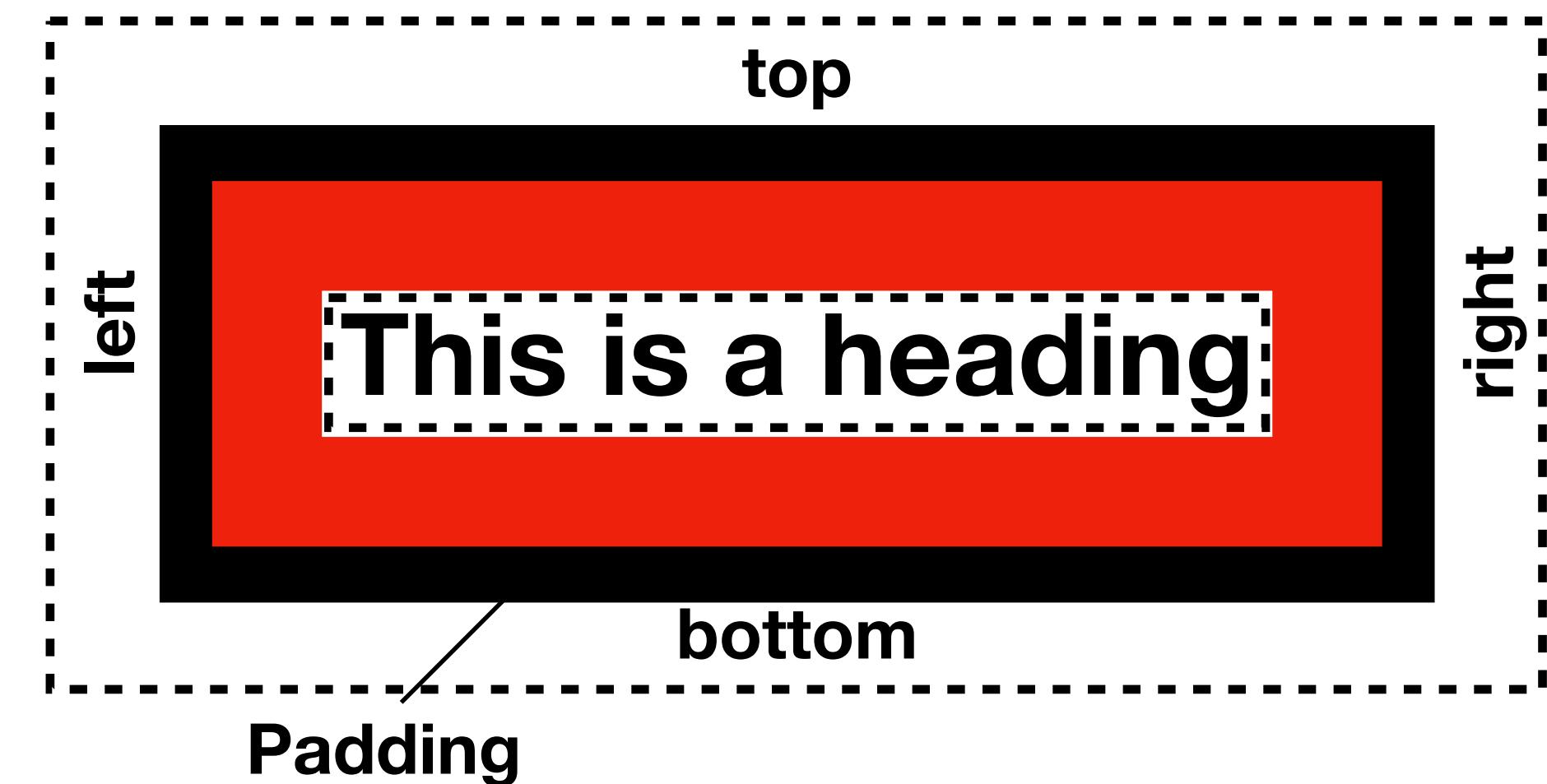
- Padding has
  - padding-top
  - padding-left
  - padding-right
  - padding-bottom
  - values to use: length, or inherit (default)



# CSS Boxes

- Example

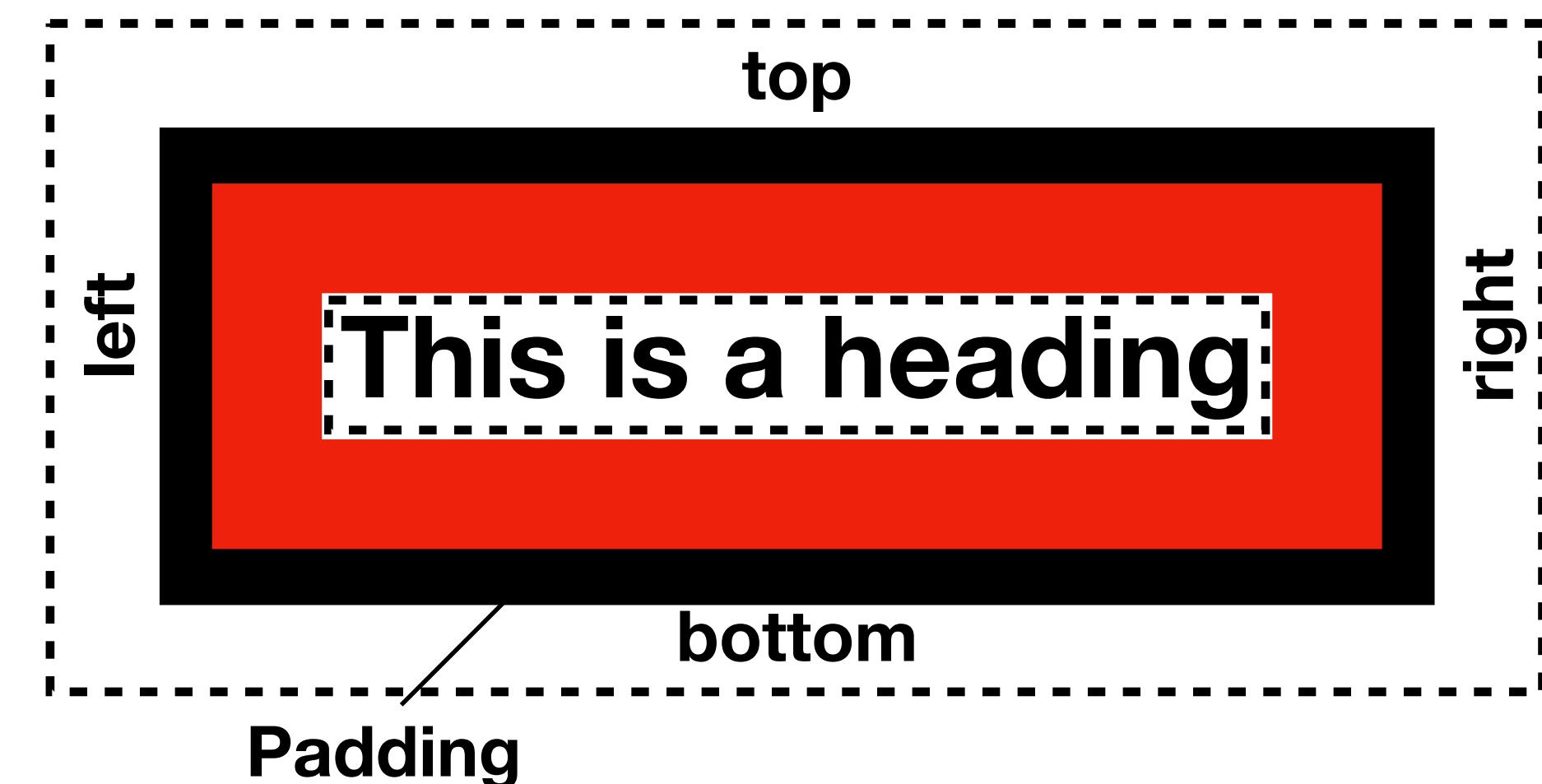
```
h1 {  
    padding-top: 10px;  
    padding-bottom:  
    10px;  
    padding-left: 15px;  
    padding-right: 15px;  
}
```



P1  
Box

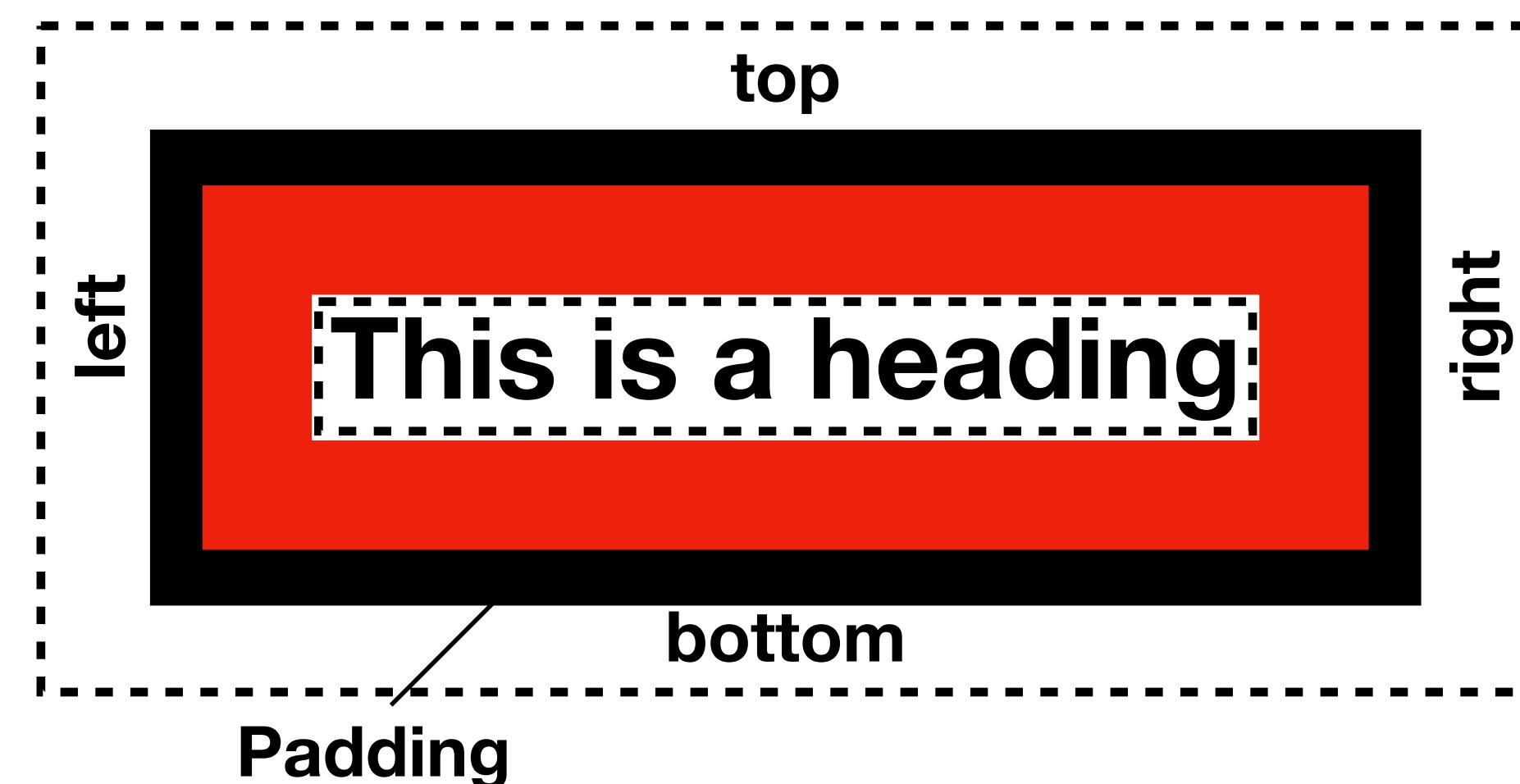
# CSS Boxes

- Padding also has
  - padding: 25px;
  - All four paddings are 25px
  - This is OK



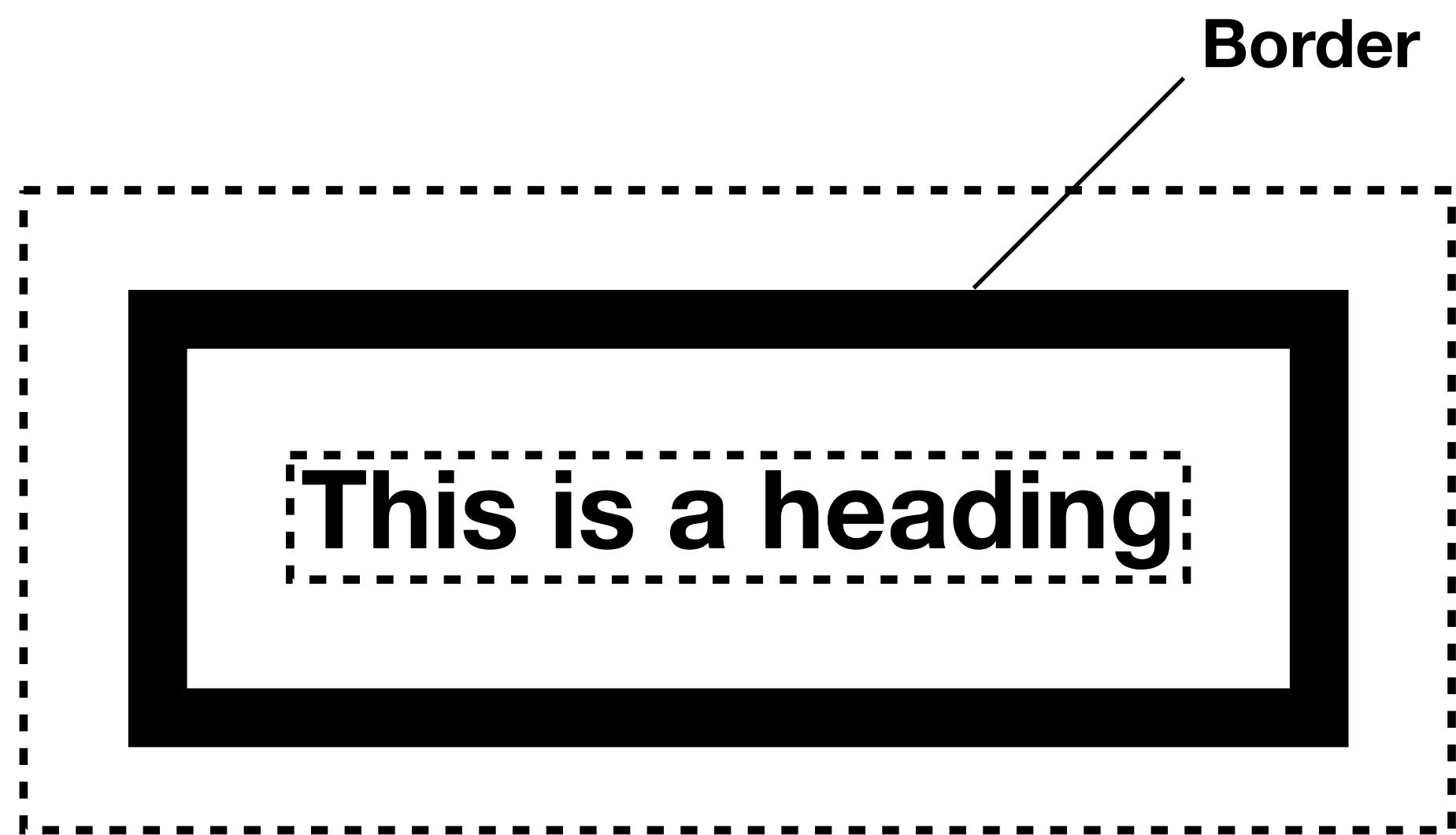
# CSS Boxes

- Padding also has
  - padding: 25px 50px 75px 100px;
  - This goes: top, right, bottom, left
  - This is not a good approach
  - Defining paddings like this can be visually difficult to debug and change
  - sometimes you just want to do Ctrl-F (or Command-F)



# CSS Boxes

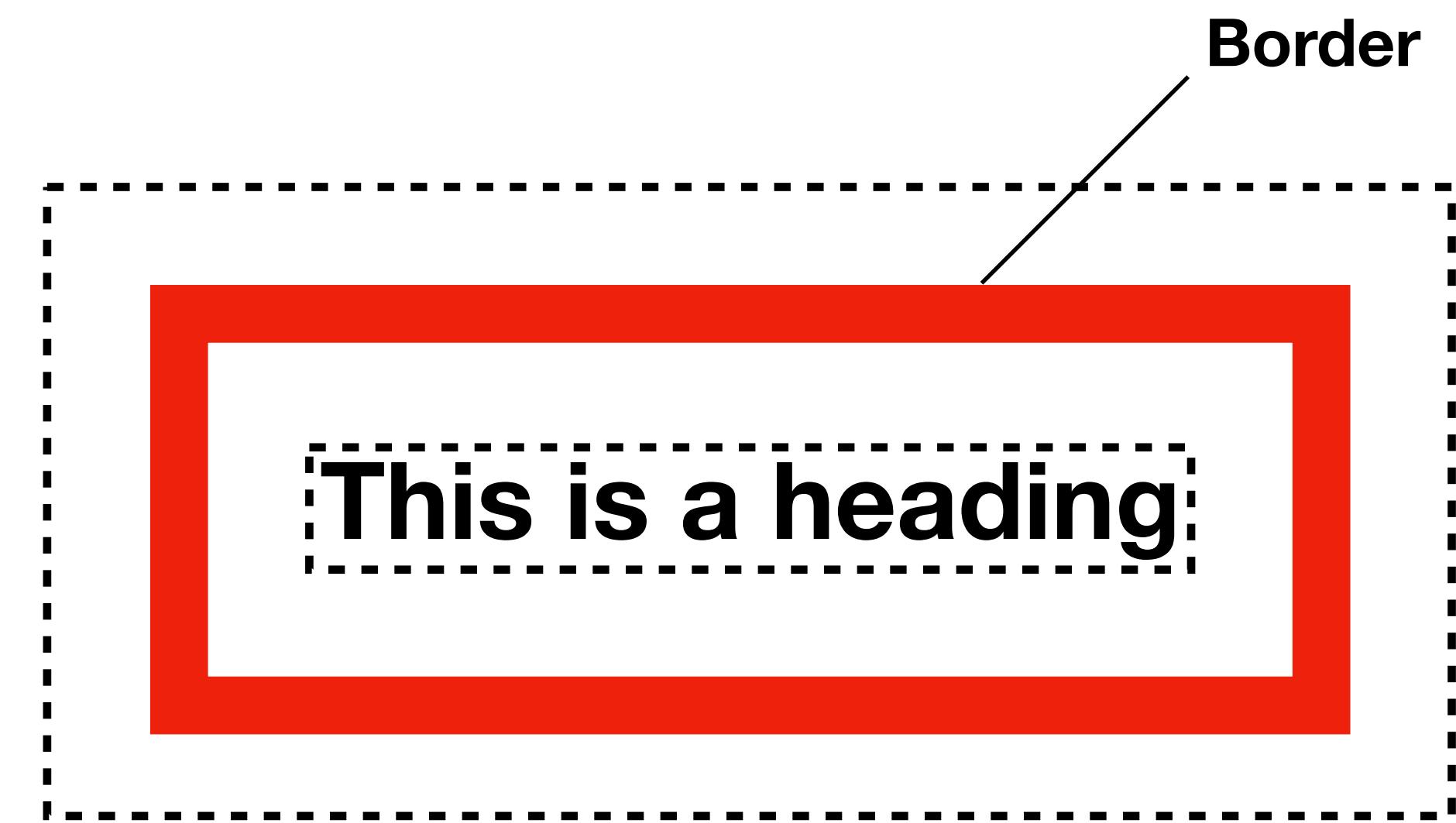
- Border has
  - border-width  
Thickness, use unit of measurements here
  - Can be defined like padding (top, right, bottom, left)
- border-style
  - dotted, dashed, solid, double, hidden, none, etc.
- border-color



# CSS Boxes

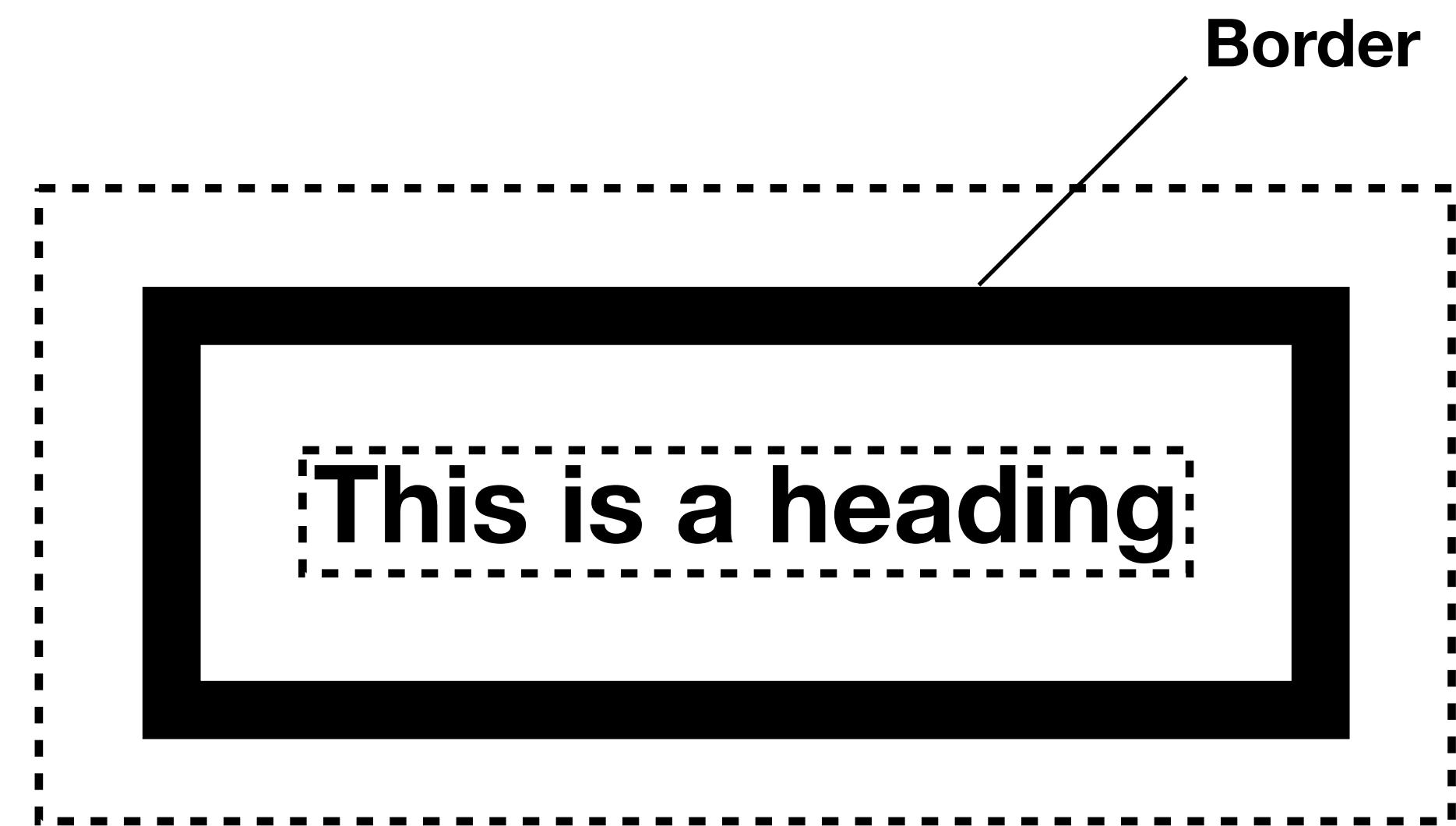
- Example:

```
h1 {  
    border-style: solid;  
    border-width: 20px;  
    border-color: red;  
}
```



# CSS Boxes

- Each side can be changed differently. e.g.
  - border-top-style
  - border-left-style
  - ...



# CSS Boxes

- Margin has
  - margin-top
  - margin-left
  - margin-right
  - margin-bottom
  - values to use: length, or inherit (default)



Margin

# CSS Boxes

- margin: auto;
- The element (border included) will take up its specified width, the rest of the space is split evenly between left and right



# CSS Selectors

- Previously, we discussed **element selector**, where you used tag names to do styling
- e.g.

```
h1, h2 {  
  ...  
}
```

# CSS ID selectors

- Sometimes you have a lot of elements with the same tag, but you want to adjust them individually
- This is especially the case with `<div>`, which is a common element used to create content blocks in HTML
- `<div>` itself doesn't offer anything except for being a block, no special abilities

# CSS ID selectors

- index.html

```
<div id="nav">  
  <h1><a ...>link1</a></h1>  
  <h1><a ...>link2</a></h1>  
</div>
```

```
<div>  
  <h1>Q1 Answer</h1>  
  <p>...</p>  
</div>
```

```
<div>  
  <h1>Q2 Answer</h1>  
  <p>...</p>  
</div>
```

- style.css

```
#nav {  
  text-align: center;  
  color: red;  
}
```

- Here we have a navigational bar, similar to a menu bar

- done using `<div>`, with attribute `id`

- This needs to be styled differently from the rest of the `<div>`

# CSS ID selectors

- id names
  - any combination of letters, underscore, numbers, but cannot start with numbers
- in HTML
  - id attribute can be used on any element tags
- in CSS
  - start with #, then followed by id name (no space between)

# CSS ID selectors

- Here are some examples in CSS
  - `#nav { ... }`
  - `h1#main { ... }`  
**h1 elements with id=main attribute**
  - `p#main { ... }`  
**p elements with id=main attribute**
- Tip: id names should be more unique in a page

# CSS Class Selectors

- **index.html**

```
<div id="nav">  
  <h1><a ...>link1</a></h1>  
  <h1><a ...>link2</a></h1>  
</div>
```

```
<div class="answer">  
  <h1>Q1 Answer</h1>  
  <p>...</p>  
</div>
```

```
<div class="answer">  
  <h1>Q2 Answer</h1>  
  <p>...</p>  
</div>
```

- Here you have multiple `<div>`s that should be styled similarly
- Use the `class` attribute for multiple elements of a single class

# CSS Class Selectors

- **index.html**

```
<div id="nav">
  <h1><a ...>link1</a></h1>
  <h1><a ...>link2</a></h1>
</div>
```

```
<div class="answer">
  <h1>Q1 Answer</h1>
  <p>...</p>
</div>
```

```
<div class="answer">
  <h1>Q2 Answer</h1>
  <p>...</p>
</div>
```

- **style.css**

```
.answer {
  text-align: center;
  color: red;
}
```

- start with period dot, then followed by class name

- all instances of the same class will be affected

# CSS Special Subs

- Some elements can have special subs

```
a:link { color: red; }
```

```
a:visited { color: purple; }
```

```
a:active, a:hover { text-decoration: underline; }
```

- This changes the behaviour of a hyperlink, from when you just seen it, hover your mouse over it, or when you have clicked on it
- `hover` exists for a lot of elements, including `<div>`

# CSS Positioning

- Previously, all of the elements are rendered from top to bottom
- You didn't really have elements next to one another on the same horizontal level
- By browser default, each element's box will take up all of the available width of the browsing area
- What if you want something different?

# CSS Positioning

- Partitioning?
  - The whole page can be partitioned
  - Top menu: grey
  - Left menu: green
  - Main content: red

The screenshot shows the W3Schools CSS Tutorial page. The top navigation bar is grey, containing links for Tutorials, Exercises, Certificates, Services, a search bar, and user account options (Sign Up, Log in). Below the navigation is a horizontal menu bar with links for HTML, CSS (which is highlighted in green), JAVASCRIPT, SQL, PYTHON, JAVA, PHP, HOW TO, W3.CSS, C, C++, C#, BOOTSTRAP, REACT, and MYSQL. To the left of the main content area is a green sidebar titled "CSS Tutorial" with a "CSS HOME" section and a list of CSS topics from "CSS Introduction" to "CSS Combinators". The main content area has a pink background and is divided into two columns. The left column contains the title "CSS Tutorial", a "Home" link, and a "Next >" button. The right column contains text about CSS being the language for styling HTML documents and describes how it styles elements. It also mentions that the tutorial covers basic to advanced topics and includes a "Start learning CSS now »" button. Below this is a section titled "Examples in Each Chapter" with text about containing hundreds of examples and an online editor. A "CSS Example" box shows the following CSS code:

```
body {  
    background-color: lightblue;  
}
```

A blue diagonal banner at the bottom right reads "Technical". In the bottom right corner, there is a small brown square with the text "Create" and three colored squares (blue, orange, red) above it.

# CSS Positioning

- Tools
  - position **property**
    - static: default value, not affected by other positioning properties, will just follow the rest of the elements
    - relative: setting the top, right, bottom, and left properties will cause it to be adjusted away from its normal position.
    - fixed: positioned relative to the viewport, will not be affected by scrolling
    - absolute: positioned relative to the nearest positioned parent, with fixed position in it

# CSS Positioning

- position: relative/fixed/absolute options
  - top, left, bottom, right: length measurements
  - e.g.

```
#nav {  
    top: 0;  
    left: 0;  
}
```

# CSS Positioning

- Tools
  - overflow property
    - defines a scrollable box, used to create sub-content boxes
    - visible: default, when there's more stuff than the content box allows, it renders outside (overflows)
    - hidden: when there's more stuff than the content box allows, it doesn't get rendered (hidden)
    - scroll: adds a scroll bar
    - auto: adds scroll bar only when needed

# CSS Positioning

- For each three parts
- Find out:
  - id/class of containing <div>
  - position?
  - overflow?
  - height and width?

The screenshot shows the W3Schools website with the URL <https://www.w3schools.com/css/>. The page title is "CSS Tutorial". The left sidebar has a green header "CSS Tutorial" and a "CSS HOME" section, followed by a list of CSS topics. The main content area has a pink background and contains text about CSS being the language for styling HTML documents, its purpose, and a summary of the tutorial's content. It includes a "Start learning CSS now" button and an "Examples in Each Chapter" section. A code snippet for a CSS rule is shown in a red box. A blue diagonal banner at the bottom right reads "Technical".

W3Schools

Tutorials ▾ Exercises ▾ Certificates ▾ Services ▾ Search... Get Certified Sign Up Log in

HTML CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C C++ C# BOOTSTRAP REACT MYSQL >

**CSS Tutorial**

**CSS HOME**

- CSS Introduction
- CSS Syntax
- CSS Selectors
- CSS How To
- CSS Comments
- CSS Colors
- CSS Backgrounds
- CSS Borders
- CSS Margins
- CSS Padding
- CSS Height/Width
- CSS Box Model
- CSS Outline
- CSS Text
- CSS Fonts
- CSS Icons
- CSS Links
- CSS Lists
- CSS Tables
- CSS Display
- CSS Max-width
- CSS Position
- CSS Z-index
- CSS Overflow
- CSS Float
- CSS Inline-block
- CSS Align
- CSS Combinations

**CSS Tutorial**

Home Next >

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

This tutorial will teach you CSS from basic to advanced.

Start learning CSS now »

**Examples in Each Chapter**

This CSS tutorial contains hundreds of CSS examples.

With our online editor, you can edit the CSS, and click on a button to view the result.

**CSS Example**

```
body {  
background-color: lightblue;  
}
```

Create