



jQuery



Au Mau Duong  
Technical Trainers

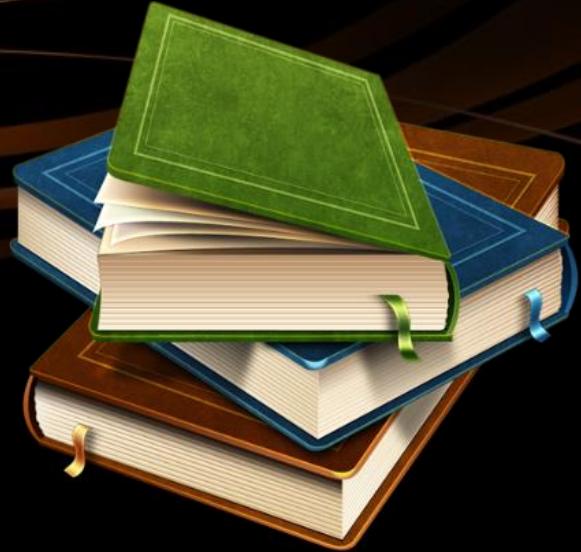
September 2013						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

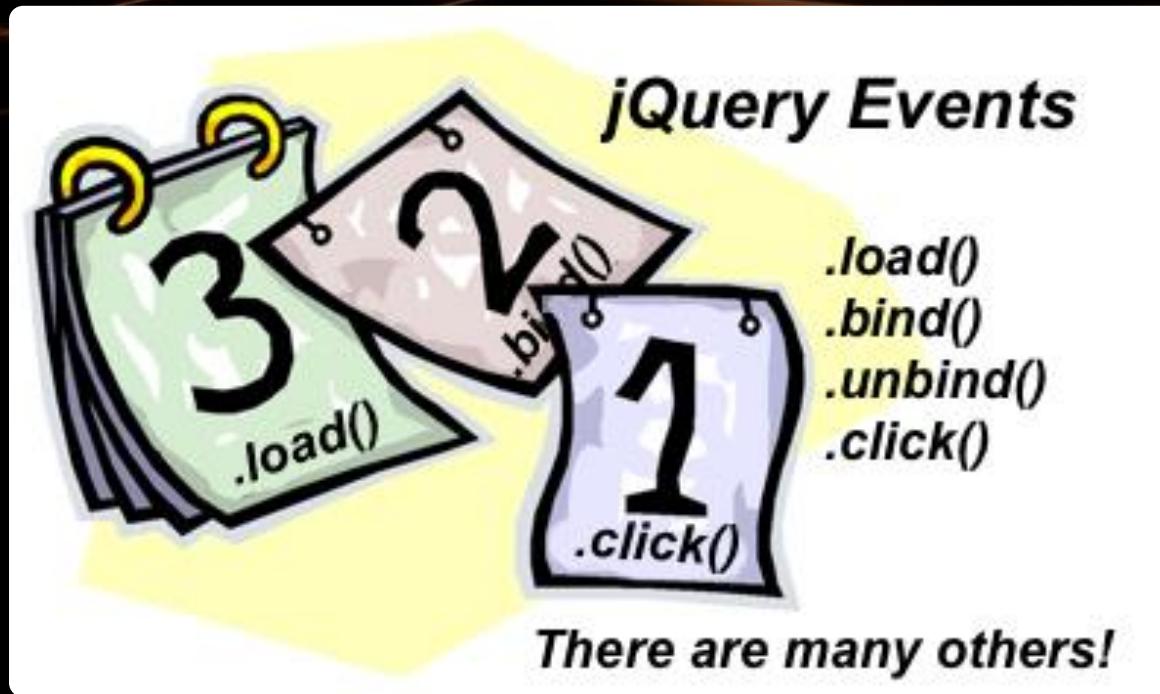
jQuery  
Event, Traversal



# Table of Contents

- jQuery Event
  - Click, dblClick
  - Mouseenter, Mouseleave, Mousedown, Mouseup
  - Hover, focus, blur
  - on
- Traversal
  - Ancestors, Descendants
  - Siblings
  - Filtering





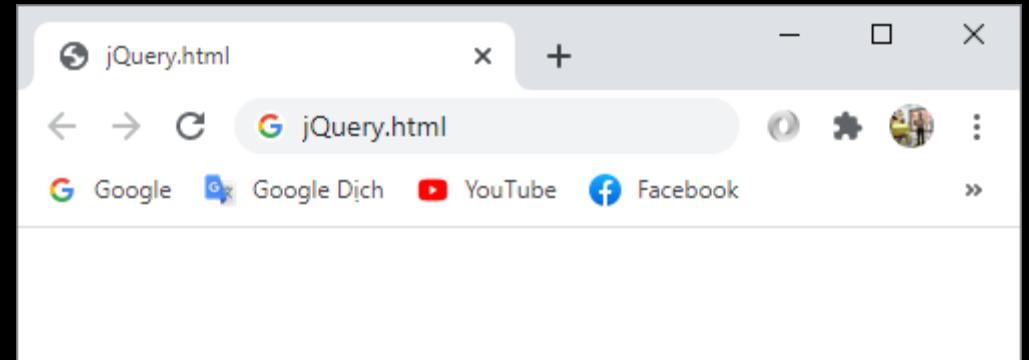
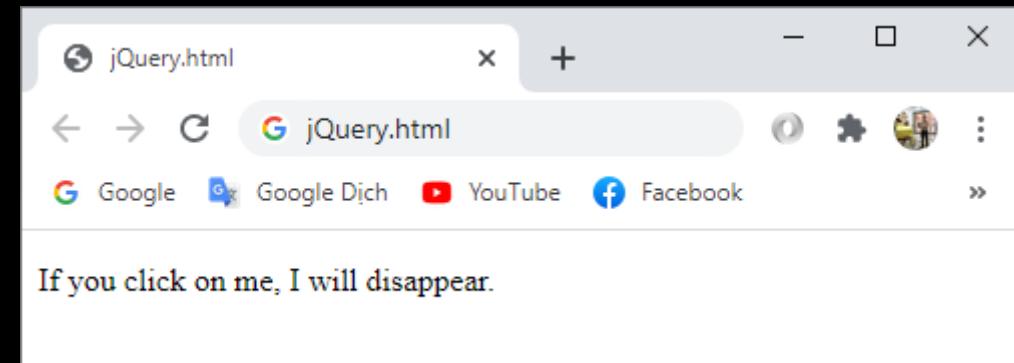
# jQuery Events

## Handling Events with Ease

# jQuery Events: Click

```
<p>If you click on me, I will disappear.</p>
```

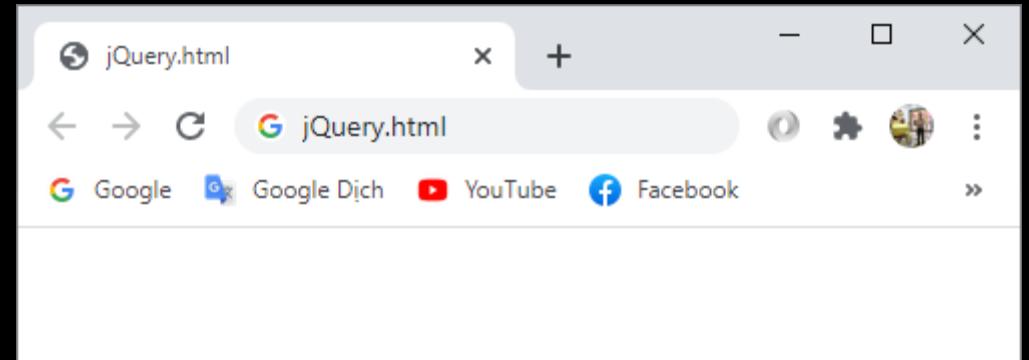
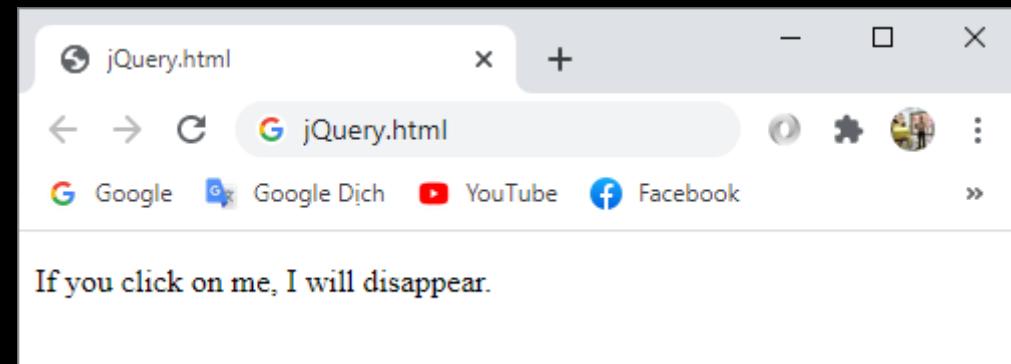
```
<script>
$(document).ready(function(){
    $("p").click(function(){
        $(this).hide();
    });
});
</script>
```



# jQuery Events: dblClick

```
<p>If you click on me, I will disappear.</p>
```

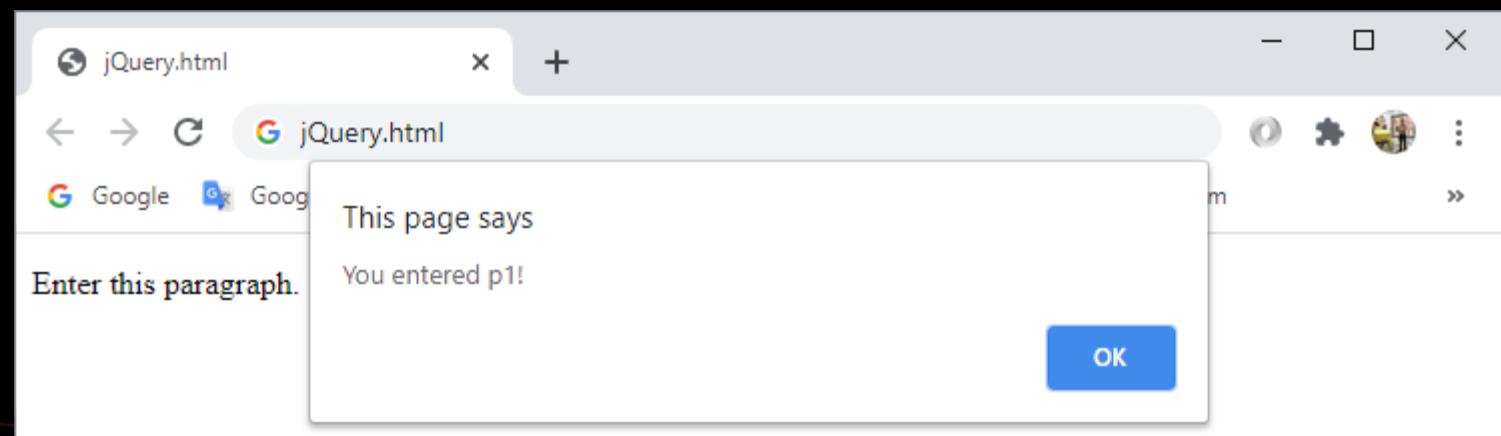
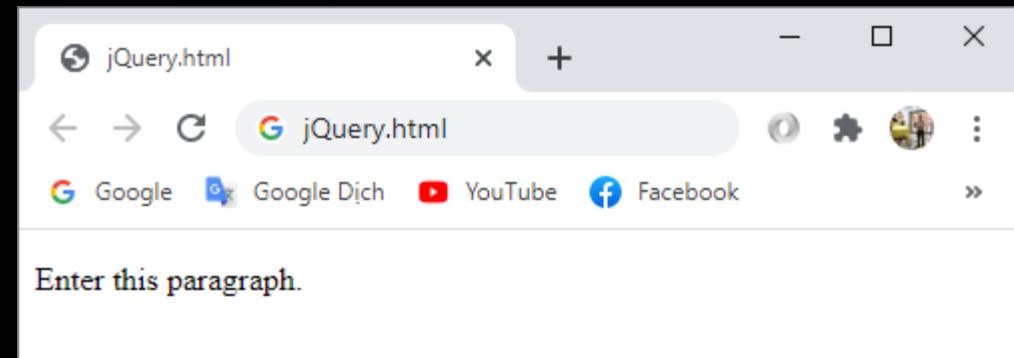
```
<script>
$(document).ready(function(){
    $("p").dblclick(function(){
        $(this).hide();
    });
});
</script>
```



# jQuery Events: mouseenter

```
<p id="p1">Enter this paragraph.</p>
```

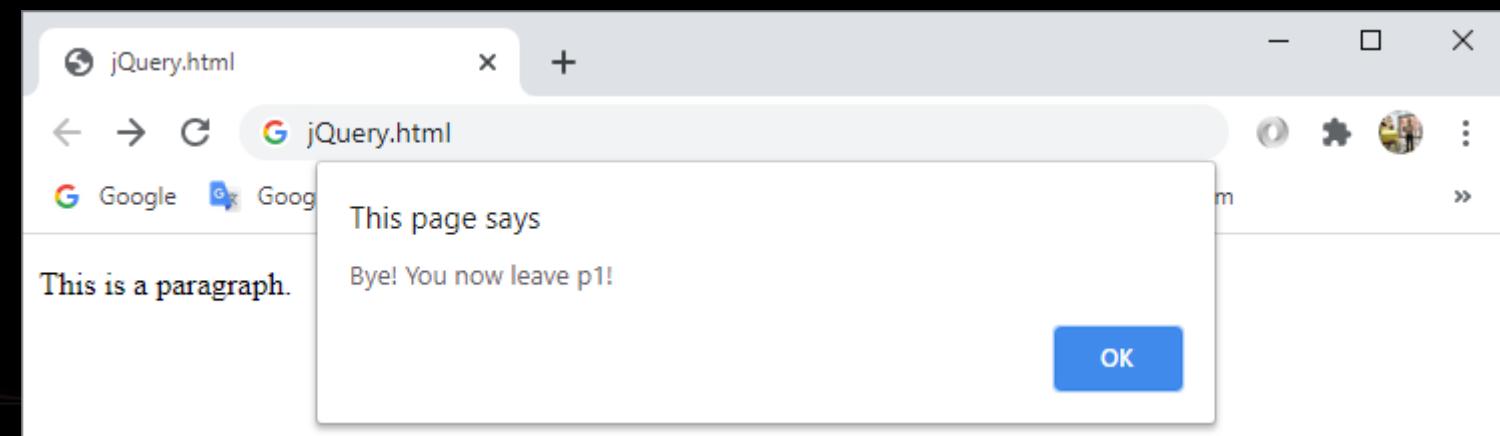
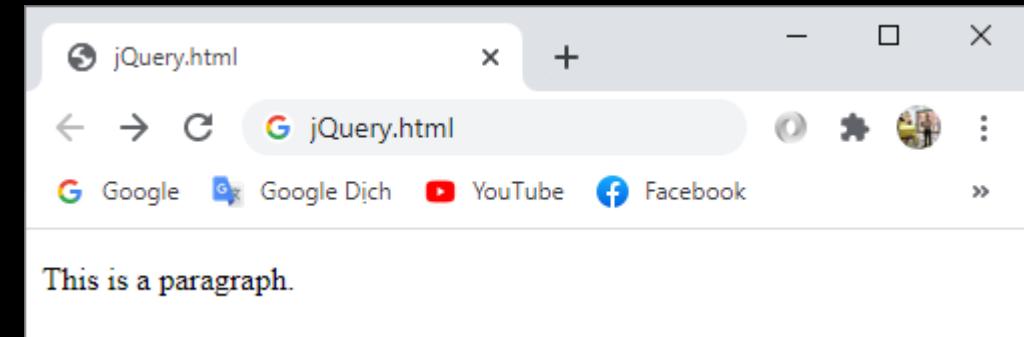
```
<script>
$(document).ready(function(){
  $("#p1").mouseenter(function(){
    alert("You entered p1!");
  });
});
</script>
```



# jQuery Events: mouseleave

```
<p id="p1">This is a paragraph.</p>
```

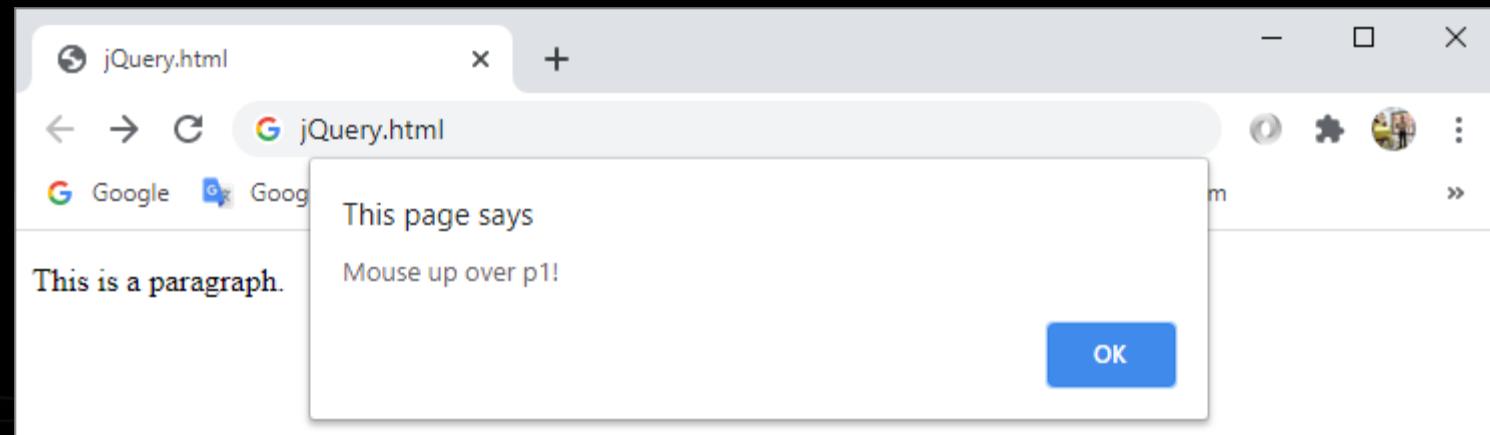
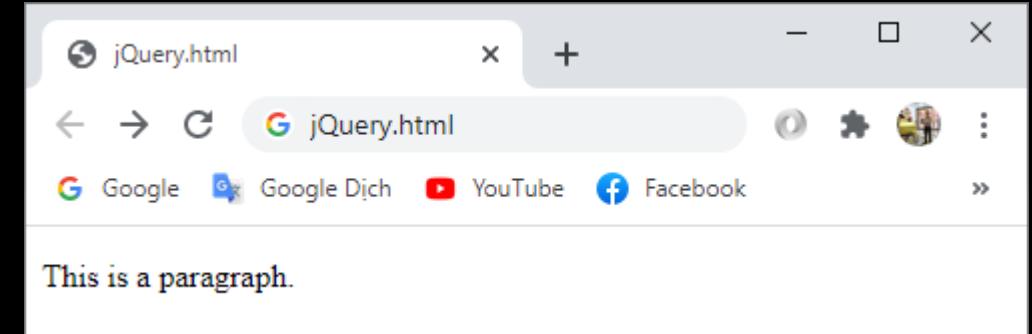
```
<script>
$(document).ready(function(){
  $("#p1").mouseleave(function(){
    alert("Bye! You now leave p1!");
  });
});
</script>
```



# jQuery Events: mousedown

```
<p id="p1">This is a paragraph.</p>
```

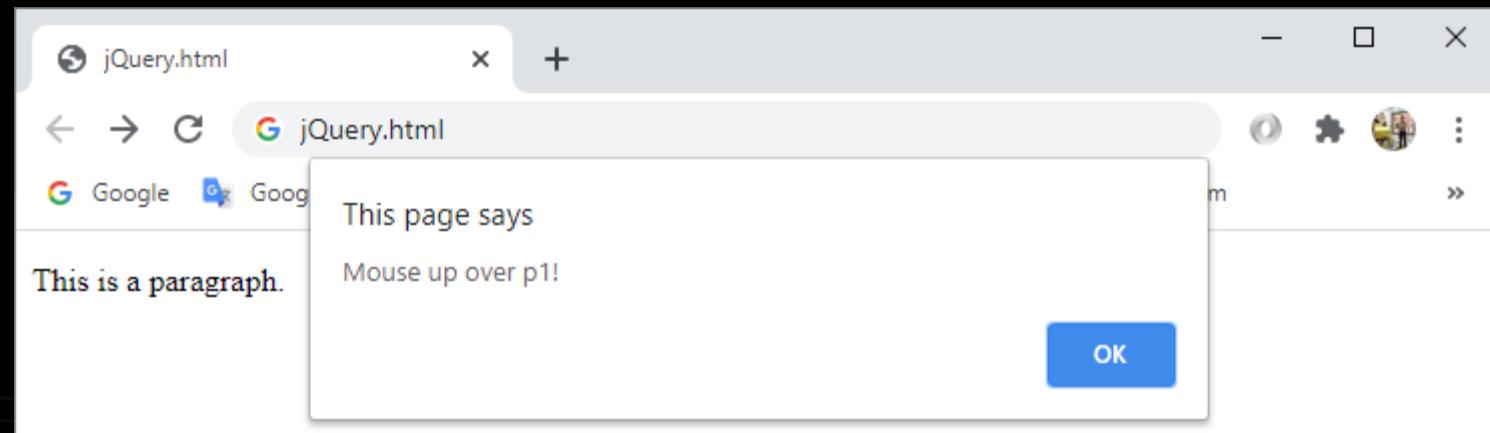
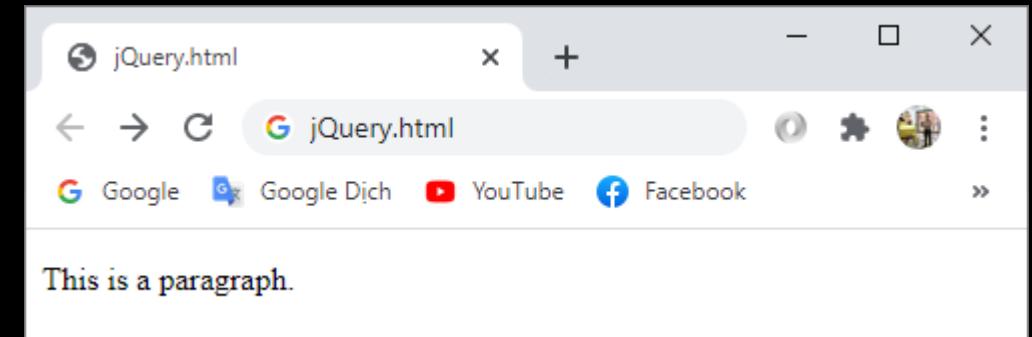
```
<script>
$(document).ready(function(){
    $("#p1").mousedown(function(){
        alert("Mouse down over p1!");
    });
});
</script>
```



# jQuery Events: mouseup

```
<p id="p1">This is a paragraph.</p>
```

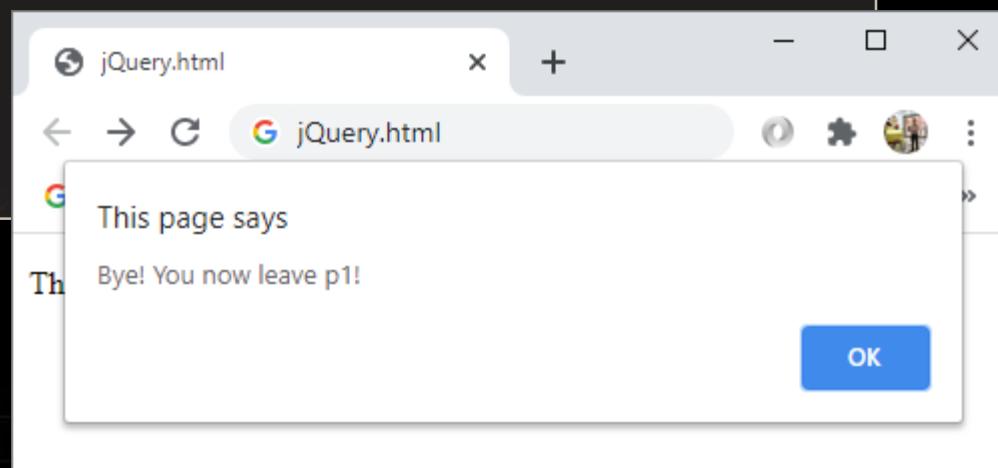
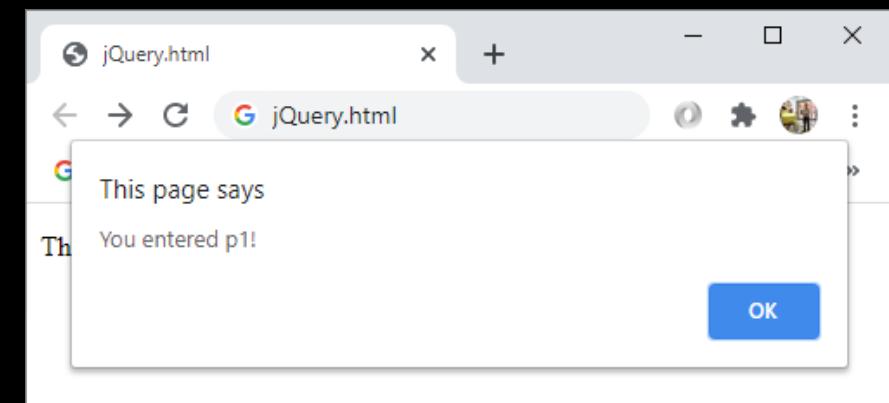
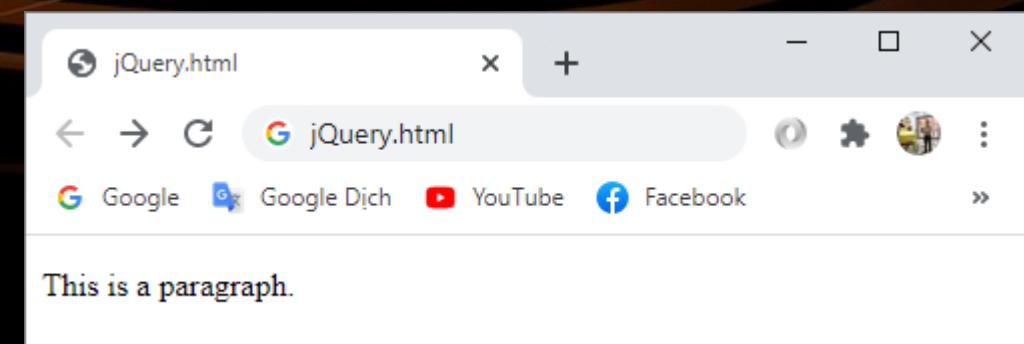
```
<script>
$(document).ready(function(){
  $("#p1").mouseup(function(){
    alert("Mouse up over p1!");
  });
});
</script>
```



# jQuery Events: hover

```
<p id="p1">This is a paragraph.</p>
```

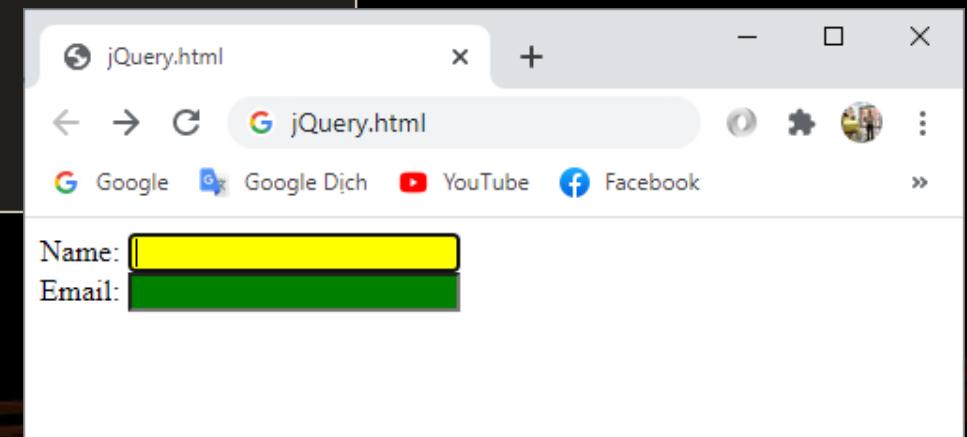
```
<script>
$(document).ready(function(){
    $("#p1").hover(function(){
        alert("You entered p1!");
    },
    function(){
        alert("Bye! You now leave p1!");
    });
});
</script>
```



# jQuery Events: focus and blur

```
Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">
```

```
<script>
$(document).ready(function(){
    $("input").focus(function(){
        $(this).css("background-color", "yellow");
    });
    $("input").blur(function(){
        $(this).css("background-color", "green");
    });
});
</script>
```



# jQuery Events: Attach / Remove

- Attaching events on certain elements

```
$('a.button').on('click', buttonClicked);  
function buttonClicked() {  
    $('.selected').removeClass('selected');  
    $(this).addClass('selected');  
    // "this" is the event source (the hyperlink clicked)  
}
```

- Removing event handler from certain elements

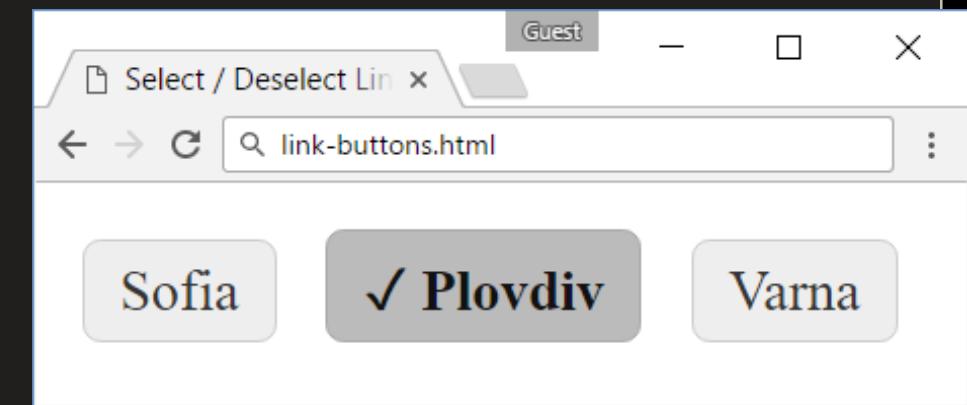
```
$('a.button').off('click', buttonClicked);
```

# Problem: Link Buttons – HTML

link-buttons.html

```
<head>
  <link rel="stylesheet" href="link-buttons.css" />
  <script src="jquery-3.1.1.min.js"></script>
  <script src="link-buttons.js"></script>
</head>

<body onload="attachEvents()">
  <a class="button">Sofia</a>
  <a class="button">Plovdiv</a>
  <a class="button">Varna</a>
</body>
```



# Problem: Link Buttons – CSS

## link-buttons.css

```
a.button {  
    border: 1px solid #CCC;  
    background: #EEE;  
    padding: 5px 10px;  
    border-radius: 5px;  
    color: #333;  
    text-decoration: none;  
    display: inline-block;  
    margin: 5px;  
}
```

```
a.button.selected {  
    color: #111;  
    font-weight: bold;  
    border: 1px solid #AAA;  
    background: #BBB;  
}  
a.button.selected::before {  
    content: "\2713\20\20";  
}  
a:hover {cursor: pointer;}
```

# Solution: Link Buttons – JavaScript

## link-buttons.js

```
function attachEvents() {  
    $('a.button').on('click', buttonClicked);  
  
    function buttonClicked() {  
        $('.selected').removeClass('selected');  
        $(this).addClass('selected');  
        // "this" is the event source (the hyperlink clicked)  
    }  
}
```

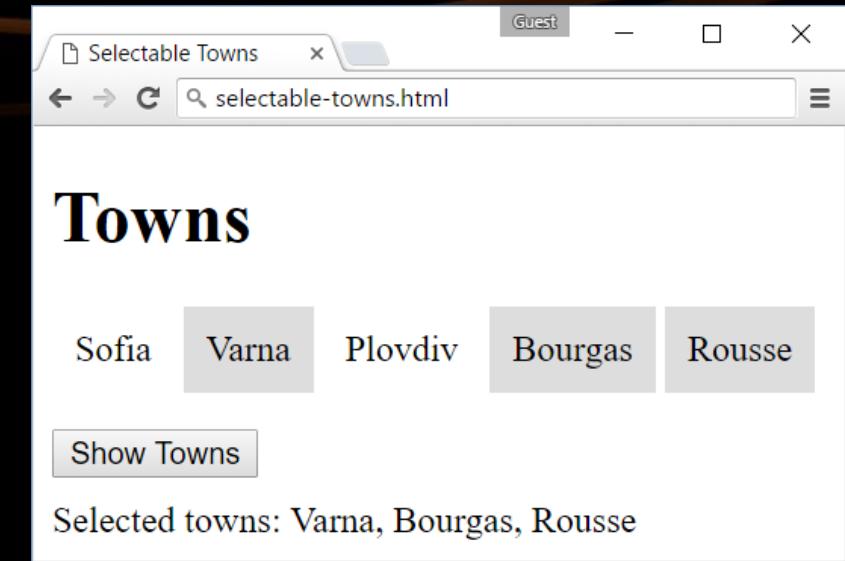
# jQuery Events: Multiple

- Handling events on multiple elements
  - Add a handler on the parent element

```
function onListItemClick() {  
    $('.selected').removeClass('selected');  
    $(this).addClass('selected');  
}  
  
$('ul').on('click', 'li', onListItemClick);
```

# Problem: Selectable Towns

- An HTML page listing towns
  - Clicking on a town should select / deselect it
  - A button shows all selected towns



```
<style>li { display: inline-block }</style>
<ul id="items"><li>Sofia</li><li>Varna</li>...</ul>
<button id="showTownsButton">Show Towns</button>
<div id="selectedTowns"></div>
<script>$(()=>attachEvents())</script>
```

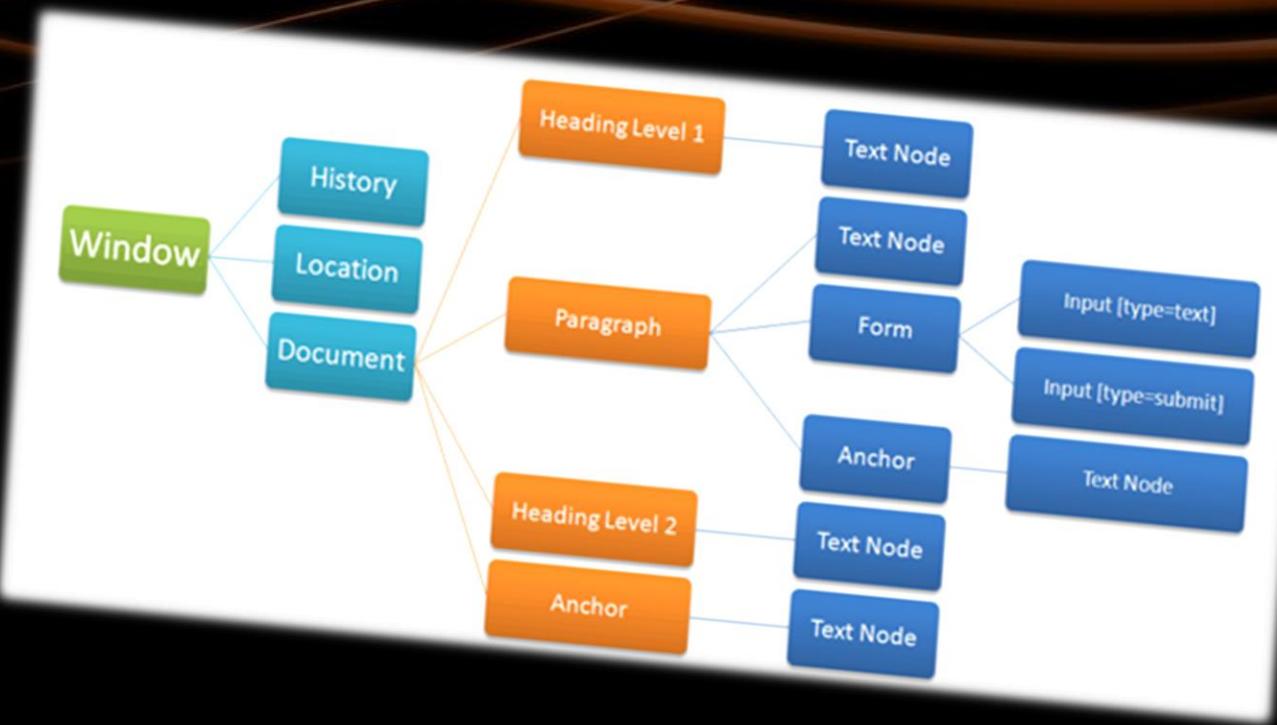
# Solution: Selectable Towns – Click Towns

```
function attachEvents() {  
  $('#items').on('click', 'li', function() {  
    let li = $(this);  
    if (li.attr('data-selected')) {  
      li.removeAttr('data-selected');  
      li.css('background', '');  
    } else {  
      li.attr('data-selected', 'true');  
      li.css('background', '#DDD');  
    }  
  });  
}
```

Attach attribute  
`'data-selected' = 'true'`  
to each selected `<li>`

# Solution: Selectable Towns – Show Towns

```
$('#showTownsButton').on('click', function() {
  let selLi = $('#items li[data-selected=true]');
  let towns = selLi.toArray()
    .map(li => li.textContent).join(', ');
  $('#selectedTowns')
    .text("Selected towns: " + towns);
});
```

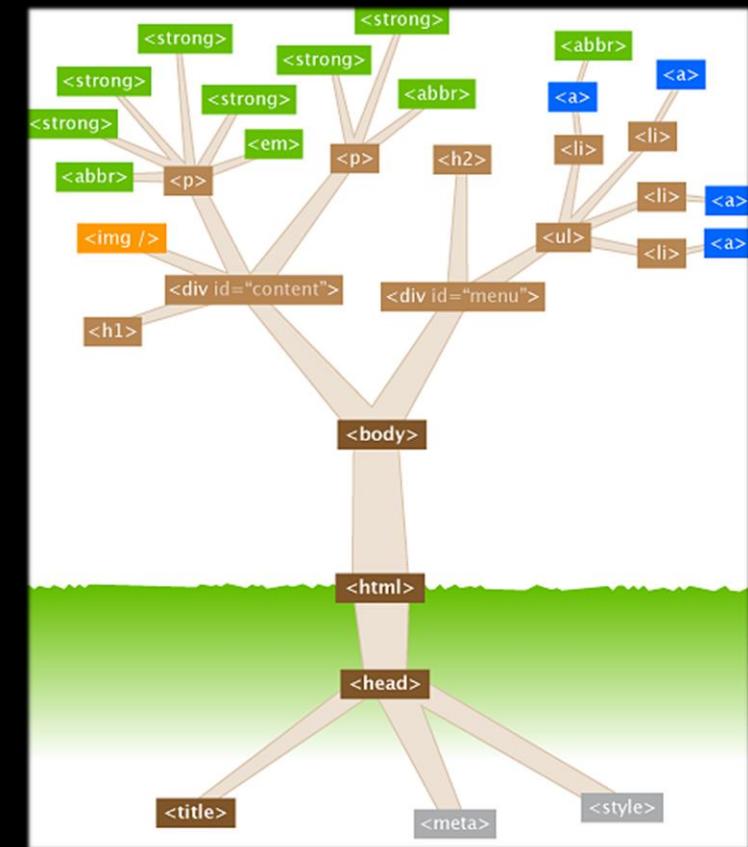


# DOM Traversal

Traversing the Nodes of the DOM

# DOM Traversal

- The DOM can be traversed with jQuery, as with plain JavaScript
  - jQuery has properties for:
    - Next and previous siblings
    - Parents and children
    - First and last in a collection
  - It is impossible to reference elements that are not yet rendered at the time of script execution!



# jQuery Traversing – Ancestors

- With jQuery you can traverse up the DOM tree to find ancestors of an element
- An ancestor is a parent, grandparent, great-grandparent, and so on
- Three useful jQuery methods for traversing up the DOM tree are:
  - `parent()`
  - `parents()`
  - `parentsUntil()`

# jQuery Traversing – Ancestors - parent() (1)

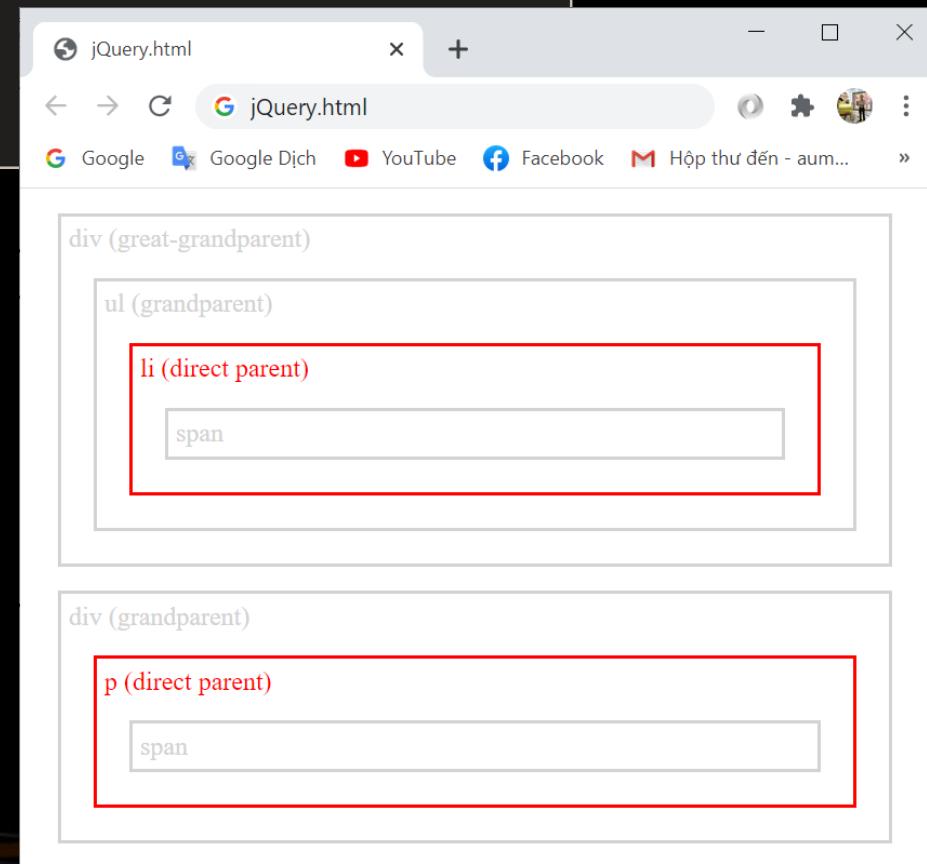
```
<style>
.ancestors * {
    display: block;
    border: 2px solid
lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
```

```
<div class="ancestors">
    <div style="width:500px;">div (great-grandparent)
        <ul>ul (grandparent)
            <li>li (direct parent)
                <span>span</span>
            </li>
        </ul>
    </div>

    <div style="width:500px;">div (grandparent)
        <p>p (direct parent)
            <span>span</span>
        </p>
    </div>
</div>
```

# jQuery Traversing – Ancestors - parent() (2)

```
<script>
$(document).ready(function(){
    $("span").parent().css({"color": "red", "border": "2px solid red"});
});
</script>
```



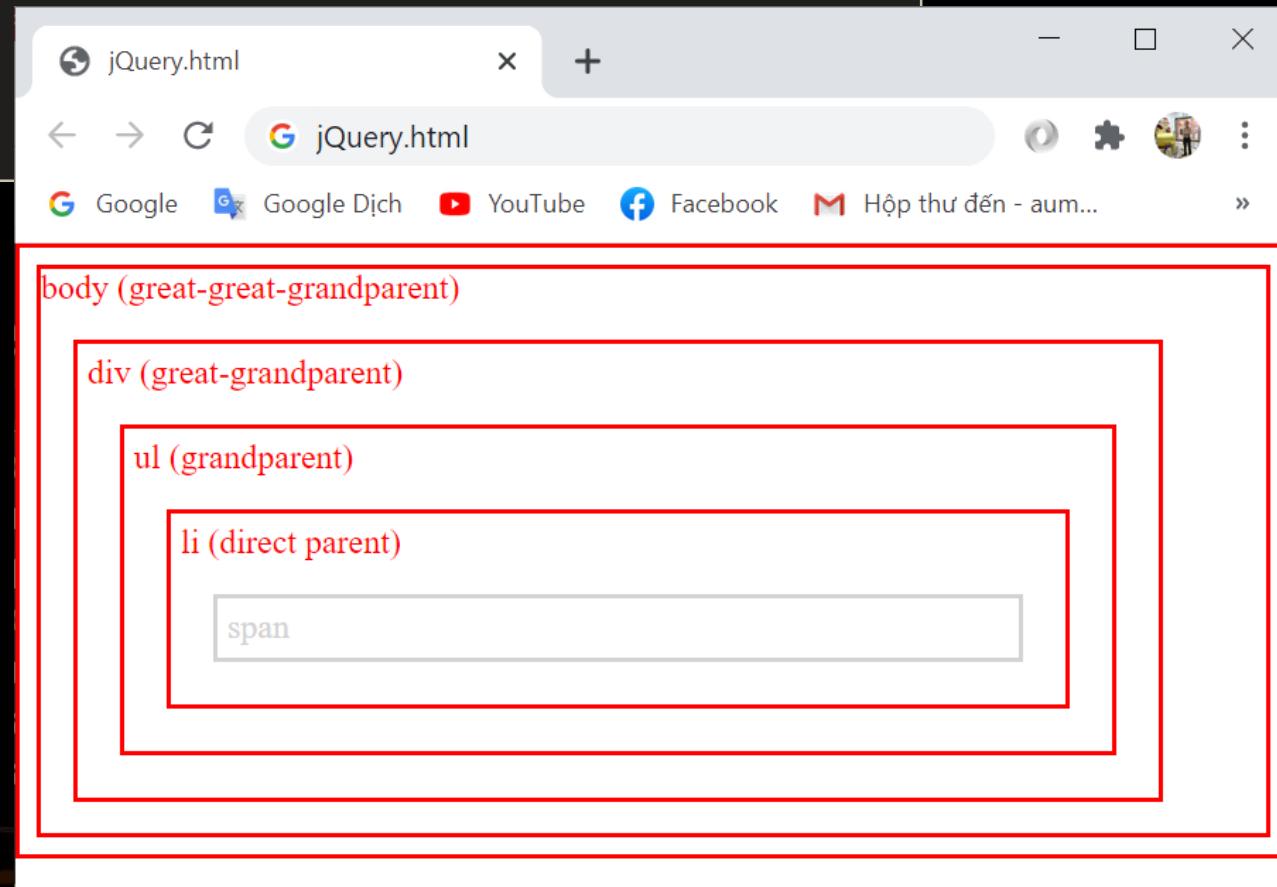
# jQuery Traversing – Ancestors - parents() (1)

```
<style>
.ancestors * {
    display: block;
    border: 2px solid
lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
```

```
<body class="ancestors">body (great-great-grandparent)
<div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
        <li>li (direct parent)
            <span>span</span>
        </li>
    </ul>
</div>
</body>
```

# jQuery Traversing – Ancestors - parents() (2)

```
<script>
$(document).ready(function(){
    $("span").parents().css({"color": "red", "border": "2px solid red"});
});
</script>
```



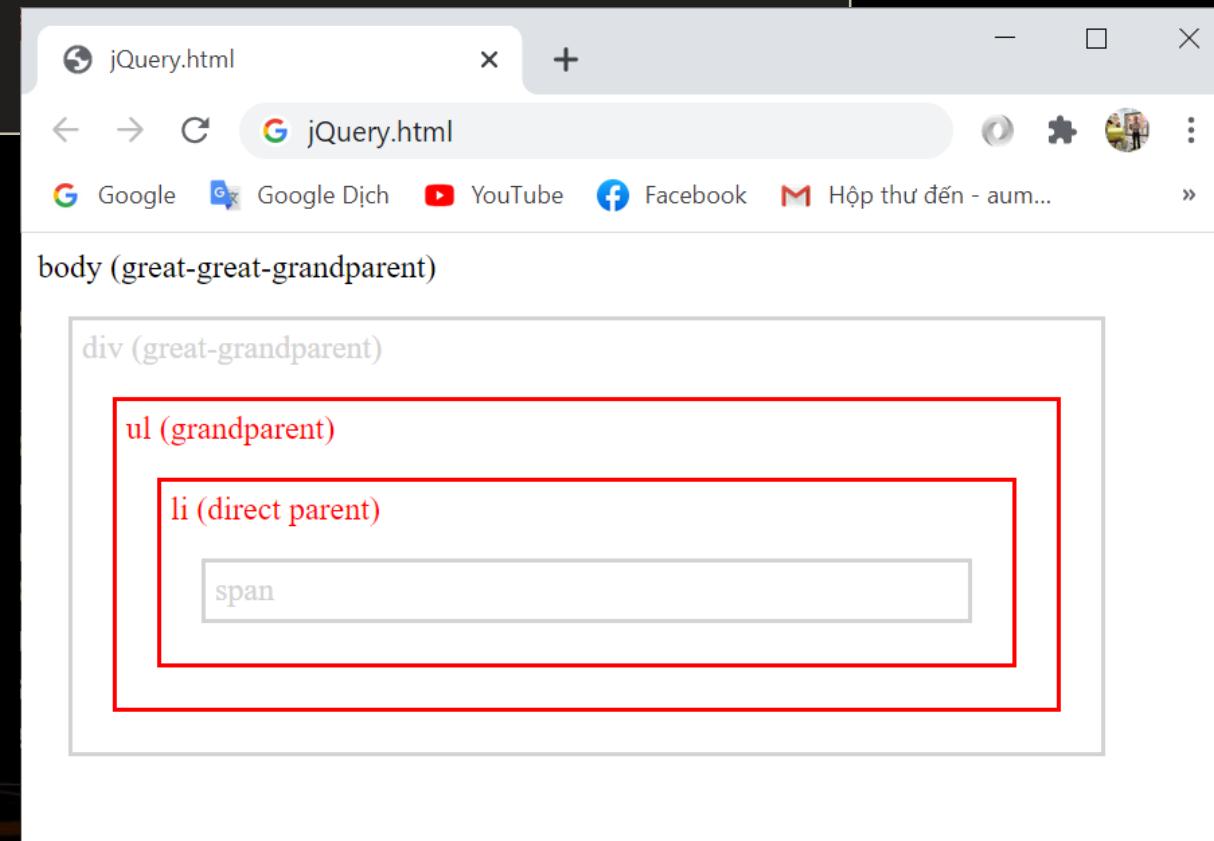
# jQuery Traversing – Ancestors - parentsUntil() (1)

```
<style>
.ancestors * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
```

```
<body class="ancestors"> body (great-great-grandparent)
    <div style="width:500px;">div (great-grandparent)
        <ul>ul (grandparent)
            <li>li (direct parent)
                <span>span</span>
            </li>
        </ul>
    </div>
</body>
```

# jQuery Traversing – Ancestors - parentsUntil() (2)

```
<script>
$(document).ready(function(){
    $("span").parentsUntil("div").css({"color": "red",
"border": "2px solid red"});
});
</script>
```



# jQuery Traversing – Descendants

- With jQuery you can traverse down the DOM tree to find descendants of an element
- A descendant is a child, grandchild, great-grandchild, and so on
- Two useful jQuery methods for traversing down the DOM tree are:
  - `children()`
  - `find()`

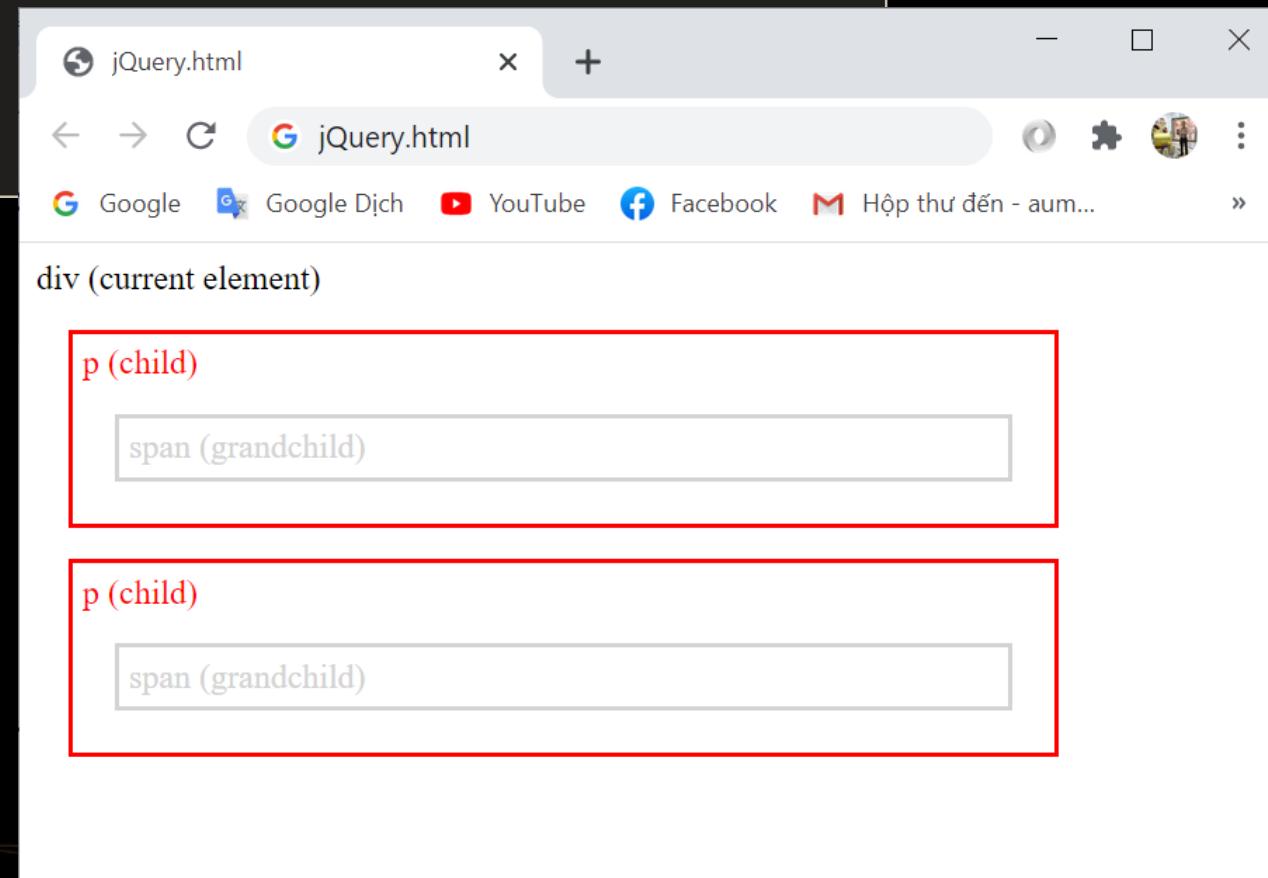
# jQuery Traversing – Descendants - children() (1)

```
<style>
.ancestors * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
```

```
<div class="descendants" style="width:500px;">div
(current element)
<p>p (child)
    <span>span (grandchild)</span>
</p>
<p>p (child)
    <span>span (grandchild)</span>
</p>
</div>
```

# jQuery Traversing – Ancestors - children() (2)

```
<script>
$(document).ready(function(){
    $("div").children().css({"color": "red", "border": "2px solid red"});
});
</script>
```



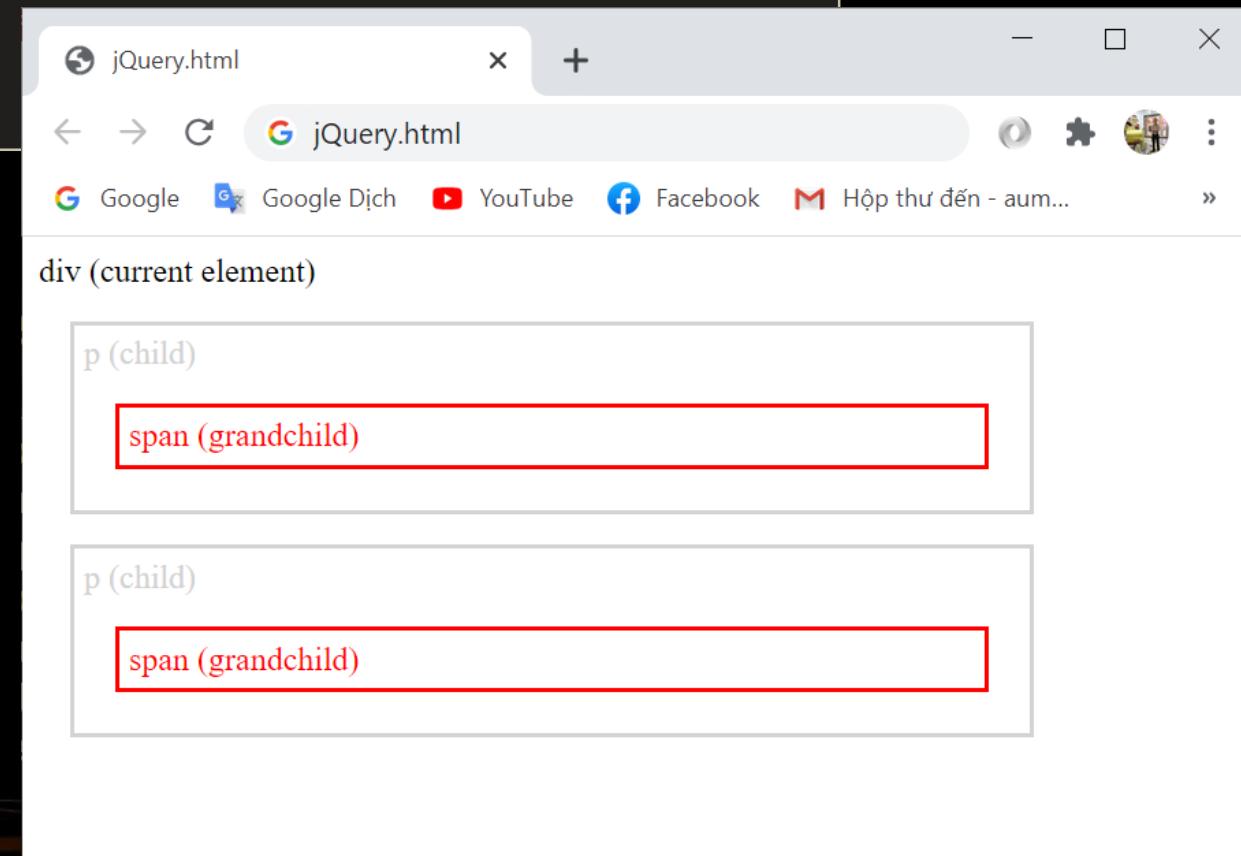
# jQuery Traversing – Descendants - find() (1)

```
<style>
.ancestors * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
```

```
<div class="descendants" style="width:500px;">div
(current element)
<p>p (child)
    <span>span (grandchild)</span>
</p>
<p>p (child)
    <span>span (grandchild)</span>
</p>
</div>
```

# jQuery Traversing – Ancestors - find() (2)

```
<script>
$(document).ready(function(){
    $("div").find("span").css({"color": "red",
"border": "2px solid red"}));
});
</script>
```



# jQuery Traversing – Siblings

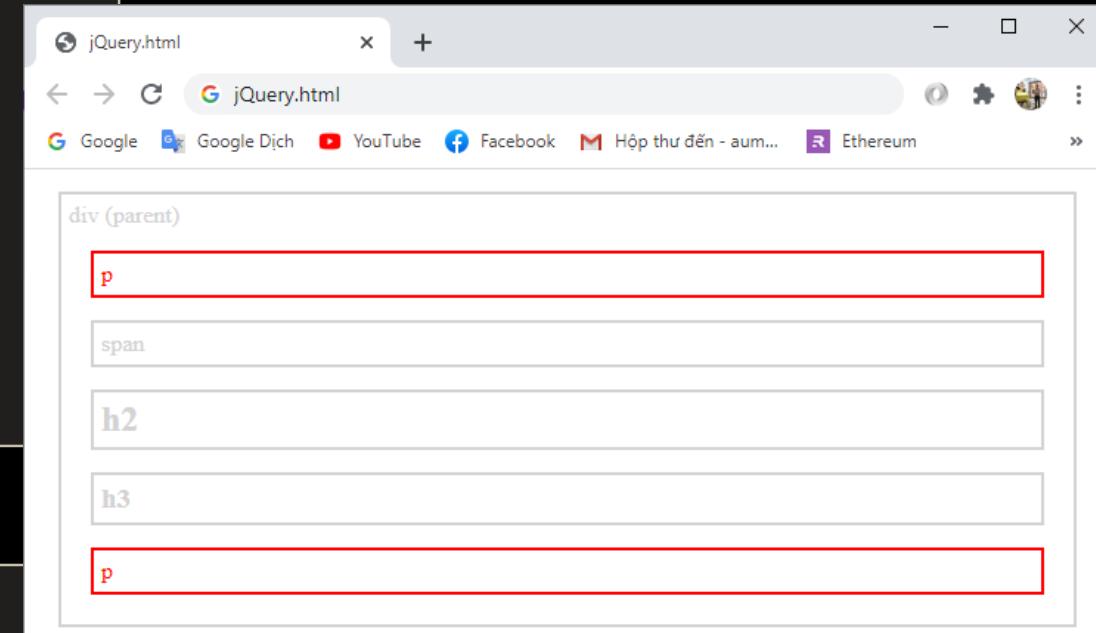
- With jQuery you can traverse sideways in the DOM tree to find siblings of an element
- Siblings share the same parent
- There are many useful jQuery methods for traversing sideways in the DOM tree:
  - `siblings()`
  - `next()`
  - `nextAll()`
  - `nextUntil()`
  - `prev()`
  - `prevAll()`
  - `prevUntil()`

# jQuery Traversing – Siblings - siblings()

```
<style>
  .siblings * {
    display: block;
    border: 2px solid
    lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
  }
</style>
```

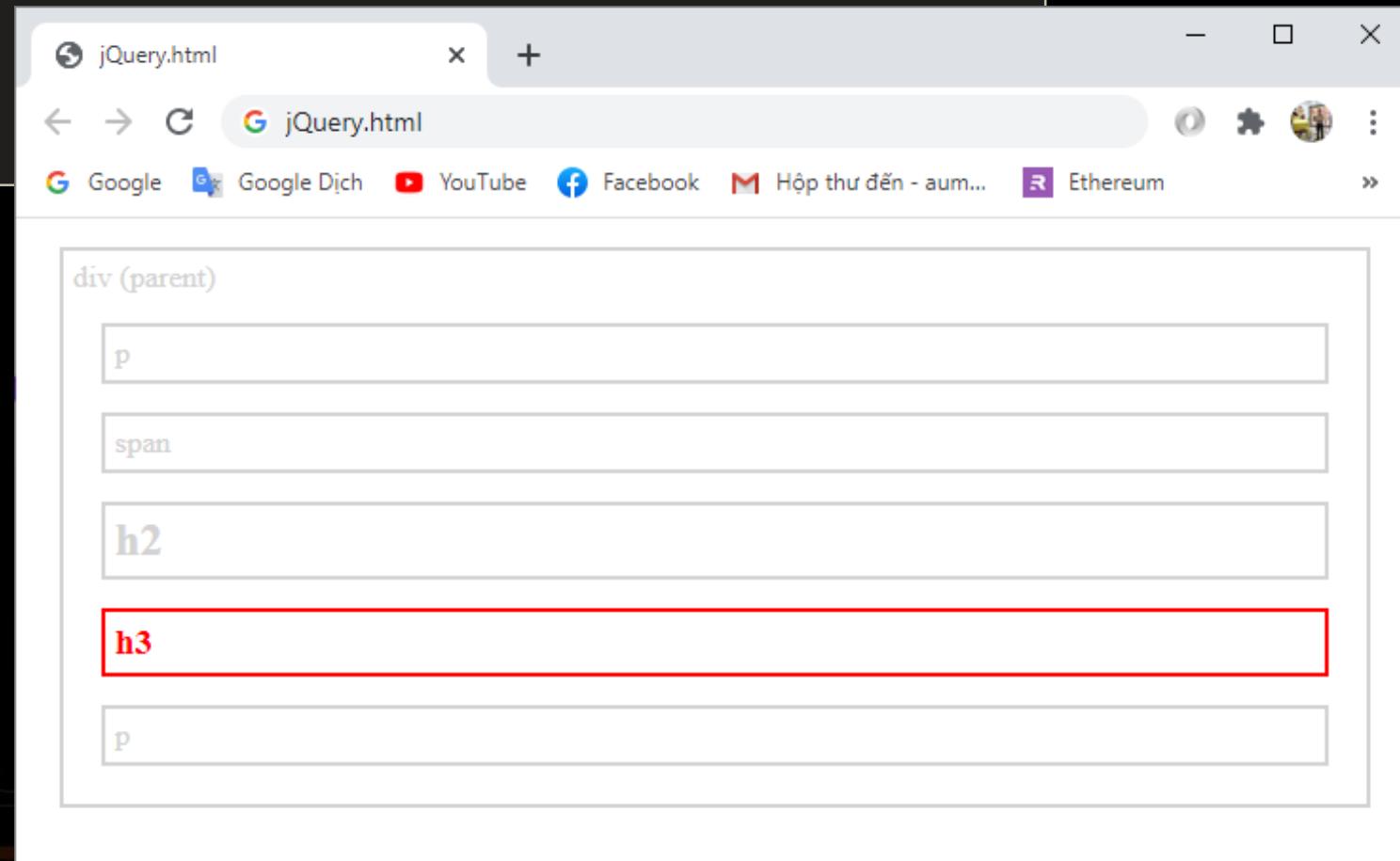
```
<body class="siblings">
<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>
</body>
```

```
<script>
$(document).ready(function(){
  $("h2").siblings("p").css({"color": "red",
  "border": "2px solid red"});
})
</script>
```



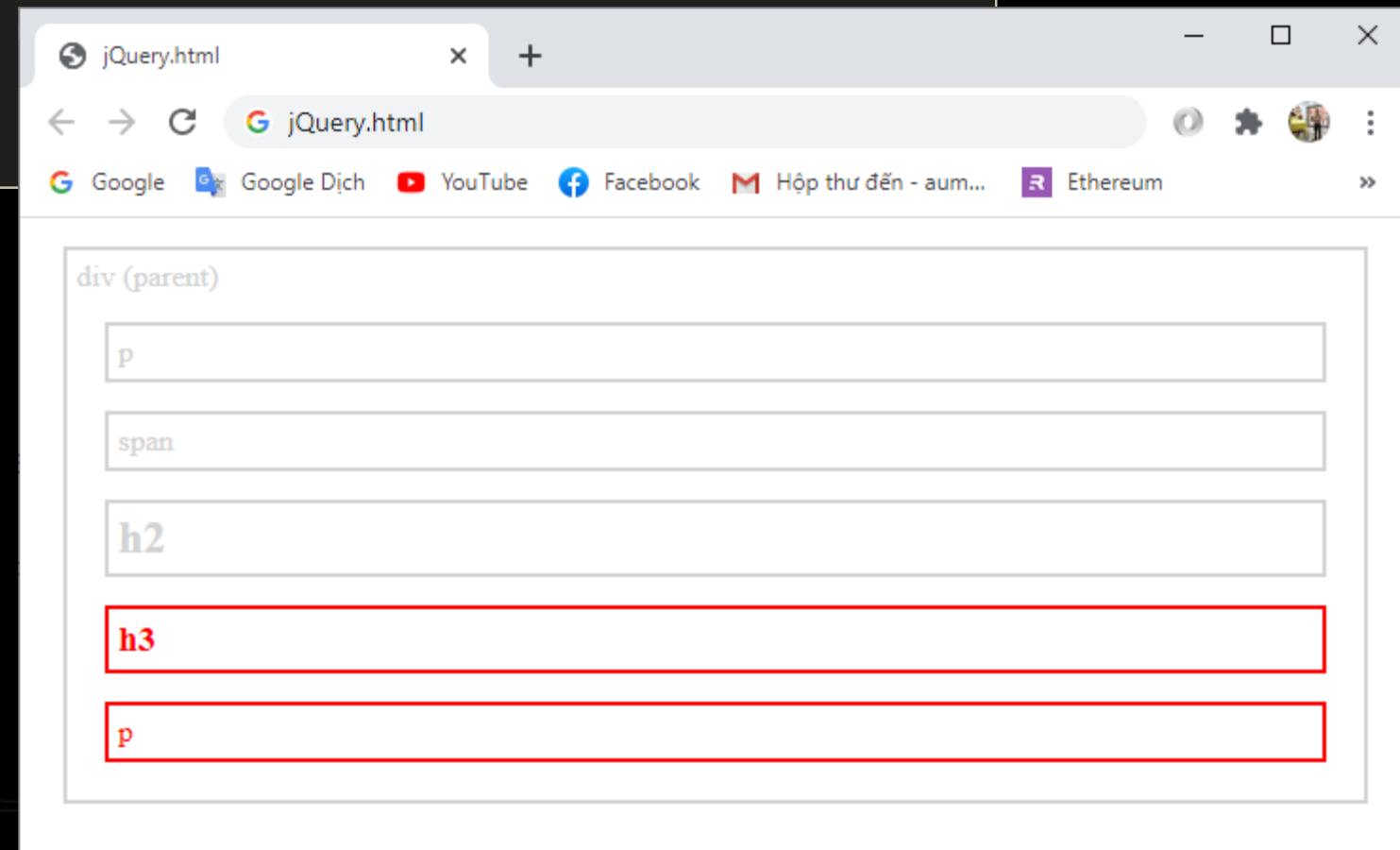
# jQuery Traversing – Siblings - next()

```
<script>
$(document).ready(function(){
    $("h2").next().css({"color": "red", "border": "2px
solid red"});
});
</script>
```



# jQuery Traversing – Siblings - nextAll()

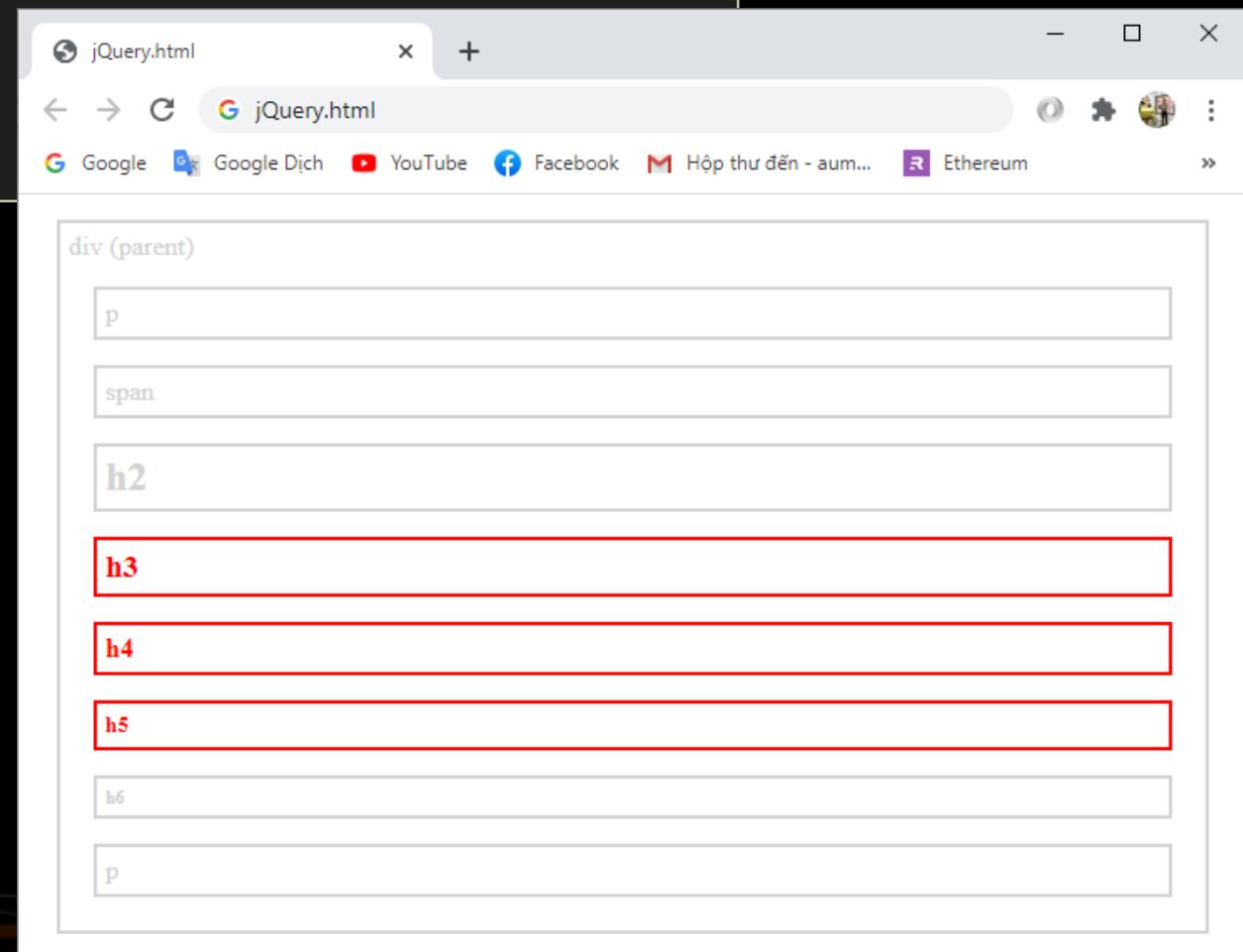
```
<script>
$(document).ready(function(){
    $("h2").nextAll().css({"color": "red", "border": "2px solid red"});
});
</script>
```



# jQuery Traversing – Siblings - nextUntil()

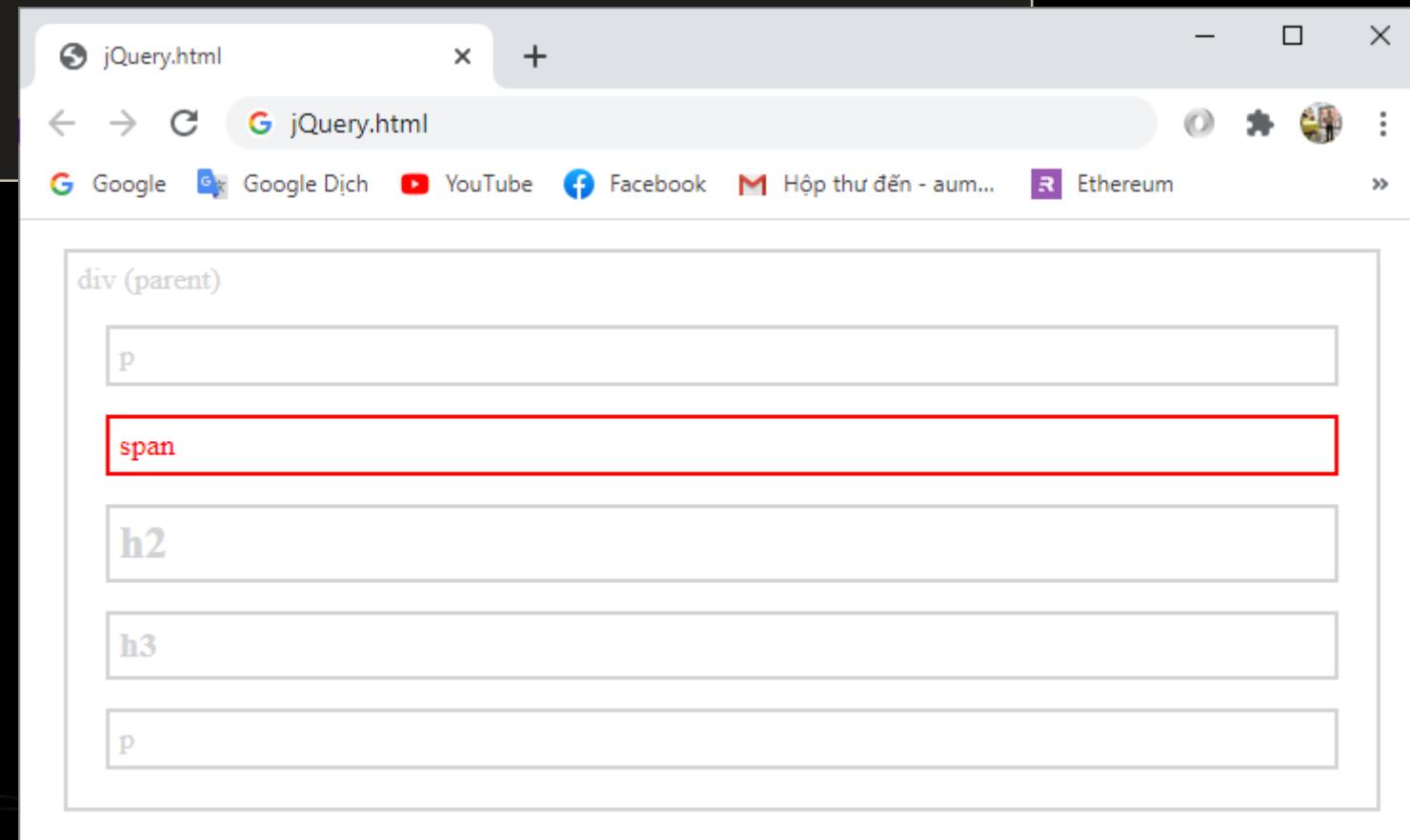
```
<script>
$(document).ready(function(){
    $("h2").nextUntil("h6").css({"color": "red",
"border": "2px solid red"});
});
</script>
```

```
<div>div (parent)
<p>p</p>
<span>span</span>
<h2>h2</h2>
<h3>h3</h3>
<h4>h4</h4>
<h5>h5</h5>
<h6>h6</h6>
<p>p</p>
</div>
```



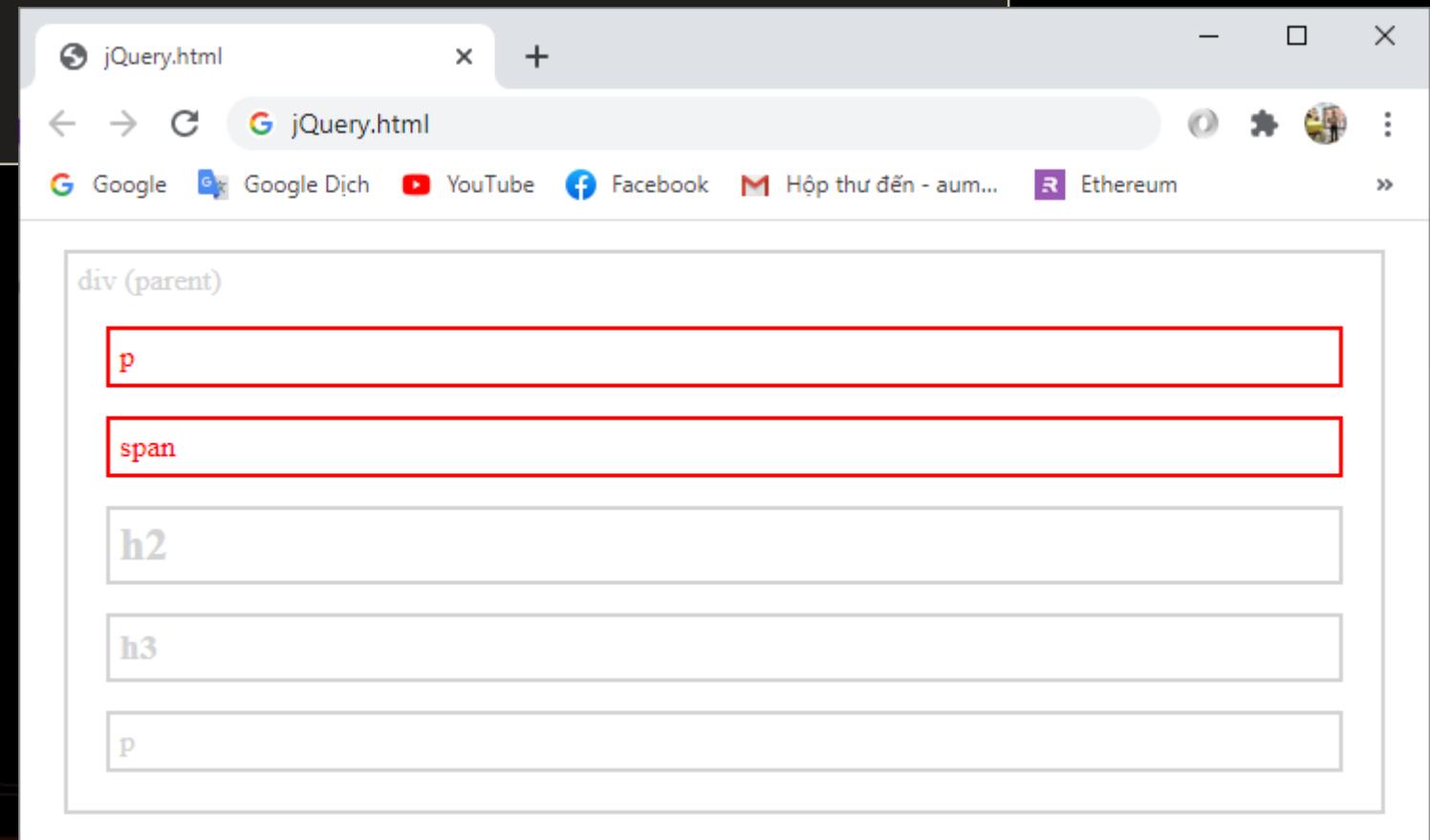
# jQuery Traversing – Siblings - prev()

```
<script>
$(document).ready(function(){
    $("h2").prev().css({"color": "red", "border": "2px
solid red"});
});
</script>
```



# jQuery Traversing – Siblings - prevAll()

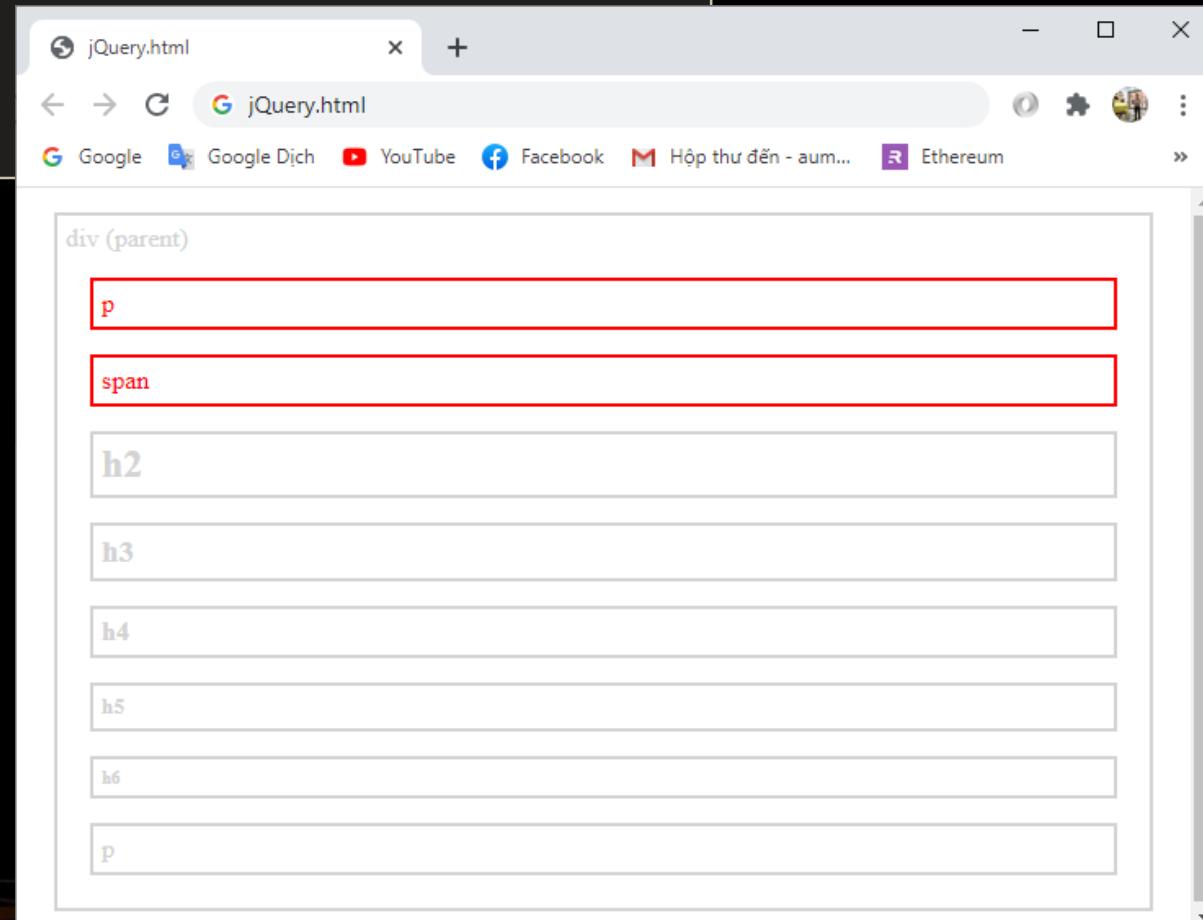
```
<script>
$(document).ready(function(){
    $("h2").prevAll().css({"color": "red", "border": "2px solid red"});
});
</script>
```



# jQuery Traversing – Siblings - prevUntil()

```
<script>
$(document).ready(function(){
    $("h2").prevUntil("h6").css({"color": "red",
"border": "2px solid red"});
});
</script>
```

```
<div>div (parent)
<p>p</p>
<span>span</span>
<h2>h2</h2>
<h3>h3</h3>
<h4>h4</h4>
<h5>h5</h5>
<h6>h6</h6>
<p>p</p>
</div>
```

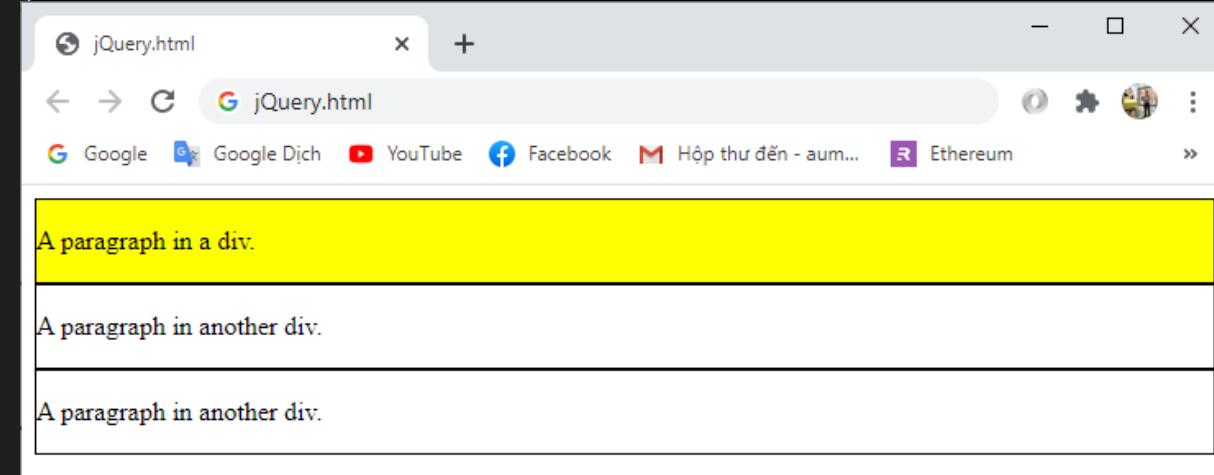


# jQuery Traversing – Filtering

- The most basic filtering methods are `first()`, `last()` and `eq()`, which allow you to select a specific element based on its position in a group of elements
- Other filtering methods, like `filter()` and `not()` allow you to select elements that match, or do not match, a certain criteria

# jQuery Traversing – Filtering - first()

```
<div style="border: 1px solid black;">
  <p>A paragraph in a div.</p>
</div>
<div style="border: 1px solid black;">
  <p>A paragraph in another div.</p>
</div>
<div style="border: 1px solid black;">
  <p>A paragraph in another div.</p>
</div>
```

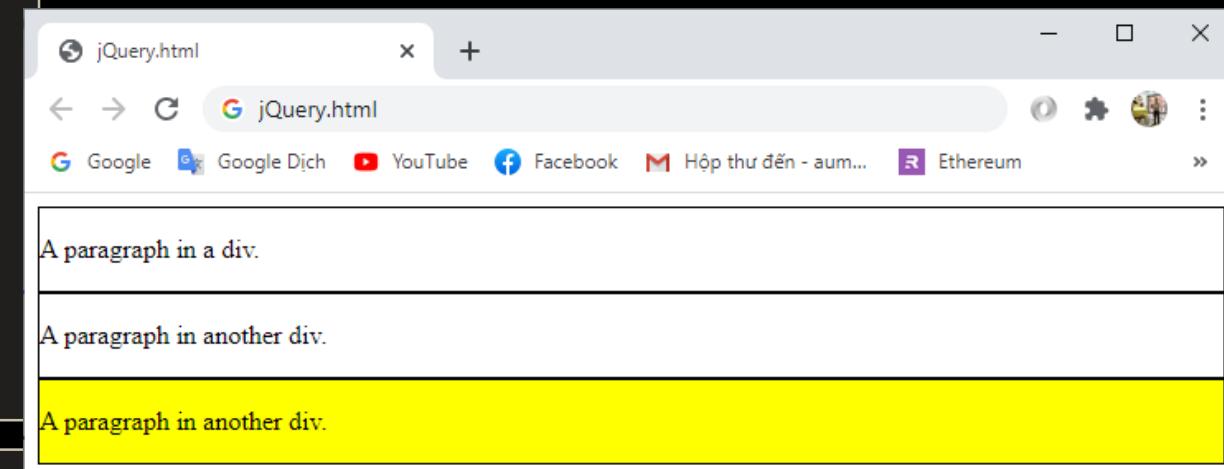


```
<script>
$(document).ready(function(){
  $("div").first().css("background-color",
"yellow");
});
</script>
```

# jQuery Traversing – Filtering - last()

```
<div style="border: 1px solid black;">
  <p>A paragraph in a div.</p>
</div>
<div style="border: 1px solid black;">
  <p>A paragraph in another div.</p>
</div>
<div style="border: 1px solid black;">
  <p>A paragraph in another div.</p>
</div>
```

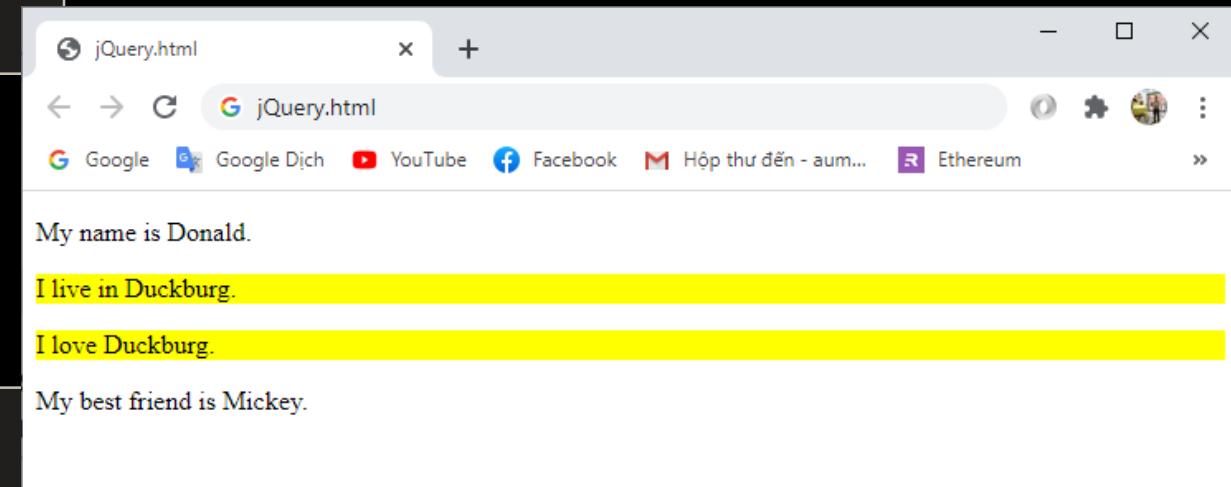
```
<script>
$(document).ready(function(){
  $("div").last().css("background-color",
"yellow");
});
</script>
```



# jQuery Traversing – Filtering - filter()

```
<p>My name is Donald.</p>
<p class="intro">I live in
Duckburg.</p>
<p class="intro">I love Duckburg.</p>
<p>My best friend is Mickey.</p>
```

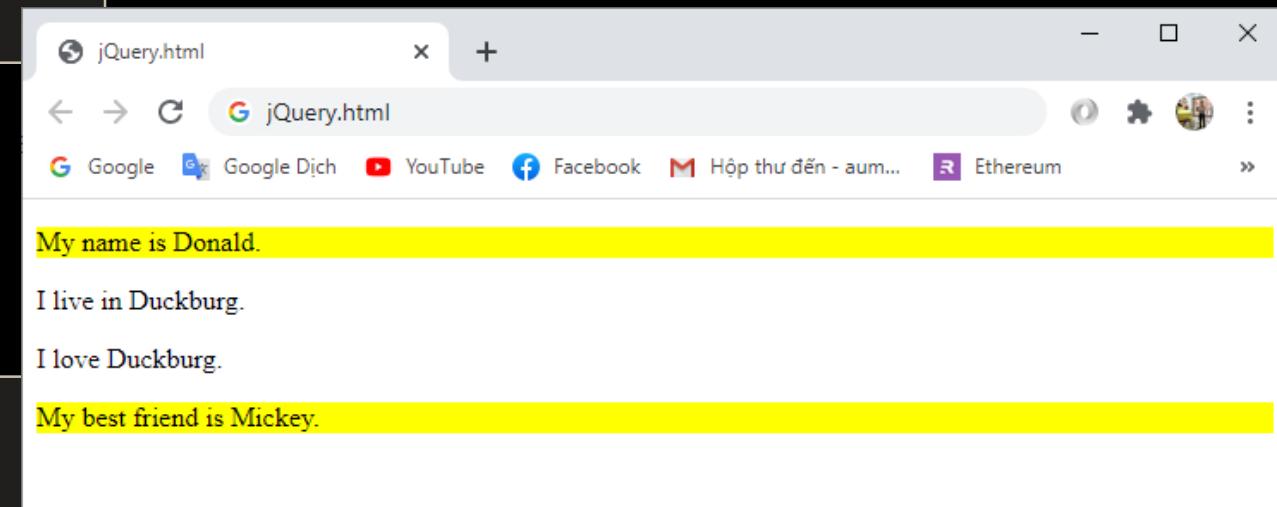
```
<script>
$(document).ready(function(){
    $("p").filter(".intro").css("background-
color", "yellow");
});
</script>
```



# jQuery Traversing – Filtering - not()

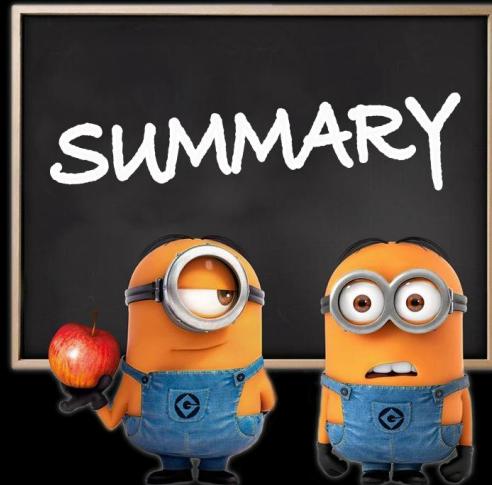
```
<p>My name is Donald.</p>
<p class="intro">I live in
Duckburg.</p>
<p class="intro">I love Duckburg.</p>
<p>My best friend is Mickey.</p>
```

```
<script>
$(document).ready(function(){
    $("p").not(".intro").css("background-
color", "yellow");
});
</script>
```



# Summary

- jQuery Event
  - Click, dblClick
  - Mouseenter, Mouseleave, Mousedown, Mouseup
  - Hover, focus, blur
  - on
- Traversal
  - Ancestors, Descendants
  - Siblings
  - Filtering



# jQuery - Event, Traversal



# Questions?

