

css

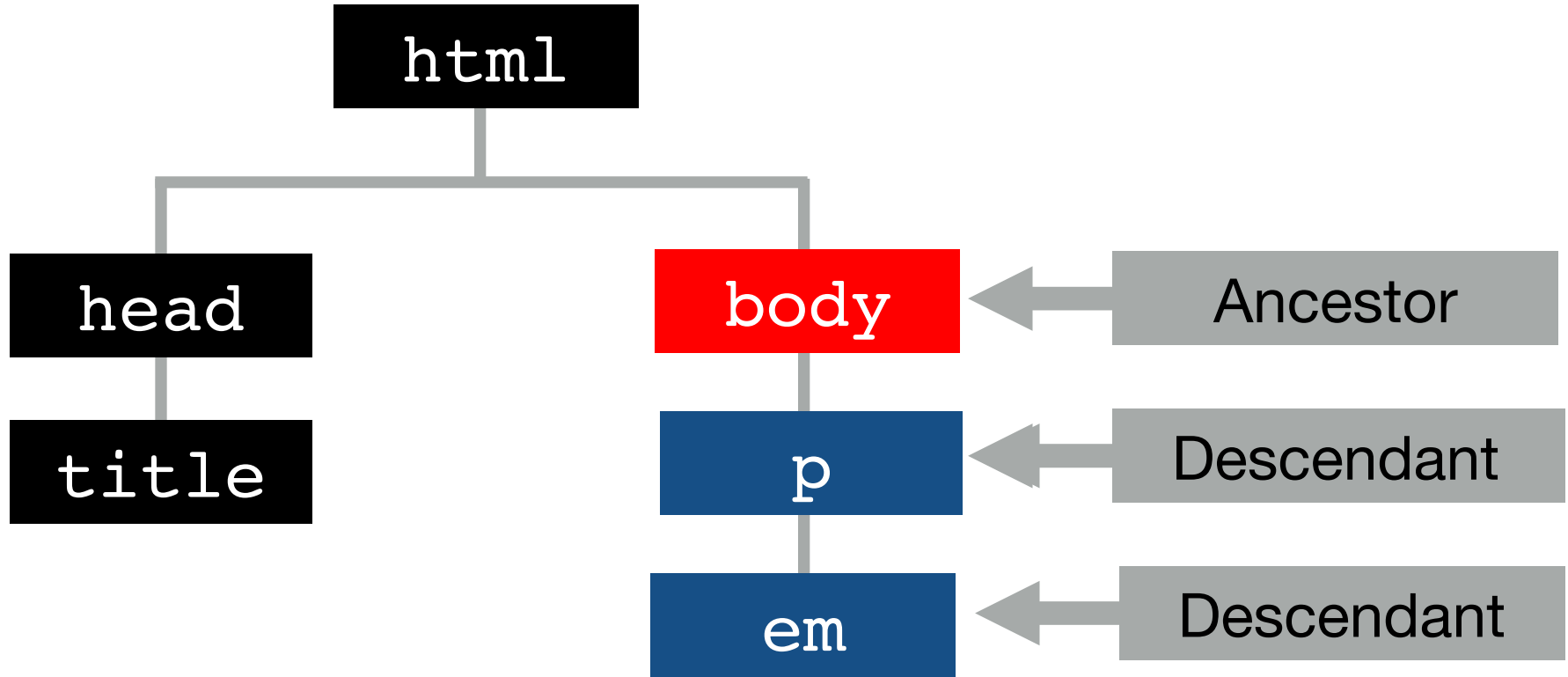
INHERITANCE

What is inheritance?

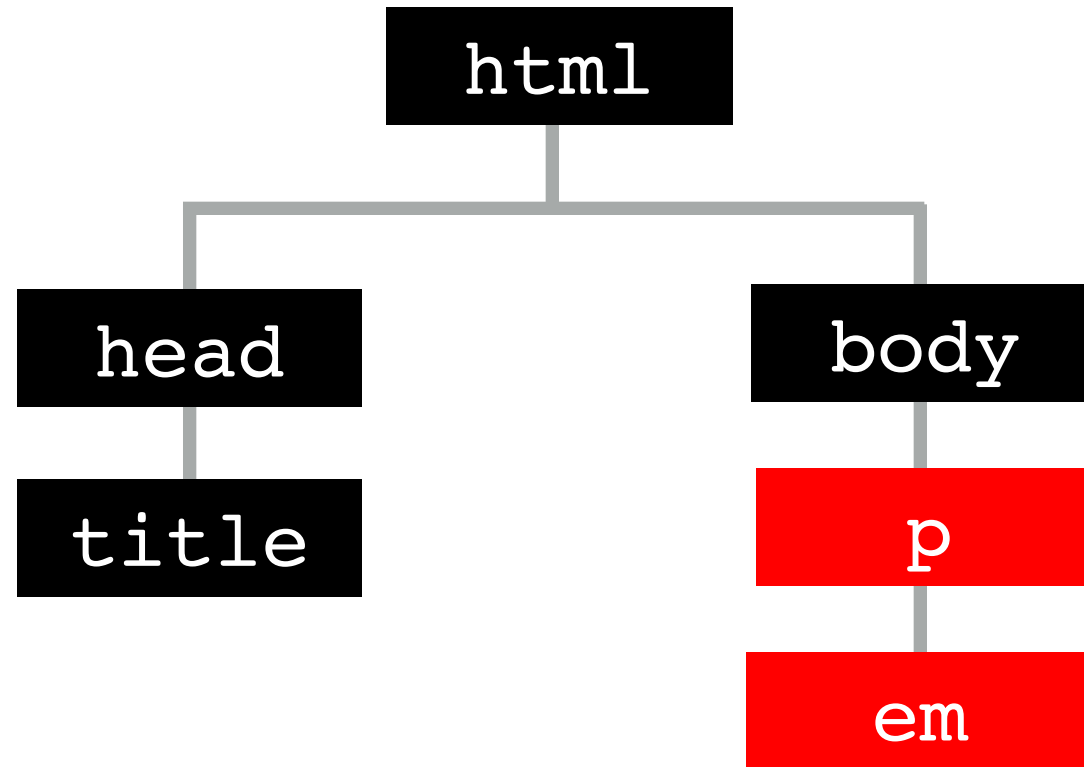
Inheritance is where CSS properties are **passed down from parent elements to descendant elements.**

```
/* Examples of CSS properties */  
p { font-family: arial, sans-serif; }  
p { font-size: 80%; }  
p { font-weight: bold; }  
p { text-align: right; }  
p { color: red; }  
p { line-height: 1.4; }
```

In the following **document tree diagram**, the `<body>` element is the ancestor. The `<p>` and `<em>` elements are descendants.



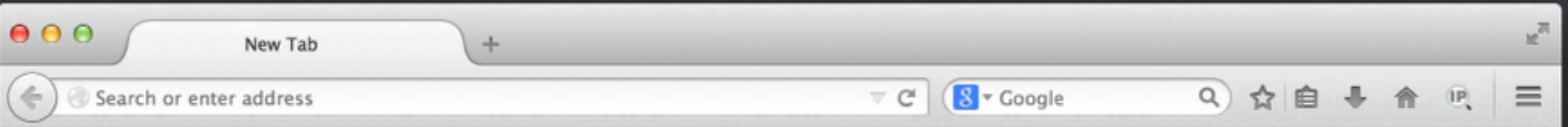
To see inheritance in action, we will look at an `<em>` element inside a `<p>` element.





This would produce **the following HTML markup:**

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple HTML page</title>
</head>
<body>
  <p>
    Lorem <em>ipsum</em> dolor etuer.
  </p>
</body>
</html>
```

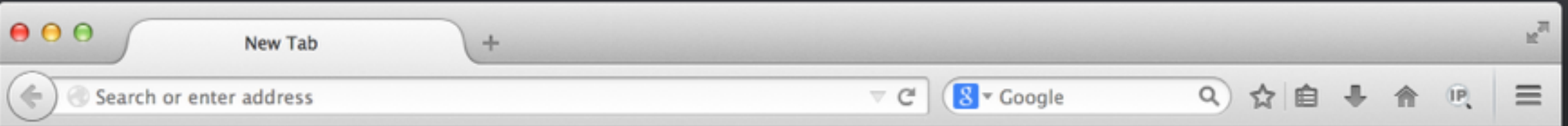


Lorem *ipsum* dolor etuer.

What would happen if we styled the `<p>` element to **“color: red”**?

```
/* basic CSS rule */  
p { color: red; }
```

In a browser, the `<p>` and `<em>` elements will **both be colored red.**



Lorem *ipsum* dolor etuer.

But why is the `<em>` element colored red when it hasn't been given **any style of its own**?



Because the `<em>` element (the child element) has **inherited the color property** from the `<p>` element (the parent element).

Exercise 1:  
color red

Why is it helpful?

Inheritance is designed to **make it easier for authors**. Otherwise we would need to style all descendant elements as well as ancestors.

```
/* if inheritance didn't exist */  
p { color: red; }  
p em { color: red; }
```

CSS files would be **much larger in size**, harder to create and harder to maintain.

All properties?

Many CSS properties are not inherited. If every CSS property was inherited, it would make things **much harder** for authors.

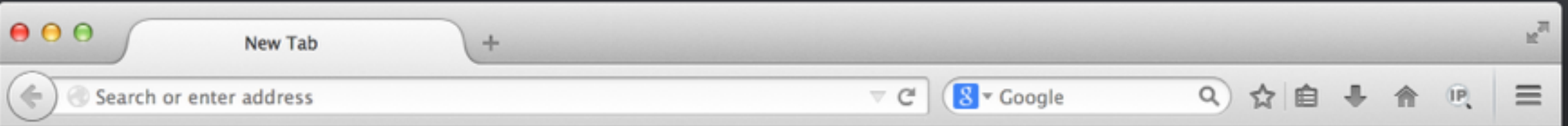


Authors would have to **turn off unwanted CSS properties** for descendant elements.

For example, what would happen if the **border property was inherited by default**, and we then applied a border to the <p> element?

```
/* adding border to the <p> element */  
p { border: 1px solid red; }
```

The `<em>` inside the `<p>` would **also have a red border** - even though we may not want this to occur.



Lorem *ipsum* dolor etuer.

Luckily, the **border property is not inherited**, so the `<em>` would not have a red border.

Generally speaking, only properties that **make our job easier** are inherited!

Exercise 2:  
border red



Which properties?

The following **CSS 2.1 properties**  
are inherited:

azimuth, border-collapse, border-spacing, caption-side, color, cursor, direction, elevation, empty-cells, font-family, font-size, font-style, font-variant, font-weight, font, letter-spacing, line-height, list-style-image, list-style-position, list-style-type, list-style, orphans, pitch-range, pitch, quotes, richness, speak-header, speak-numeral, speak-punctuation, speak, speech-rate, stress, text-align, text-indent, text-transform, visibility, voice-family, volume, white-space, widows, word-spacing

This list might seem a bit overwhelming, so let's focus on some of the **key groups of properties.**

All **text-related properties** are inherited.

azimuth, border-collapse, border-spacing,  
caption-side, color, cursor, direction, elevation,  
empty-cells, font-family, font-size, font-style, font-  
variant, font-weight, font, letter-spacing, line-  
height, list-style-image, list-style-position, list-  
style-type, list-style, orphans, pitch-range, pitch,  
quotes, richness, speak-header, speak-numeral,  
speak-punctuation, speak, speech-rate, stress,  
text-align, text-indent, text-transform, visibility,  
voice-family, volume, white-space, widows, word-  
spacing

All **list-related properties** are inherited.

azimuth, border-collapse, border-spacing, caption-side, color, cursor, direction, elevation, empty-cells, font-family, font-size, font-style, font-variant, font-weight, font, letter-spacing, line-height, **list-style-image**, **list-style-position**, **list-style-type**, **list-style**, orphans, pitch-range, pitch, quotes, richness, speak-header, speak-numeral, speak-punctuation, speak, speech-rate, stress, text-align, text-indent, text-transform, visibility, voice-family, volume, white-space, widows, word-spacing



And, importantly, the **color property** is also inherited.

azimuth, border-collapse, border-spacing,  
caption-side, **color**, cursor, direction, elevation,  
empty-cells, font-family, font-size, font-style, font-  
variant, font-weight, font, letter-spacing, line-  
height, list-style-image, list-style-position, list-  
style-type, list-style, orphans, pitch-range, pitch,  
quotes, richness, speak-header, speak-numeral,  
speak-punctuation, speak, speech-rate, stress,  
text-align, text-indent, text-transform, visibility,  
voice-family, volume, white-space, widows, word-  
spacing

font-size inheritance

The font-size property is inherited in a different way to many other properties. Rather than the actual value being inherited, the **calculated value** is inherited.

Before explaining how font-size inheritance works, we need to look at **why font-size is not directly inherited.**

By default, the body element has a font-size of approximately 16px. What would happen if we set the `<p>` element to a **font-size of 80%**?

```
/* setting the font-size */  
p { font-size: 80%; }
```

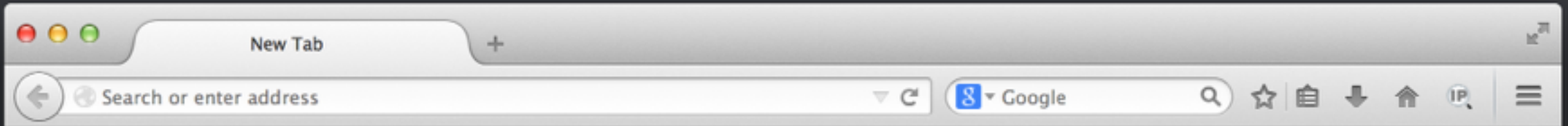
The <p> element would be 80% x 16px which is **a font-size of 12.8px.**



# font-size table

element	percent	initial value	final value
<body>	100%	16px	16px
<p>	80%	16px	12.8px

If the font-size value of 80% were to be inherited, the `<em>` would be sized to **80% of the `<p>` element** and would be a font-size of 10.24px.



Lorem *ipsum* dolor etuer.



`<em>` element smaller than other content

However, this is not the case! The `<em>` is the same size as the `<p>`.  
So **how does inheritance work for font-size?**

The answer is that **the calculated value is inherited**, rather than the value itself. Let's look at three examples using the same HTML code as before:

# Example 1: Pixels

In the following example, the `<p>` element has been given **a font-size of 14px.**

```
/* font-size in pixels */  
p { font-size: 14px; }
```



The pixel value (14px) overrides the browser's default font-size value (16px). This **new calculated value (14px)** is inherited by descendants.

So, the `<em>` element will have **a font-size of 14px.**

# pixel example

element	value	calculated value
<body>	unspecified	16px
<p>	14px	14px
<em>	unspecified	inherited value = 14px

Example 2:  
percentage

In the following example, the <p> element has been given **a font-size of 80%**.

```
/* font-size in percents */  
p { font-size: 80%; }
```

The percentage value (80%) is applied to the browser's default font-size value (16px). This **new calculated value (80% x 16px = 12.8px)** is inherited by descendants.

So, the `<em>` element will have **a font-size of 12.8px.**



# percentage example

element	value	calculated value
<body>	unspecified	16px
<p>	85%	16px x 80% = 12.8px
<em>	unspecified	inherited value = 12.8px

Example 3: em

In the following example, the `<p>` element has been given **a font-size of .7em.**

```
/* font-size in ems */  
p { font-size: .7em; }
```

The em value (.7em) is applied the browsers default font-size value (16px). This **new calculated value (.7 x 16px = 11.2px)** is inherited by descendants.

So, the `<em>` element will have **a font-size of 11.2px.**

# em example

element	value	calculated value
<body>	unspecified	16px
<p>	7em	16px x 7em = 11.2px
<em>	unspecified	inherited value = 11.2px

Using inheritance  
for **efficiency**



Authors can use inheritance to **write efficient CSS**. For example, you can set the font-size and font-family on the body element. These properties will be inherited by all descendant elements.

In the following example, the **<body> element** has been given font-family and font-size values.

```
/* inheritance example */  
body  
{  
    font-family: arial, helvetica, sans-serif;  
    font-size: 90%;  
}
```

You can then override the properties as needed. In the following example, the **heading elements** have been given a different font-family.

```
/* inheritance example */  
body  
{  
    font-family: arial, helvetica, sans-serif;  
    font-size: 90%;  
}  
  
h1, h2, h3, h4, h5, h6  
{  
    font-family: georgia, times, serif;  
}
```

In the following example, the `<h1>` **element** has been given a unique font-size.

```
/* inheritance example */  
body  
{  
    font-family: arial, helvetica, sans-serif;  
    font-size: 90%;  
}  
  
h1, h2, h3, h4, h5, h6  
{  
    font-family: georgia, times, serif;  
}  
  
h1 { font-size: 220%; }
```

# Exercise 3: font-size and font- family





# Russ Weakley

Max Design

**Site:** [maxdesign.com.au](http://maxdesign.com.au)

**Twitter:** [twitter.com/russmaxdesign](https://twitter.com/russmaxdesign)

**Slideshare:** [slideshare.net/maxdesign](https://slideshare.net/maxdesign)

**Linkedin:** [linkedin.com/in/russweakley](https://linkedin.com/in/russweakley)

Title S

Lesson

Lesson2

Section hi

Action

body highlight

body smaller

quote url

code c-h c-g

label

T