```
<!-- Lloyd-Kendall Junos CIS 245
Please copy and paste this into a text editor.
event.StopImmediatePropagation() will not only prevent events from
executing on an element's parents when bubbling up,
but it will also prevent other event handlers to execute on the same element.
event.StopPropagation on the other hand will execute all events that are
listening to the same element, but not bubble up.
-->
<!DOCTYPE html>
<html>
<head>
<title>Assignment 2, Q2</title>
</head>
<style>

    #layer3{
        background-color: black;
        margin: auto;
        width: 300px;
        height: 350px;
        overflow: auto;
        margin-top: 50px;
    }

    #layer2{
        background-color: firebrick;
        margin: auto;
        width: 250px;
        height: 280px;
        overflow: auto;
        margin-top: 50px;
    }

    #layer1{
        background-color: wheat;
        margin: auto;
        width: 150px;
        height: 150px;
        overflow: auto;
        margin-top: 50px;
    }
    #other3{
        background-color: black;
        margin: auto;
        width: 300px;
        height: 300px;
        overflow: auto;
        margin-top: 50px;
    }
    #other2{
        background-color: firebrick;
        margin: auto;
        width: 250px;
        height: 250px;
        overflow: auto;
        margin-top: 50px;
    }

    #other1{
        background-color: wheat;
        margin: auto;
        width: 150px;
        height: 150px;
        overflow: auto;

</style>
<body>

<div id="layer3">
    <div id="layer2">
        <p>stopImmediate<br>Propagation</p>
        <div id="layer1"><p>StopImmediate<br>Propagation</p></div>
    </div>
</div>

<div id="other3">
    <div id="other2">
        <p>stopPropagation</p>
        <div id="other1"><p>stopPropagation</p></div>
    </div>
</div>

<script>
    var layer1 = document.getElementById("layer1");
    var layer2 = document.getElementById("layer2");
    var layer3 = document.getElementById("layer3");

    /* As you can see, there are 4 click events all
    * listening to the same div element. Since
    * stopImmediatePropagation is seen first before
    * the other click events, none of them execute.
    * Also notice that the event doesn't bubble
    * like in stopPropagation.
    */
    layer1.addEventListener("click", function(){
        event.stopImmediatePropagation();
    });
    layer1.addEventListener("click", function(){
        layer1.style.color = "red";
    });
    layer1.addEventListener("click", function(){
        layer1.style.backgroundColor = "blue";
    });
    layer1.addEventListener("click", function(){
        layer1.style.fontSize = "12px";
    });
    layer2.addEventListener("click", function(){
        layer2.style.backgroundColor = "green";
    })

    /* Once again, there are 4 click events all
    * listening to the same div element. However,
    * stopPropagation is used this time. This is why
    * the other events execute on the div element.
    */
    other1.addEventListener("click", function(){
        event.stopPropagation();
    });
    other1.addEventListener("click", function(){
        other1.style.color = "red";
    });
    other1.addEventListener("click", function(){
        other1.style.backgroundColor = "blue";
    });
    other1.addEventListener("click", function(){
        other1.style.fontSize = "12px";
    });
    other2.addEventListener("click", function(){
        other2.style.backgroundColor = "green";
    })

</script>
</body>
</html>
```