Welcome to the CSS Lecture

The session will start shortly...

CoGrammar

Johannesburg Team Housekeeping

- Please be mindful and respectful to everyone in this supportive learning environment. Mutual respect and tolerance are fundamental values we uphold.
- There are no bad or silly questions—feel free to ask anything! You can ask Sashlin or myself questions at any time, regardless of the situation. Even if you find yourself in a dire situation—like stuck in quicksand—you're still welcome to ask us a question (though we recommend calling or shouting first!).
- A few additional reminders for onsite behavior:
  - Keep shared spaces tidy—clean up after yourself in the break areas.
  - Please mute your devices during sessions to minimize distractions.
  - Avoid making personal phone calls in common areas—use designated quiet zones if you need to step away.
- Additionally, please remember to put any dishes in the sink before 2 p.m., as Lizbeth will have already finished for the day. If you’re feeling unwell, kindly inform Ingrid or myself via email.

CSS

CSS is the technology that brings visual life to websites. Before diving in, it's important to understand that CSS works hand-in-hand with HTML. While HTML provides structure, CSS provides style. This relationship is fundamental to web development.

CoGrammar

What is a CSS ?

CSS is an acronym for Cascading Style Sheets, let’s have a look at what each word represents:

- ***"Cascading"*** refers to how styles can inherit and override each other in a hierarchical manner
- ***"Style"*** means the visual properties we can modify
- ***"Sheets"*** indicates that we write these rules in separate sheets or sections

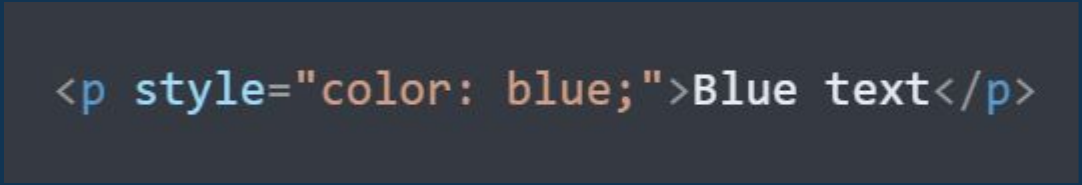
CoGrammar

Three Ways to Add CSS

- ***"Inline CSS"*** Added directly to HTML elements using the style attribute
- ***"Internal CSS"*** Added in the *<head>* section of HTML document
- ***"External CSS"*** Stored in separate .css files

CoGrammar

Inline CSS

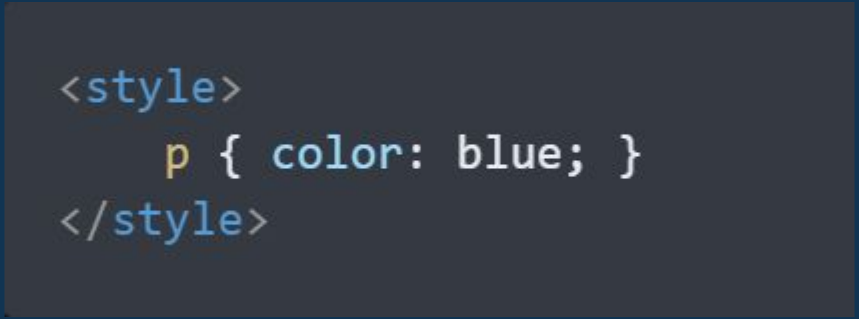


```html
<p style="color: blue;">Blue text</p>
```

1. Highest specificity (always overrides other styles)
2. Useful for quick tests or unique cases
3. Not recommended for regular use as it mixes content with presentation

CoGrammar

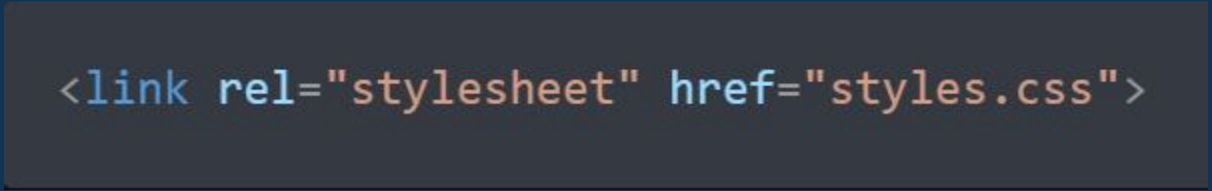
Internal CSS



```
<style>
    p { color: blue; }
</style>
```

1. Affects only the current page
2. Useful for single-page websites
3. Better than inline but still mixes CSS with HTML

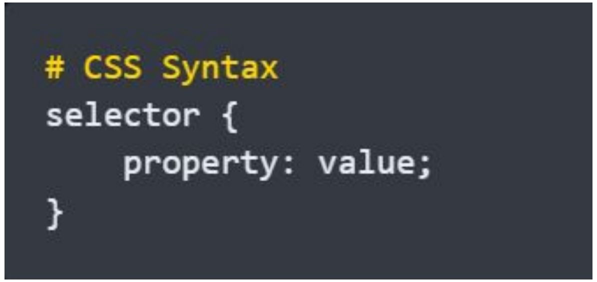
External CSS



```html
<link rel="stylesheet" href="styles.css">
```

1. Linked to HTML using tag
2. Best practice for most websites
3. Allows for better maintenance and caching

CSS Syntax

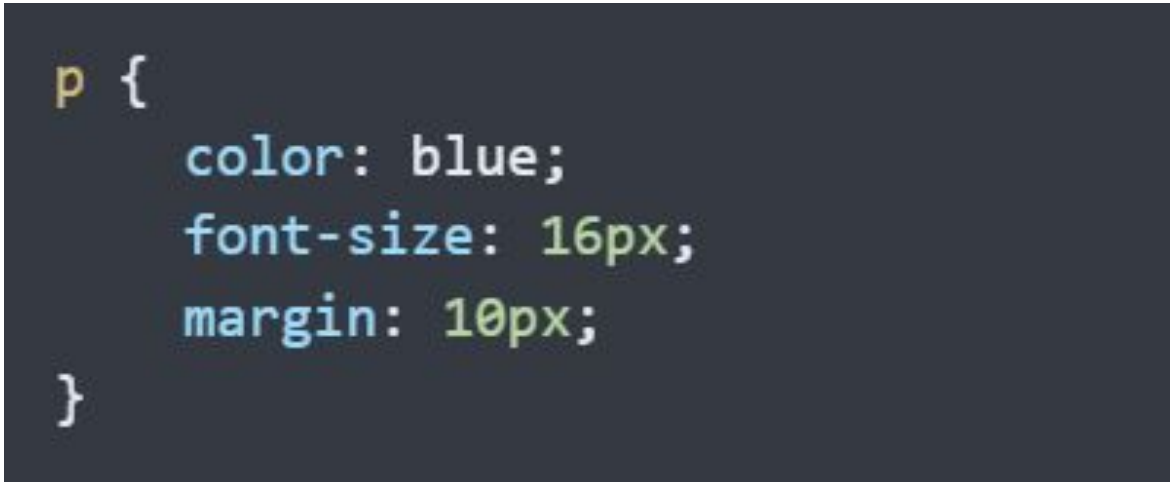


```
# CSS Syntax
selector {
    property: value;
}
```

- **Selector:**
  - Targets HTML elements you want to style
- **Property:**
  - The aspect you want to change
- **Value:**
  - The specific setting for the property

CoGrammar

CSS Syntax example



```css
p {
    color: blue;
    font-size: 16px;
    margin: 10px;
}
```

CoGrammar

Example Breakdown

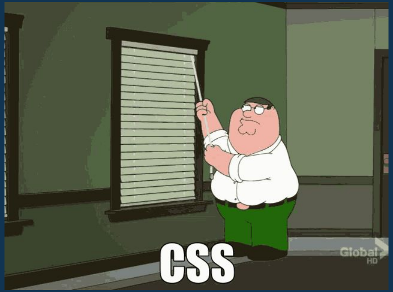
- p
  - targets all paragraph elements
- *color:*
  - blue makes the text blue
- *font-size:*
  - 16px sets text size to 16 pixels
- *margin:*
  - 10px adds 10 pixels of space around the paragraph

CoGrammar

Selectors

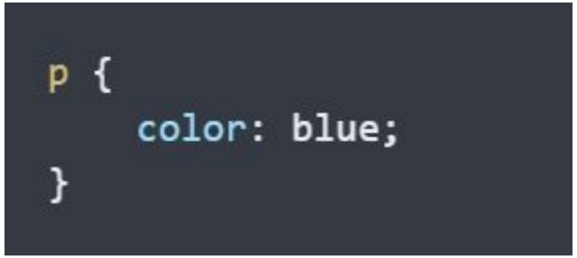
Element selectors are the simplest type of CSS selector, used to style all occurrences of a specific HTML element on a webpage.

For example, using p { color: blue; } would turn the text of all <p> elements blue.



CoGrammar

Basic Element Selector

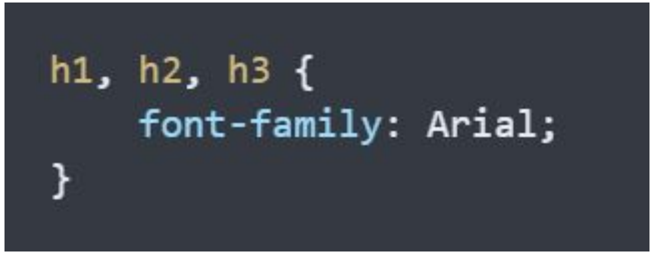


```
p {
      color: blue;
}
```

- Targets all <p> elements
- Lowest specificity
- Good for setting default styles

CoGrammar

Multiple Element Selector

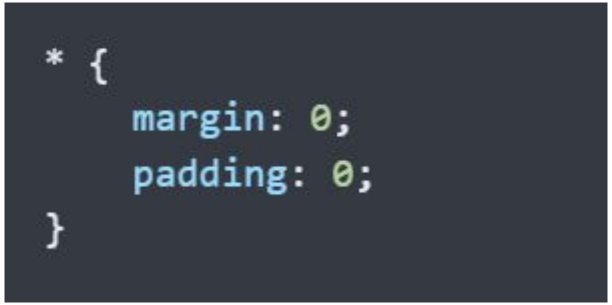


```
h1, h2, h3 {
    font-family: Arial;
}
```

- Targets multiple elements at once
- Reduces code repetition
- All selected elements get the same styles

CoGrammar

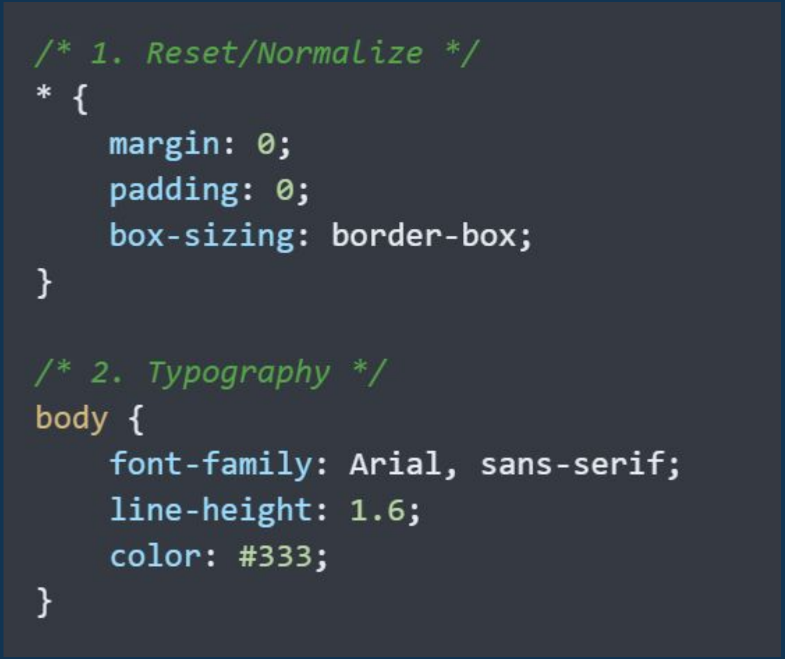
Universal Selector



```css
* {
    margin: 0;
    padding: 0;
}
```

- Targets every element
- Used for CSS resets
- Use sparingly due to performance

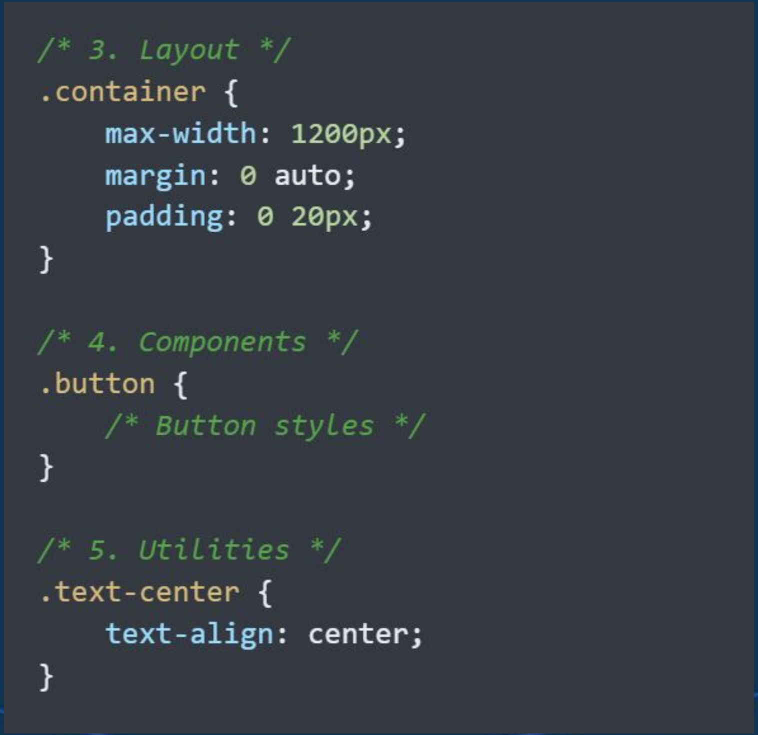
Code Organization



```css
/* 1. Reset/Normalize */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

/* 2. Typography */
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    color: #333;
}
```

Code Organization


```css
/* 3. Layout */
.container {
    max-width: 1200px;
    margin: 0 auto;
    padding: 0 20px;
}

/* 4. Components */
.button {
    /* Button styles */
}

/* 5. Utilities */
.text-center {
    text-align: center;
}
```

CoGrammar

Best Practices Cont.

1. **Start with resets or normalizers:**

   Begin by using a CSS reset or normalize to remove browser-specific defaults, ensuring consistent styles across all browsers.

2. **Define global typography:**

   Set the base font styles like family, size, and color to create a consistent look for text across the entire site.

3. **Set up layout containers:**

   Organize the layout by creating containers (header, footer, sections) with consistent spacing and alignment for a clean design.

Best Practices Cont.

4. **Style individual components:**

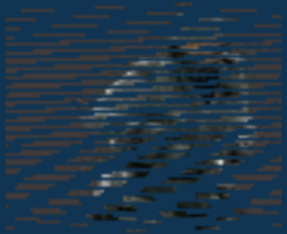
   Customize elements like buttons, forms, and cards after setting up the layout to give each a unique, polished appearance.

5. **Add utility classes last:**

   Finally, add small, reusable utility classes for quick, specific adjustments without cluttering the code.

CoGrammar

Q & A SECTION



Please use this time to ask any questions relating to the topic, should you have any.

Thank you for attending