COMP4021
Internet Computing

More on CSS and jQuery

Gibson Lam and David Rossiter

CSS and jQuery

- You have learned the basic CSS selectors:
  - Element selectors, i.e. `h1 { ... }`
  - id selectors, i.e. `#name { ... }`
  - Class selectors, i.e. `.highlight { ... }`
- You will learn a few other common CSS selectors in this presentation
- You will find them useful in both style sheets and selecting things in jQuery

The Meals Example

```html
<!DOCTYPE html>
<html>
<head>
  <title>The Meals</title>
</head>
<body>
  <h1>The Meals</h1>
  <ol>
    <li>Breakfast <b>$15.00</b></li>
    <li>Lunch <b>$25.00</b></li>
    <li>Dinner <b>$50.00</b>
      <ul>
        <li>Main course <b>$30.00</b></li>
        <li>Desert <b>$20.00</b></li>
      </ul>
    </li>
  </ol>
</body>
</html
```

The Meals

1. Breakfast **$15.00**
2. Lunch **$25.00**
3. Dinner **$50.00**
   - Main course **$30.00**
   - Desert **$20.00**

- You have seen a similar example before
- This page will be used by the examples later

Selecting Multiple Things

- You previously use only one selector in CSS rules

- You use multiple selectors by simply separating them using commas, i.e.:

```
h1, b { color: red; }
```

- This example changes **all** <h1> and <b> to use red colour

The Meals

1. Breakfast **$15.00**
2. Lunch **$25.00**
3. Dinner **$50.00**
   - Main course **$30.00**
   - Desert **$20.00**

```
<h1>The Meals</h1>
<b>$15.00</b>
<b>$25.00</b>
<b>$50.00</b>
<b>$30.00</b>
<b>$20.00</b>
```

Selecting Things Under a Parent

- You can change things under a parent using two selectors side by side (no comma), e.g.:

```
ul b { color: red; }
```

1. Breakfast **$15.00**
2. Lunch **$25.00**
3. Dinner **$50.00**
   ○ Main course **$30.00**
   ○ Desert **$20.00**

- The example changes all `<b>` that are under `<ul>` to red

```
<ul>

   <li>Main course <b>$30.00</b> </li>

   <li>Desert <b>$20.00</b> </li>

</ul>
```

Under a Parent – Example 1

`ul b { color: red; }`

The parent

These are not affected by the CSS rule

b
Red!

b
Red!
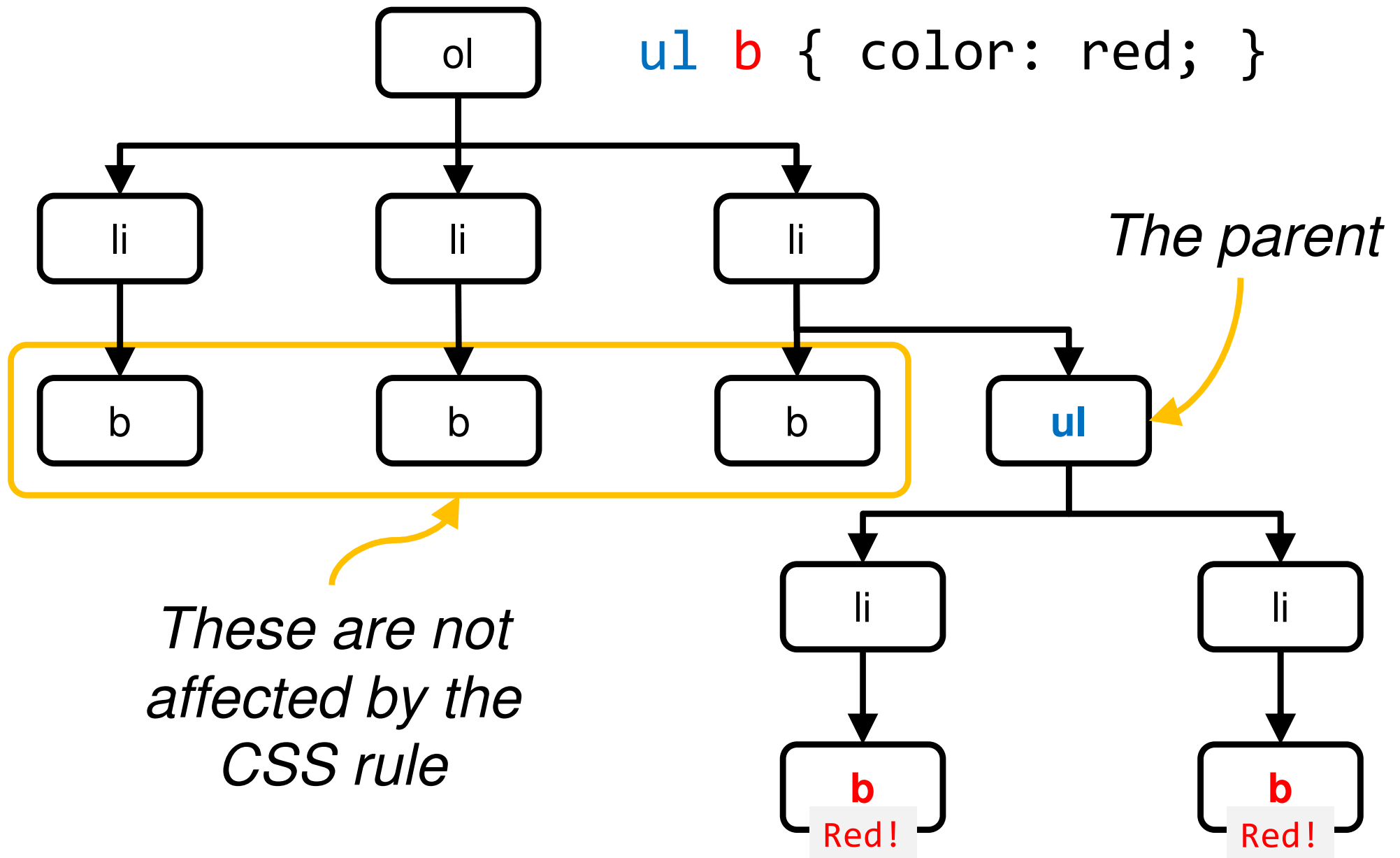
Be Careful!

- Be careful! The selector in the previous slide affects the children, grandchildren, grand grandchildren… under the parent so let's see another example:

```
ol b { color: red; }
```

- This rule affects **all** <b> under

```
<ol>
  <li>Breakfast <b>$15.00</b></li>
  <li>Lunch <b>$25.00</b></li>
  <li>Dinner <b>$50.00</b>
    <ul>
      <li>Main course <b>$30.00</b></li>
      <li>Desert <b>$20.00</b></li>
    </ul>
  </li>
</ol>
```
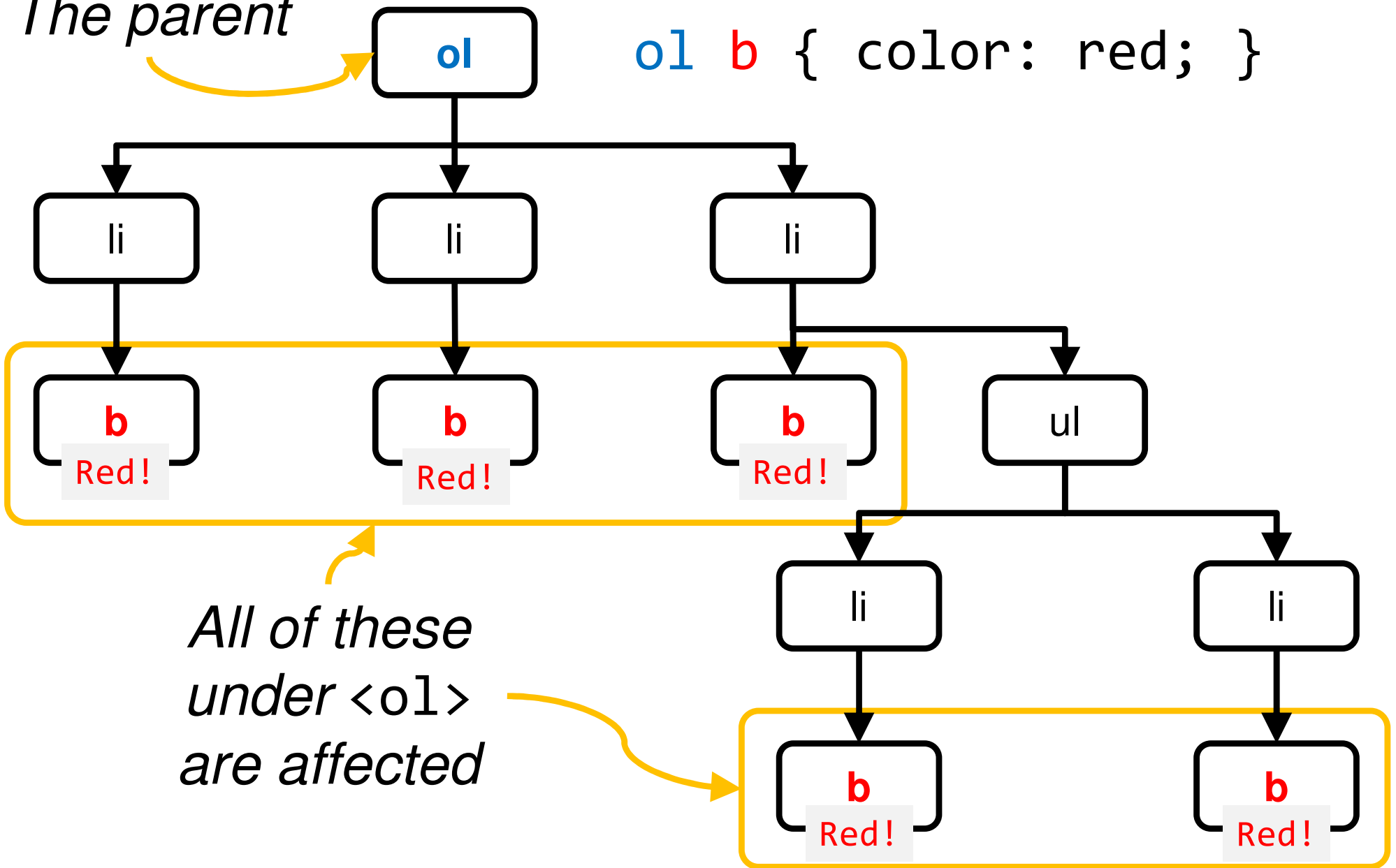
1. Breakfast **$15.00**
2. Lunch **$25.00**
3. Dinner **$50.00**
   - Main course **$30.00**
   - Desert **$20.00**

Under a Parent – Example 2

The parent

ol

`ol` `b` `{ color: red; }`

li

li

li

b
Red!

b
Red!

b
Red!

ul

All of these under <ol> are affected

li

li

b
Red!

b
Red!

Selecting Immediate Children

- Sometimes you may want only immediate children, i.e. no grand or grand grandchildren
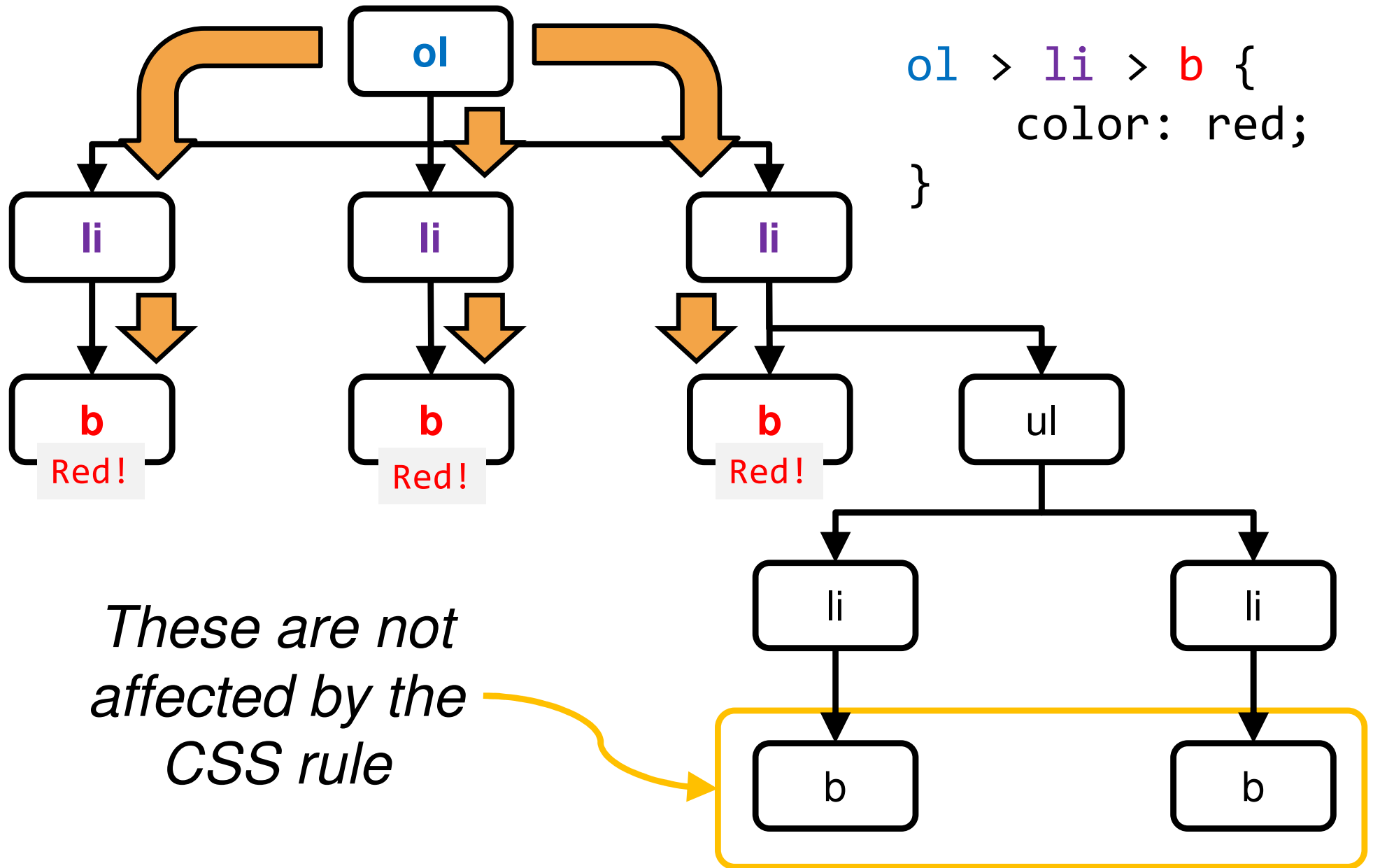- You can insert a > sign between the selectors, like this:

```
ol > li > b { color: red; }
```

*This is all <b> directly under a <li>, which is directly under a *

1. Breakfast **$15.00**
2. Lunch **$25.00**
3. Dinner **$50.00**
   - Main course **$30.00**
   - Desert **$20.00**

Immediate Children Example


```
ol > li > b {
    color: red;
}
```

Red!

Red!

Red!

These are not affected by the CSS rule

CSS Selectors in jQuery

The Meals

1. Breakfast **FREE!**
2. Lunch **FREE!**
3. Dinner **FREE!**
   - Main course **FREE!**
   - Desert **FREE!**

- You can use the CSS selectors in jQuery
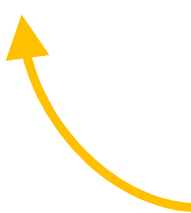
- For example, this code changes the ‹b› to FREE!

```
$("ol b").html("FREE!");
```

*Change the HTML content
of all matching ‹b›*

Using jQuery Events

- jQuery code is typically run 'inside' events
- For example, you can use the `ready` event to do something when a web page finishes loading, like this:

```
<script>
$(document).ready(function() {
    alert("The page is ready for you!");
});
</script>
```

An alert message is shown when the page is ready, i.e. the content has finished loading

The Ready Event

- The document `ready` event is very similar to the `onload` event that you have seen before

This page says

The page is ready for you!

OK

- You usually set things up for the web page inside this event in jQuery

- For example, you can set up other events inside the `ready` event
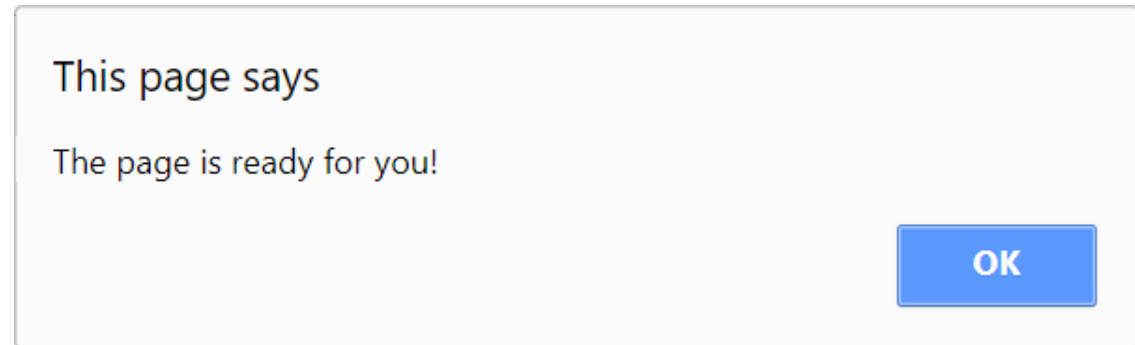
Using Other Events

- You can use events with event handling functions in jQuery easily

- You just provide the name of the event and some code to the on() function, like this:

```
$("a").on("click", function() {
    if (!confirm("Leave this page?")) {
        return false;
    }
});
```

- See next page

*Apply the event to **ALL** <a> elements*

The event is the click event

The event handling function

```
$("a").on("click", function() {
    if (!confirm("Leave this page?")) {
        return false;
    }
});
```

If the user says no, i.e. click 'Cancel', the function returns false and cancels the click event

A box asks if the user wants to leave the current page

This page says

Leave this page?

OK   Cancel

Using the Display Property

- You can use the CSS `display` property to show or hide an HTML element

- Here is an example:

```
<h1 style="display: none">
   You can't see me!
</h1>
<h1 style="display: block">
   You can see me :(
</h1>
```

Effect Functions

- You can easily write your own code to show or hide things with the CSS property

- Or, you can do that using one of these jQuery functions:

  `.show()`, `.hide()` and `.toggle()`

- All of them work with the CSS `display` property and they can also give you some nice animations

.show() and .hide()

- Here is an example to show or hide part of a web page by clicking on the `<h1>` header:

```
$("h1").on("click", function() {
    let mylist = $("ol");
    if (mylist.css("display") == "none")
        mylist.show();
    else
        mylist.hide();
});
```

Show or hide the list when the header is clicked

Showing and Hiding Things

Click on the header

The Meals

1. Breakfast **$15.00**
2. Lunch **$25.00**
3. Dinner **$50.00**
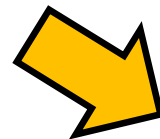   - Main course **$30.00**
   - Desert **$20.00**

The Meals

Click on the header

Using .toggle()

- You can further simplify the code by using `toggle()` only, replacing the if statement like this:

```
if (mylist.css("display") == "none")
    mylist.show();
else
    mylist.hide();
```

```
mylist.toggle();
```

- The function automatically toggles between showing and hiding of the element

Creating Animations

- jQuery can automatically use some nice animations when you show or hide elements

- Simply give a duration (in milliseconds) to `.show()`, `.hide()` or `.toggle()` , e.g.:

```
mylist.toggle(500);
```

- This code shows or hides the list using an animation that lasts for half a second

Changing the Animation

- The default animation fades / slides the content of the elements

- If you only want fading, you can use:

    `.fadeIn()`, `.fadeOut()` and `.fadeToggle()`

- If you only want sliding, you can use:

    `.slideDown()`, `.slideUp()` and `.slideToggle()`