



NEXT  
ACADEMY

# Introduction to Web Development

## Week 2: How to Make Websites Look Good with CSS





# Agenda: Week 2

- Introduction to CSS
  - colors, text, spacing and sizing, etc
  - selector overrides
- Chrome developer tools
  - or how to steal code





NEXT  
ACADEMY

# CSS

Time to make things fancy!



# HTML only

## Sher Minn

- [About](#)
- [Gallery](#)
- [Contact](#)

## Hello there.

Welcome to my universe. Here you will find strange things.

### About

Hi, my name is **Sher Minn**. I'm a frontend web engineer at Viki where I build features with HTML/CSS and JavaScript.

In my time off, I like creating screensaver art with p5.js. You can find me lurking on Twitter and taking pictures of buildings on Instagram.

### Gallery

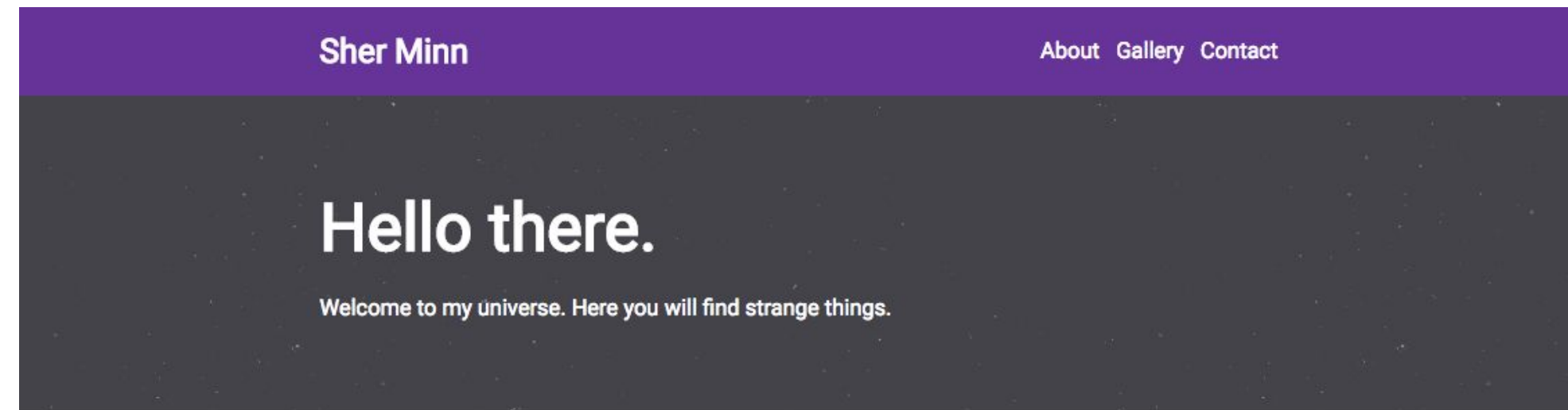


### Get in touch

You can find me on these social media links sometimes. Come say hi!



# HTML + CSS



## About

Hi, my name is Sher Minn. I'm a frontend web engineer at Viki where I build features with HTML/CSS and JavaScript.

In my time off, I like creating screensaver art with p5.js. You can find me lurking on Twitter and taking pictures of buildings on Instagram.

## Gallery





# Where to Put CSS?

- **Option 1 : In separate file and link it to HTML**
  - Save CSS in a new file
    - `style.css`
  - Add link in HTML
    - `<link rel="stylesheet" href="style.css"/>`
- **Option 2 : In <head> of HTML file**
- **Option 3 : Inline CSS**
  - Not recommended and not covering





# HTML IS MADE OUT OF TAGS







# Rules

CSS is made out of rules

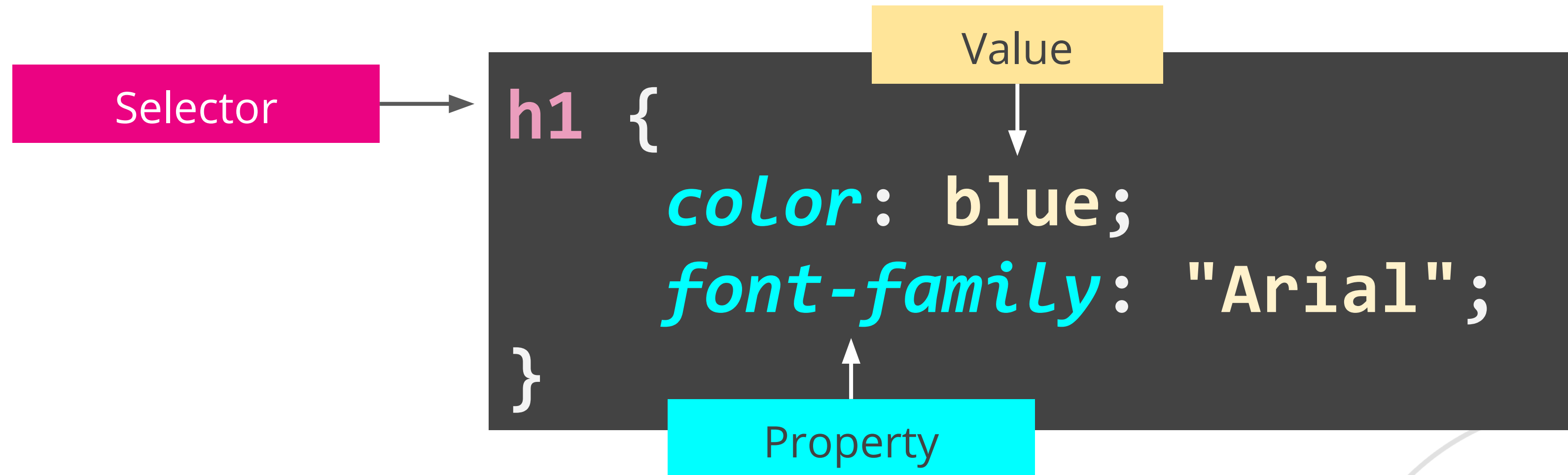
Rules consist of **selectors** that have **properties** and **values**

```
➡ h1 {  
    color: blue;  
    font-family : "Arial";  
}
```





# Anatomy of a Rule



- **Selectors** select which HTML tag to style
- **Property** refers to which style to customize
- **Value** is the value assigned to the given property
- Each rule can have as many property-value pairs



# Remember, Remember

```
h1 {  
    color: blue;  
    font-family: "Arial";  
}
```

- Always open curly brace { after selector, and close } after all the property-value pairs
- Each property has to be followed by a colon :
- Remember to end each property-value pair with semicolon ;





NEXT  
ACADEMY

# Colors

For text and backgrounds



## HTML

## CSS

## OUTPUT



# CODE DEMO





# colors - keywords

## HTML

```
<h1> Hello </h1>

<p>
  Nice to meet you
</p>
```

## CSS

```
h1 {
  color: blue;
}
p {
  color: orangered;
}
```

## OUTPUT

Hello

Nice to meet you



# colors - hex

## HTML

```
<h1> Hello </h1>  
  
<p>  
Nice to meet you  
</p>
```

## CSS

```
h1 {  
  color: #E32609;  
}
```

## OUTPUT

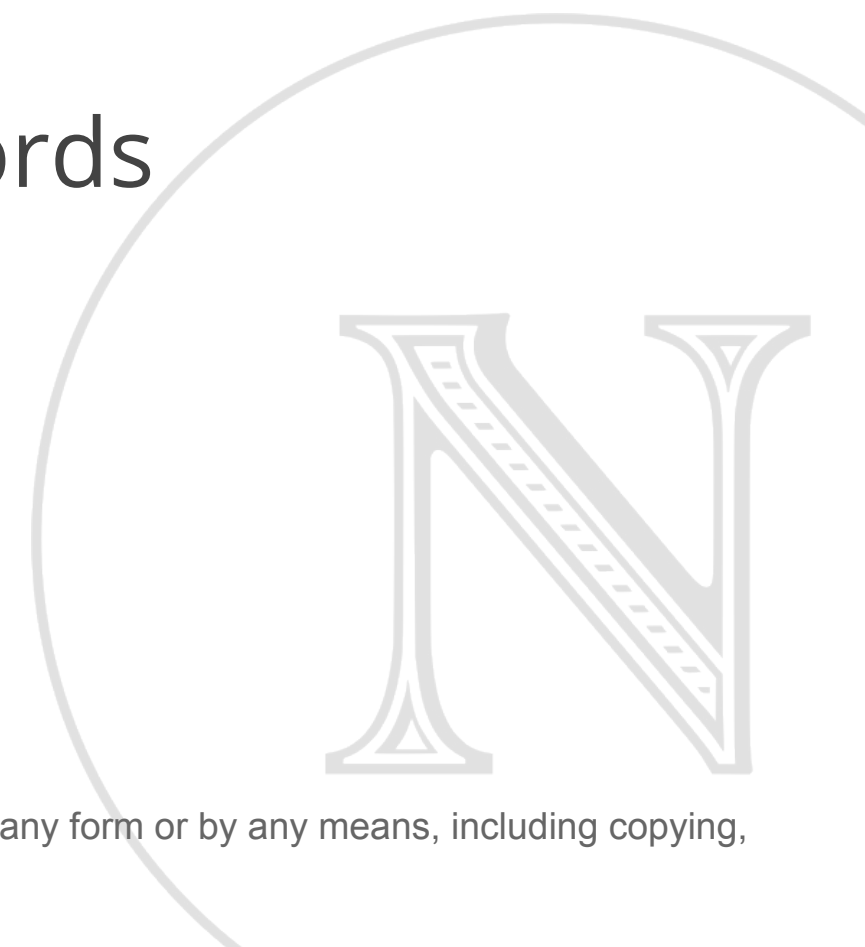
**Hello**  
Nice to meet you



# colors - keywords vs hex

**Color values can be represented 2 ways**

- **keywords**
  - E.g. red, blue, green, orangered, yellowgreen, ivory
  - [full list of keywords](#)
- **hex**
  - For more specific colors that don't have keywords
  - Can use a color-picker to get hex code
  - [color-picker tool](#)
- **rgba()**
  - For when you need opacity





# background - color

## HTML

```
<body>  
  Hello World  
</body>
```

## CSS

```
body {  
  background-color: gray;  
  color: white;  
}
```

## OUTPUT

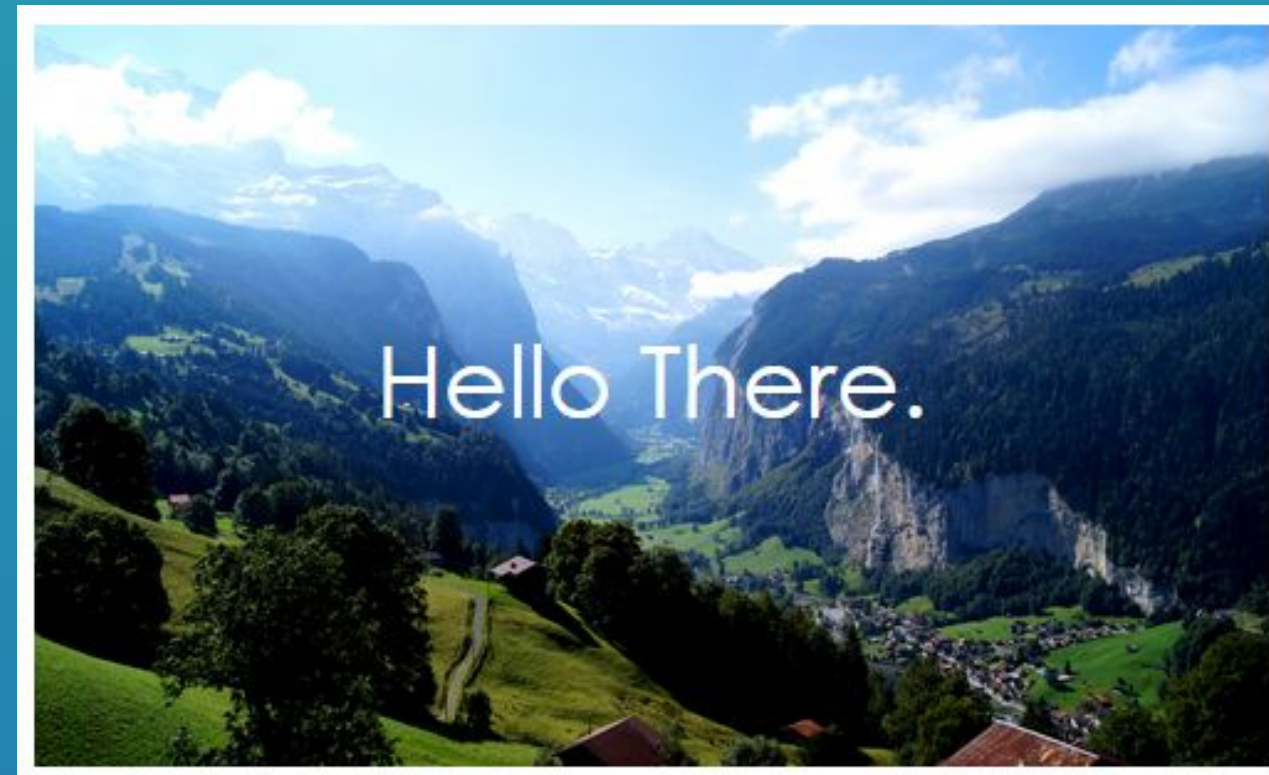
Hello World

# Bonus : Baby Got Background



NEXT  
ACADEMY

Backgrounds don't have to be boring plain colors  
**You can make image backgrounds**



**Hint:** Try the background-image property. But what value should it have? I'll let you figure that out.

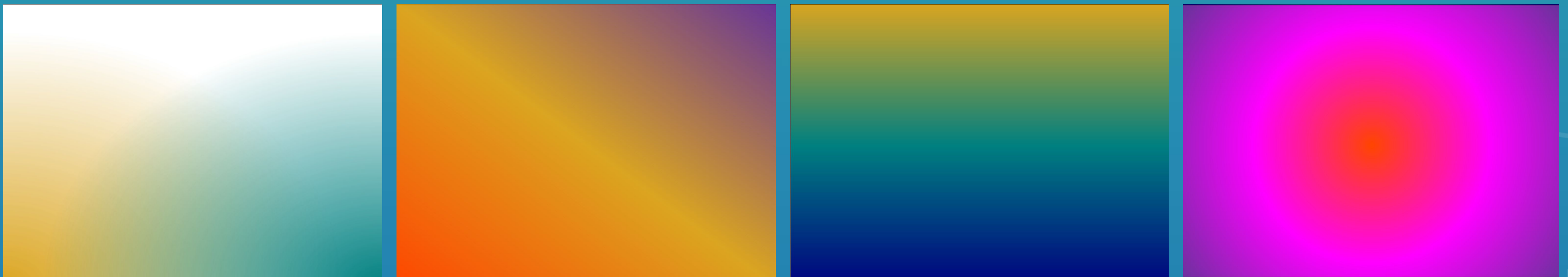
**Tip:** Here's a place to find nice background patterns: <http://subtlepatterns.com/>

# Bonus : Gradient backgrounds



NEXT  
ACADEMY

Why choose one color when you can have many?  
You can also add gradients of multiple colors as backgrounds.



**Hint:** Look up linear-gradient and radial-gradient. How many colors can a gradient have?



# Text

## Fonts and Alignment





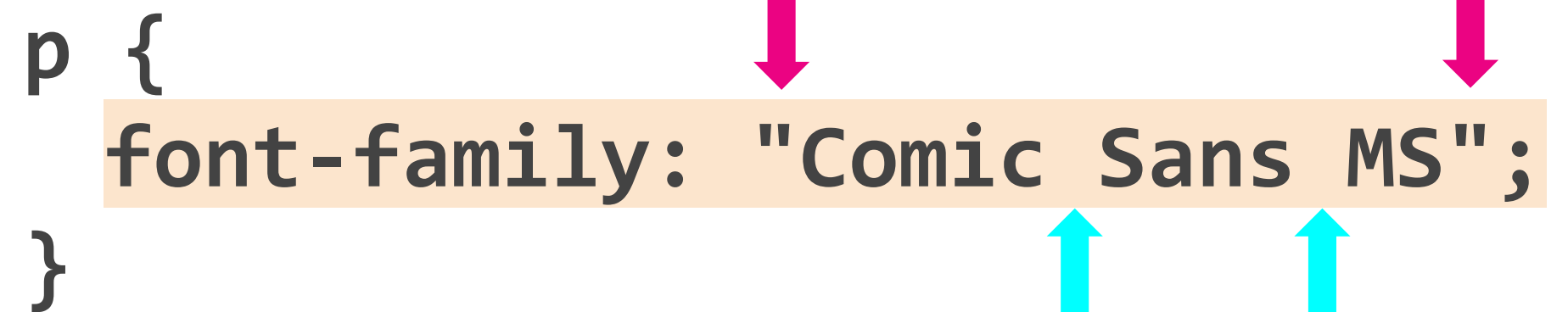
# fonts

## HTML

```
<h1> Hello </h1>  
  
<p> this is Comic  
Sans</p>
```

## CSS

```
p {  
  font-family: "Comic Sans MS";  
}
```

The diagram shows the CSS rule for the paragraph. Two pink arrows point down to the opening and closing double quotes of the font family name. Two cyan arrows point up to the spaces between the words 'Comic', 'Sans', and 'MS'.

MUST add " " if spaces exist





# fonts

## HTML

```
<h1> Hello </h1>  
  
<p> this is Comic  
Sans</p>
```

## CSS

```
p {  
  font-family: "Comic Sans MS";  
}
```

## OUTPUT

Hello

This is Comic Sans





NEXT  
ACADEMY

# SYSTEM FONTS



## Website Source Code

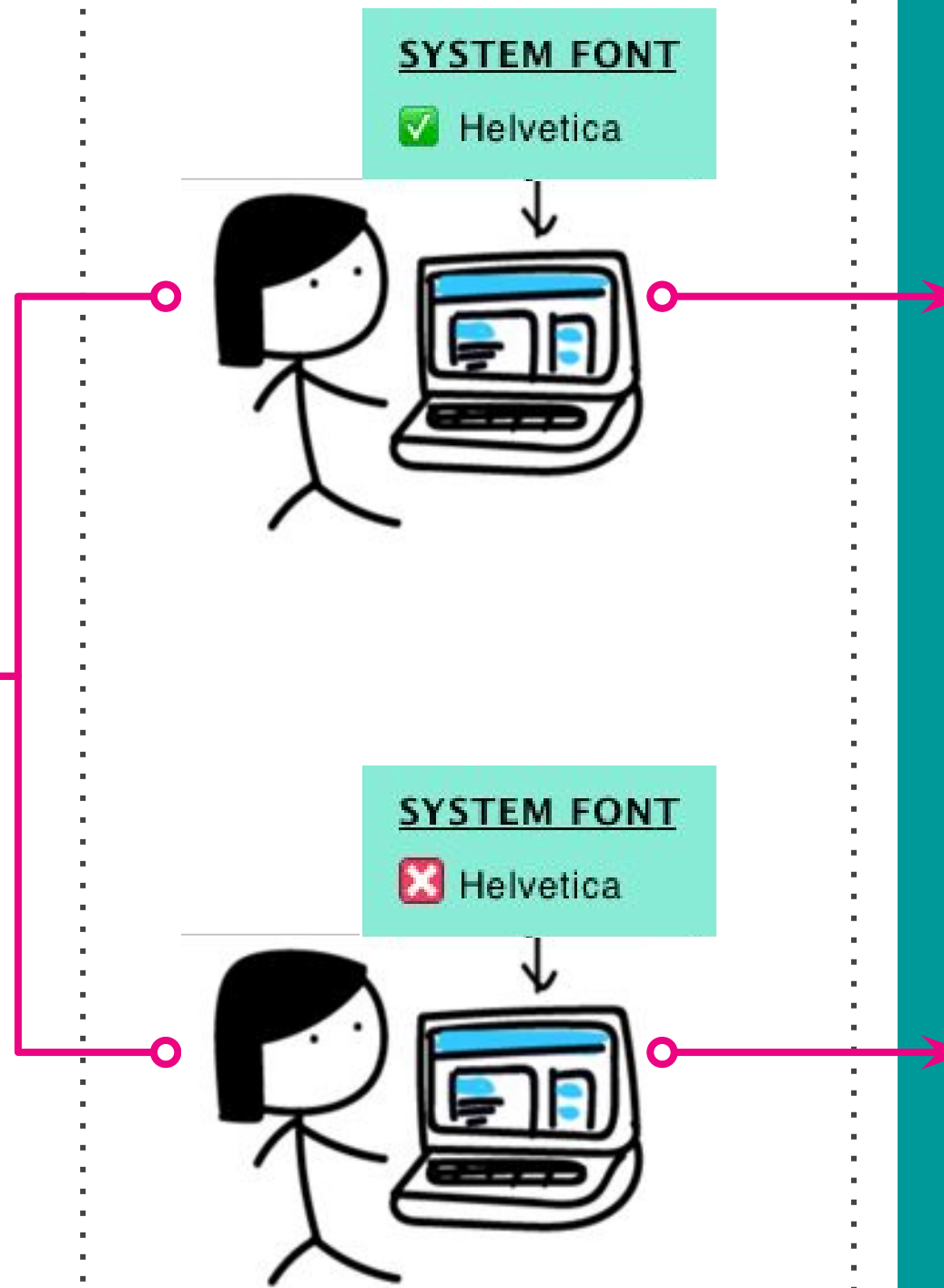
HTML

<h1> Hello </h1>

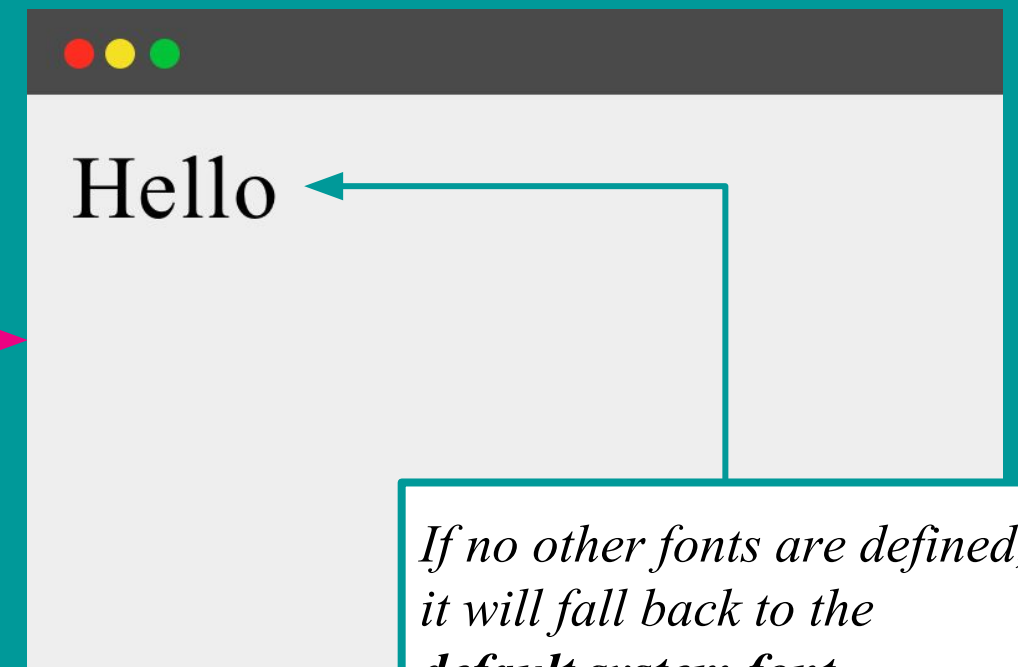
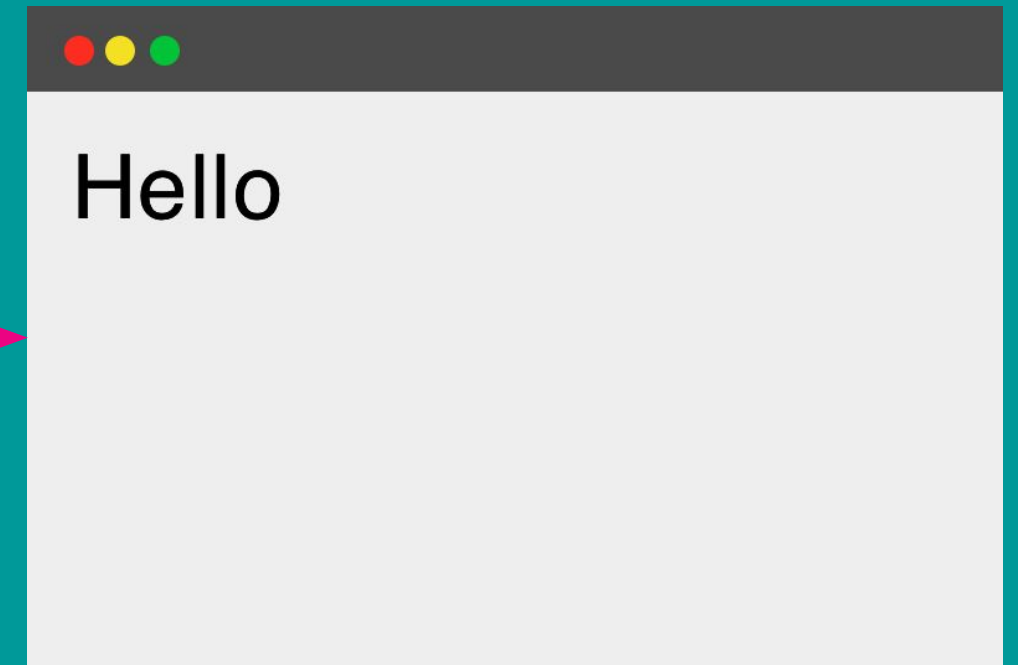
CSS

```
p {
  font-family: "Helvetica";
}
```

## Website Visitors



## What They See



*If no other fonts are defined,  
it will fall back to the  
default system font*

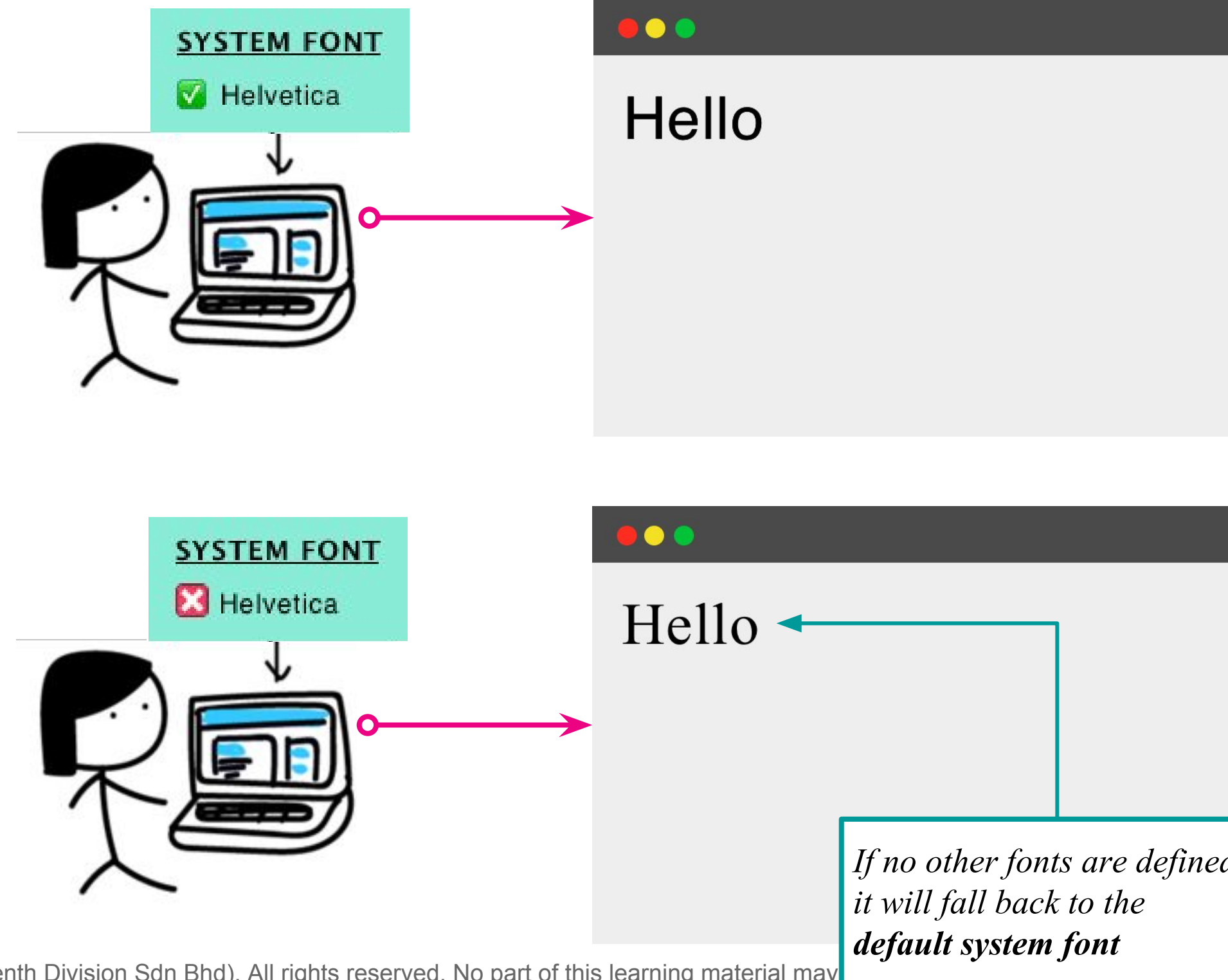


NEXT  
ACADEMY

# FONT FAMILY



# What If We Don't Want to Rely on Website Visitor's Local Computer?



# What If We Don't Want to Rely on Website Visitor's Local Computer?

- Host your own font file
- Use Web Font Services





# Using Google fonts

The screenshot shows the Google Fonts website interface. At the top, there's a navigation bar with 'Google Fonts', 'DIRECTORY', 'FEATURED', and 'ABOUT'. Below this, it says 'Viewing 818 of 818 font families'. The main content area displays several font families: Roboto (Christian Robertson, 12 styles), Asar (Sorkin Type, 1 style), UnifrakturCook (j. 'mach' wust, 1 style), and Lato (Łukasz Dziedzic, 10 styles). A modal window titled '1 Family Selected' is open, showing 'Your Selection' with 'Roboto' and a 'Clear All' button. Below this, there are 'EMBED' and 'CUSTOMIZE' tabs. The 'EMBED' tab is active, showing 'Embed Font' instructions and a code block with the following CSS code: 

```
<style>
@import url('https://fonts.googleapis.com/css?family=Roboto');
</style>
```

 There's also a section for 'Specify in CSS' with the code 

```
font-family: 'Roboto', sans-serif;
```

 and a link to the 'getting started guide'.

[link](#)



# Example

Add a Google Font is to a link to the font:

```
<head>  
  <link href='http://fonts.googleapis.com/css?family=Oswald'  
  rel='stylesheet' type='text/css'>  
</head>
```

Then use it in CSS:

```
body{  
  font-family: Oswald, Arial, sans-serif;  
}
```





# font-size

## HTML

```
<p> large text </p>
```

## CSS

```
p {  
  font-size: 72px;  
}
```

## OUTPUT

large text



# text-align

## HTML

```
<p> Oh my goodness this  
is centered! </p>
```

## OUTPUT

Default is aligned to left!





# text-align

## HTML

```
<p> Oh my goodness this  
is centered! </p>
```

## CSS

```
p {  
  text-align: center;  
}
```

## OUTPUT

Oh my goodness this is centered!

Other text-align options : **left**, **right**, and **justify**



# text-decoration

## HTML

```
<p> this is underlined!  
</p>
```

## CSS

```
p {  
  text-decoration: underline;  
}
```

## OUTPUT

this is underlined!

Check out [w3schools](https://www.w3schools.com) for more decoration options



# Spacing and Sizing



# Pixels

**Pixels are a unit of measurement for things on your screen**

- Unit name is **px**
- **1px** is the smallest blip of color on your screen

**Other units of measurements**

- **em, rem, %**



# width and height

## HTML

```

```

## CSS

```
img {  
  width: 150px;  
  height: 155px;  
}
```

## OUTPUT



**Note:** if you specify width, but not height, it will autoscale the height for you according to aspect ratio. Same goes for height.





# font-size

## HTML

```
<p>hello world</p>
```

## CSS

```
p {  
  font-size: 60px  
}
```

## OUTPUT

hello world



# The Importance of Whitespace

- Increases legibility
- Creates focus
- Sets the right tone
- Resources
  - <https://boagworld.com/design/why-whitespace-matters/>
  - <https://www.interaction-design.org/literature/article/the-power-of-white-space>





# padding, margin, and border





# padding

## HTML

```
<p> Hello there! </p>  
<p> How are you? </p>
```

## CSS

```
p {  
  padding: 12px;  
  background-color: #eeeeee;  
}
```

Without CSS

Hello there  
Hello there

With CSS

Hello there  
Hello there





# padding

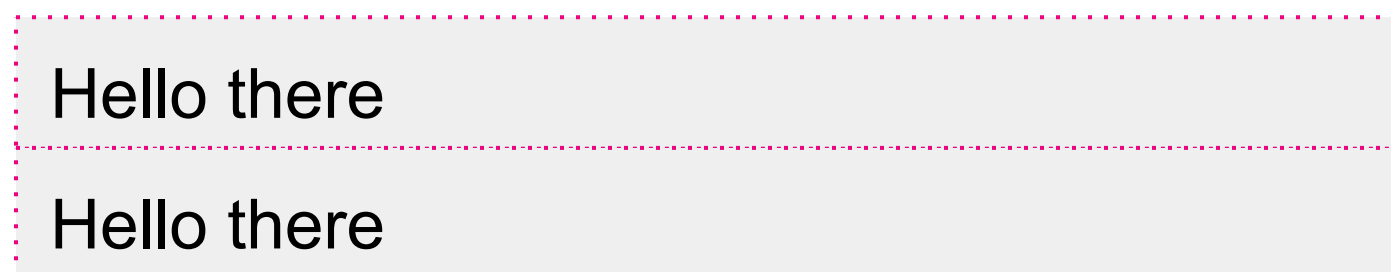
## HTML

```
<p> Hello there! </p>  
<p> How are you? </p>
```

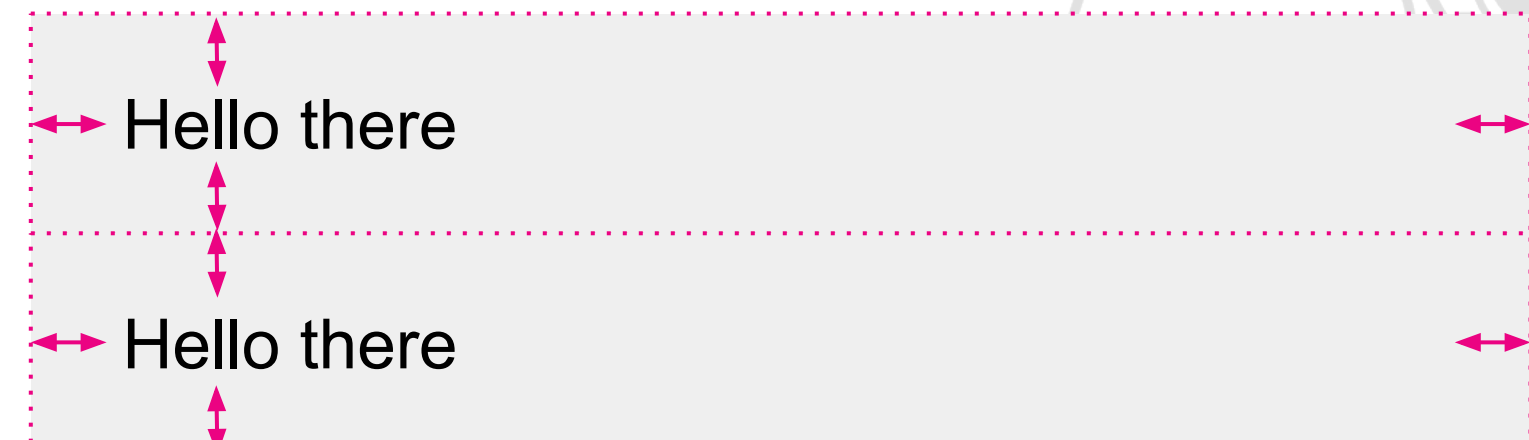
## CSS

```
p {  
  padding: 12px;  
  background-color: #eeeeee;  
}
```

Without CSS



With CSS





# margin

## HTML

```
<p> Hello there! </p>  
<p> How are you? </p>
```

## CSS

```
p {  
  margin: 18px;  
  background-color: #eeeeee;  
}
```

Without CSS

Hello there  
Hello there

With CSS

Hello there  
  
Hello there





# margin

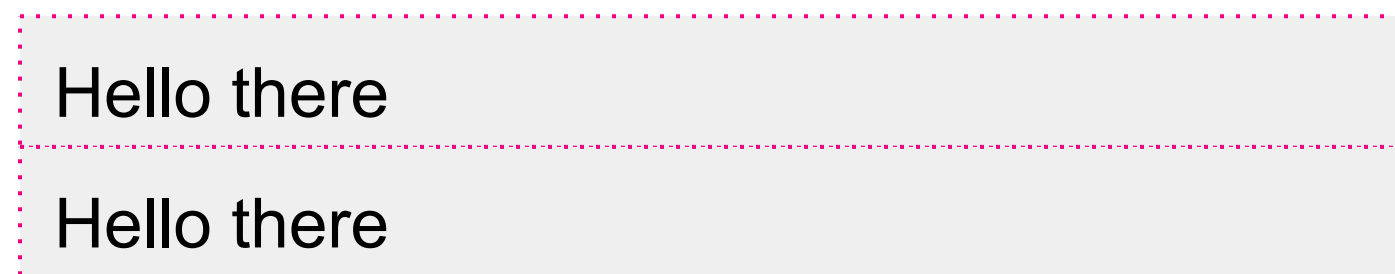
## HTML

```
<p> Hello there! </p>  
<p> How are you? </p>
```

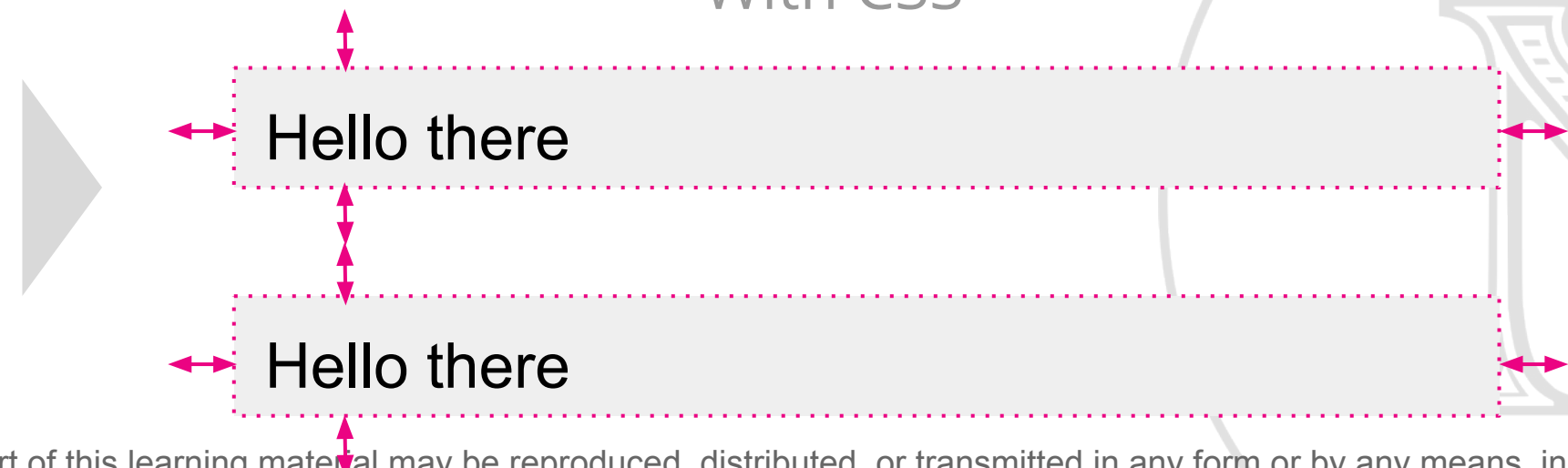
## CSS

```
p {  
  margin: 18px;  
  background-color: #eeeeee;  
}
```

Without CSS



With CSS



# border

## HTML

```

```

## CSS

```
img {  
  border: 3px solid black;  
}
```

↑  
*width*

↑  
*style*

↑  
*color*

Without CSS



With CSS







# border-radius

## HTML

```

```

## CSS

```
img {  
  border-radius: 12px;  
  width: 24px;  
  height: 24px;  
}
```

Without CSS



With CSS



# border-radius

## HTML

```

```

## CSS

```
img {
  border-radius: 12px;
  width: 24px;
  height: 24px;
}
```

**50%**

Without CSS



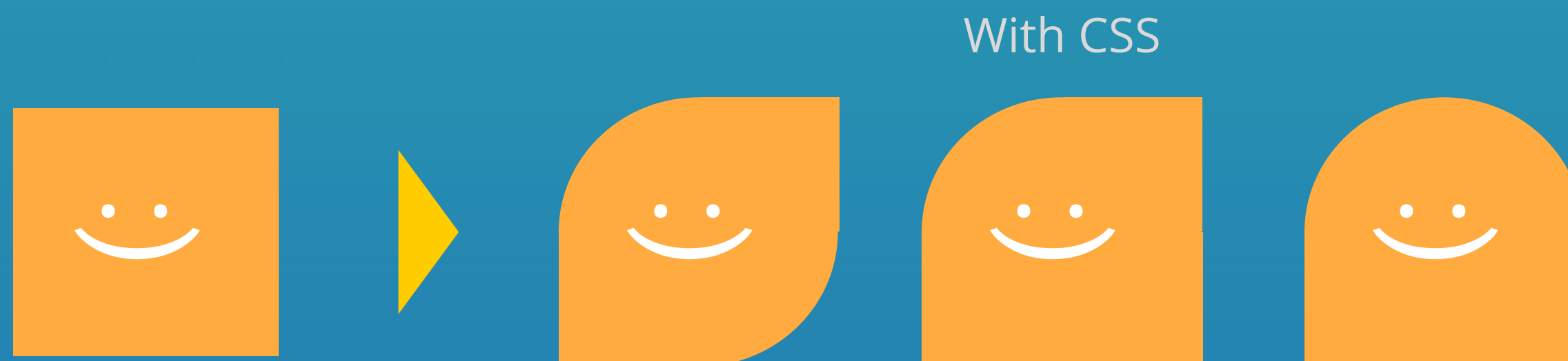
With CSS





# Bonus : Give Me Funky Edges

Instead of 4 rounded edges, be creative! Try the 3 styles below!  
The challenge: Create each style with only 1 property.



**Hint:** How many values can the property “border-radius” hold?





# Selector Specificity and Overrides

Selecting specific elements and overriding styles





# Selector Specificity

- **Select more than one tag**

```
p, h1 {  
    color : blue;  
}
```

- **Select only one type of tag in another**

```
header p {  
    color : red;  
}
```





# Selector Overrides I

## Rule of Thumb

Browser will follow rules with the **most specific** selector

```
<p>Hello</p>  
  
<header>  
  <p>Goodbye</p>  
</header>
```

```
p {  
  color: red;  
}  
header p {  
  color: blue;  
}
```



Hello

Goodbye





# Selector Overrides II

In case of duplicate selectors,  
browser will follow the one that **comes last**

```
<p> Hello </p>
```

```
p {  
  color: red;  
}  
p {  
  color: green;  
}
```



Hello





# Want to Select Just One Thing?

## HTML

```
<p id="error">look at me</p>  
<p>some text</p>  
<p>not so important</p>
```

## CSS

```
#error {  
  color: red;  
}
```



look at me

some text

not so important







# Want to Select a Type of Thing?

## HTML

```
<p class="highlight">fun</p>  
<p>boring text</p>  
<p class="highlight">see me</p>
```

## CSS

```
.highlight {  
  background-color: yellow;  
  text-transform: italic  
}
```

*fun*

boring text

*see me*





# More Properties

**There are about 170+ CSS properties**

**Complete list of CSS properties**

- <http://www.w3schools.com/cssref/>





NEXT  
ACADEMY

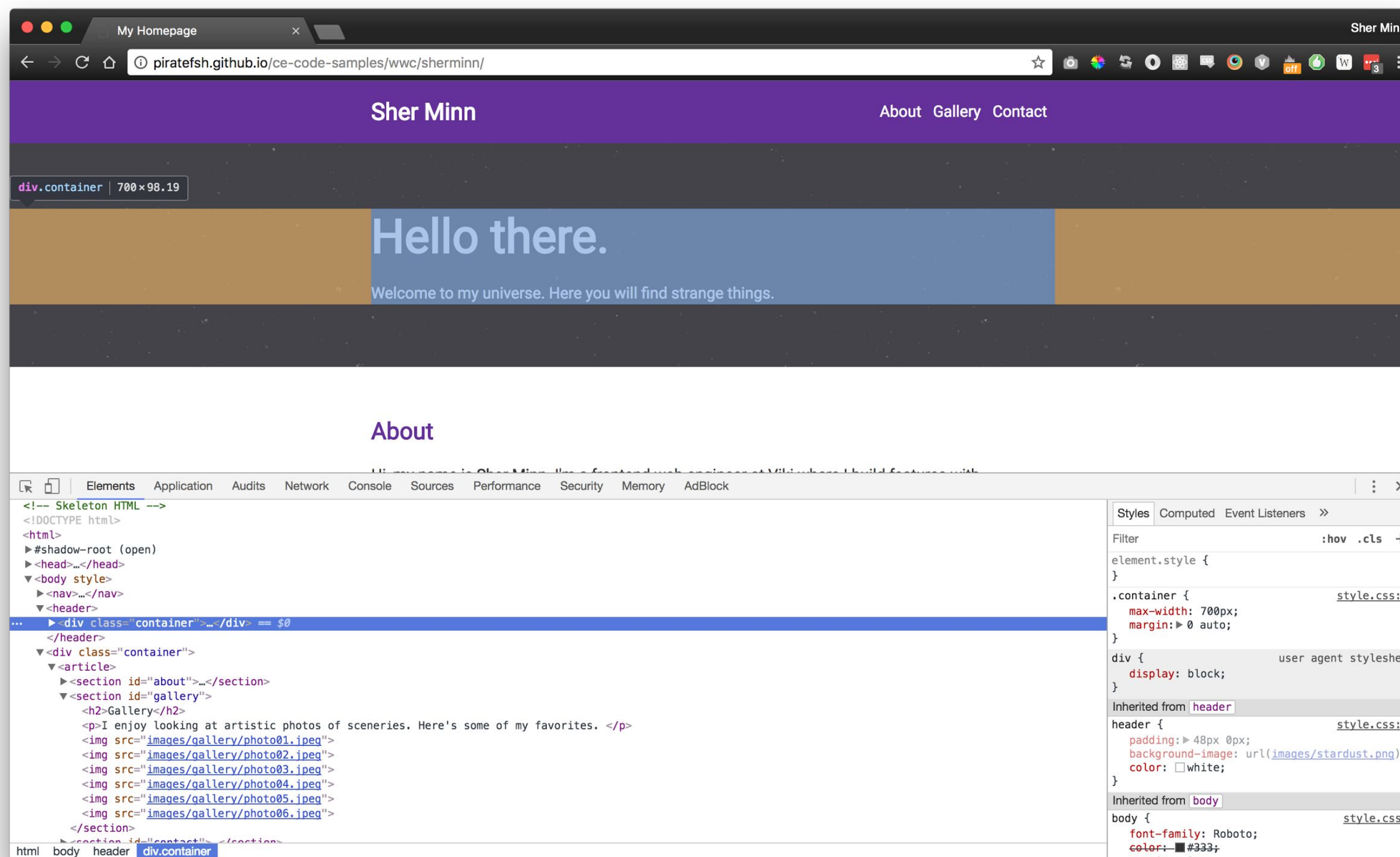
# Developer Tools

Or, how to steal code





# What are Developer Tools?





# Recap

- Introduction to CSS
  - colors, text, spacing and sizing, etc
  - selector overrides
- Chrome developer tools
  - or how to steal code





NEXT  
ACADEMY

# The End

Thank you for coming!

