jQuery Traversing – Ancestors

Traversing Up the DOM Tree

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

- parent()
- parents()
- parentsUntil()

jQuery parent() Method

The parent() method returns the direct parent element of the selected element.
This method only traverse a single level up the DOM tree.
The following example returns the direct parent element of each elements:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parent().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<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>

</body>
</html>
```

OUTPUT

jQuery parents() Method

The parents() method returns all ancestor elements of the selected element, all the way up to the document's root element (<html>).

The following example returns all ancestors of all elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("span").parents().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>

<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>

<!-- The outer red border, before the body element, is the html element (also an ancestor) -->
</html>
```

Output

You can also use an optional parameter to filter the search for ancestors.

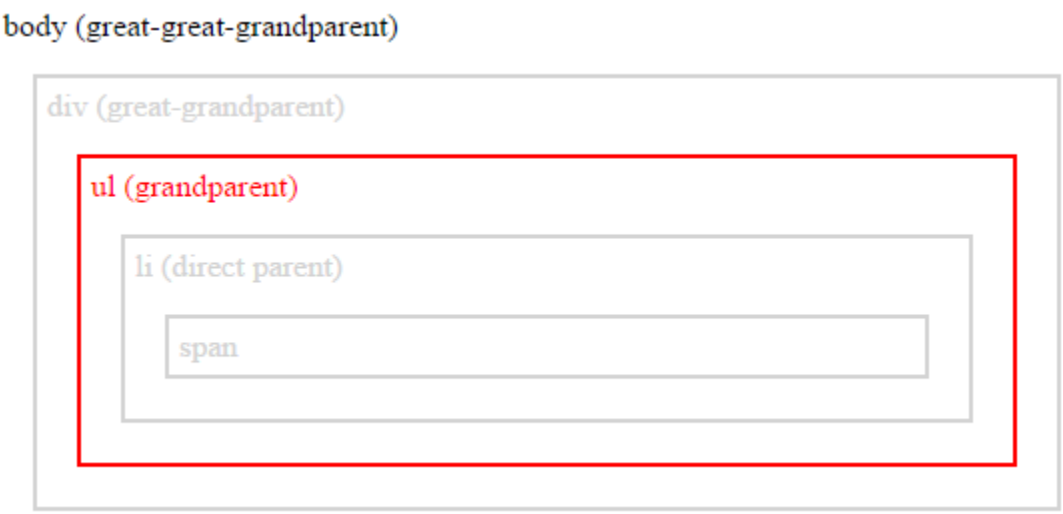
The following example returns all ancestors of all elements that are <ul> elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("span").parents("ul").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>

<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>
</html>
```

OUTPUT



jQuery parentsUntil() Method

The parentsUntil() method returns all ancestor elements between two given arguments.

The following example returns all ancestor elements between a and a element:

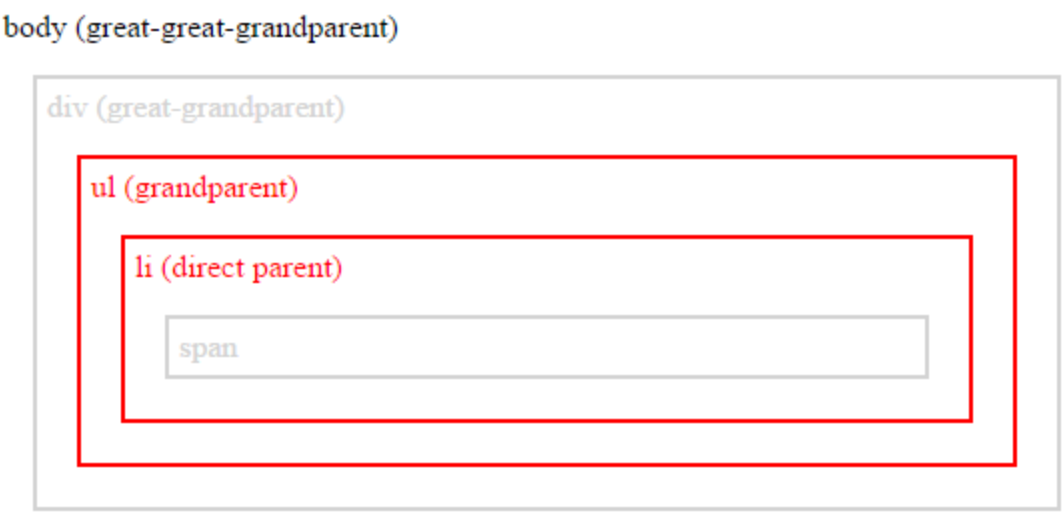
```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
```

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

<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>

</html>
```

Output



jQuery Traversing - Descendants

A descendant is a child, grandchild, great-grandchild, and so on.
With jQuery you can traverse down the DOM tree to find descendants of an element.
Traversing Down the DOM Tree

Two useful jQuery methods for traversing down the DOM tree are:

- children()
- find()

jQuery children() Method:-

The children() method returns all direct children of the selected element.

This method only traverse a single level down the DOM tree.

The following example returns all elements that are direct children of each elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("div").children().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<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>

</body>
</html>
```

You can also use an optional parameter to filter the search for children.

The following example returns all <p> elements with the class name "first", that are direct children of

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("div").children("p.first").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
```

```
  <p class="first">p (child)
    <span>span (grandchild)</span>
  </p>
  <p class="second">p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

jQuery find() Method

The find() method returns descendant elements of the selected element, all the way down to the last descendant.

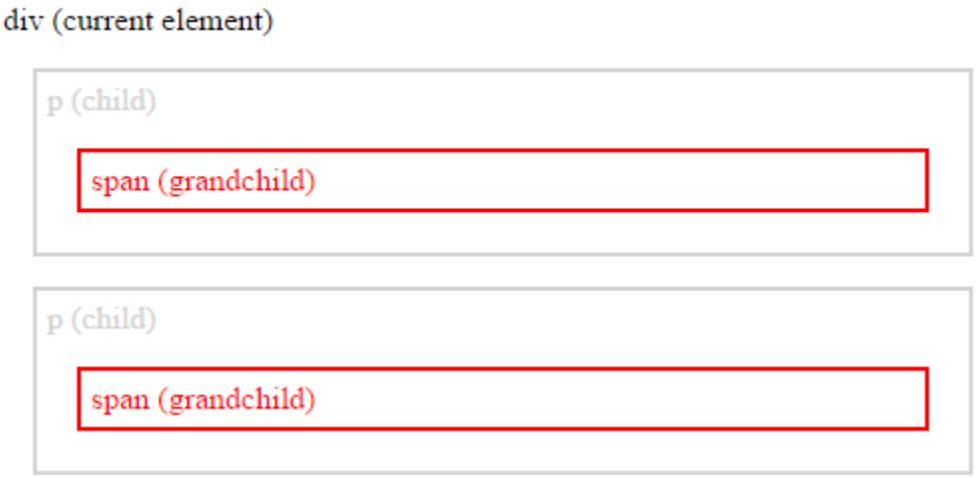
The following example returns all elements that are descendants of :

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
```

```
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("div").find("span").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<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>

</body>
</html>
```



The following example returns all descendants of :

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
.descendants * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("div").find("*").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<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>

</body>
</html>
```

jQuery Traversing - Siblings

There are many useful jQuery methods for traversing sideways in the DOM tree:

- siblings()
- next()
- nextAll()
- nextUntil()
- prev()
- prevAll()
- prevUntil()

jQuery siblings() Method

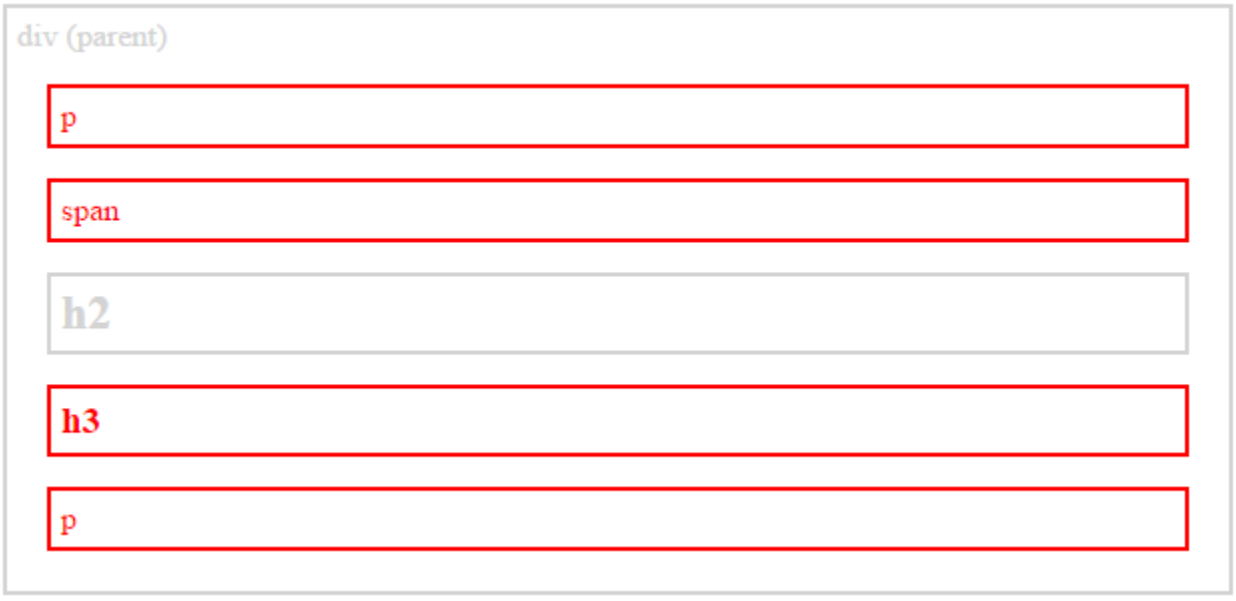
- The siblings() method returns all sibling elements of the selected element.
- The following example returns all sibling elements of <h2>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
```

```
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("h2").siblings().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



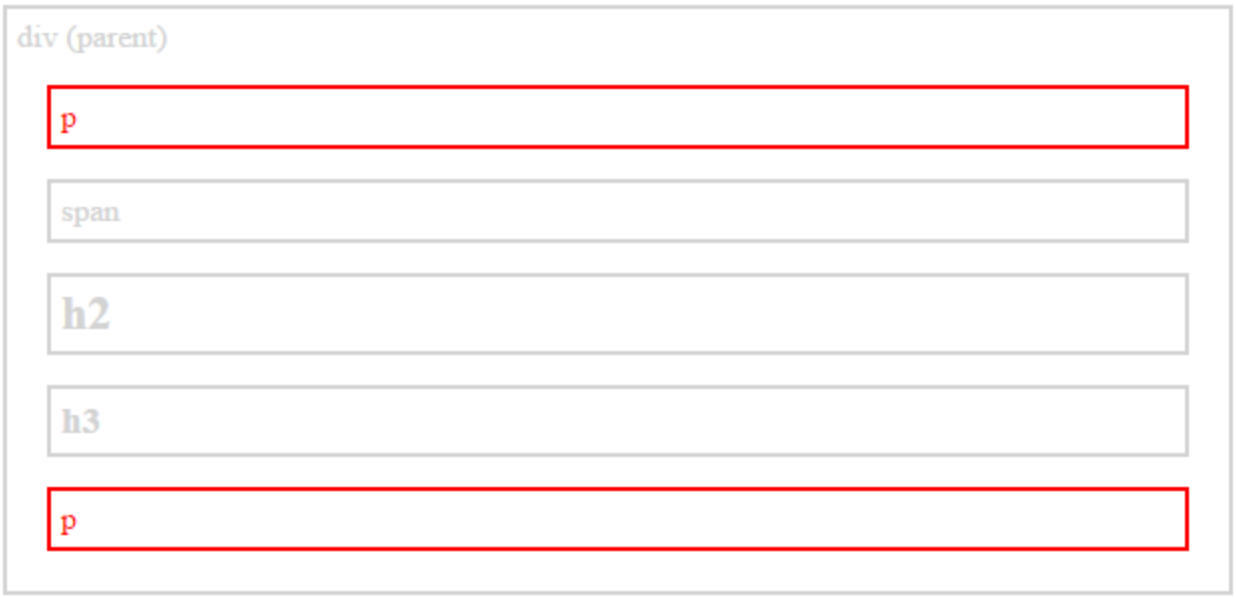
You can also use an optional parameter to filter the search for siblings.

The following example returns all sibling elements of <h2> that are <p> elements

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("h2").siblings("p").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



jQuery next() Method

The next() method returns the next sibling element of the selected element.

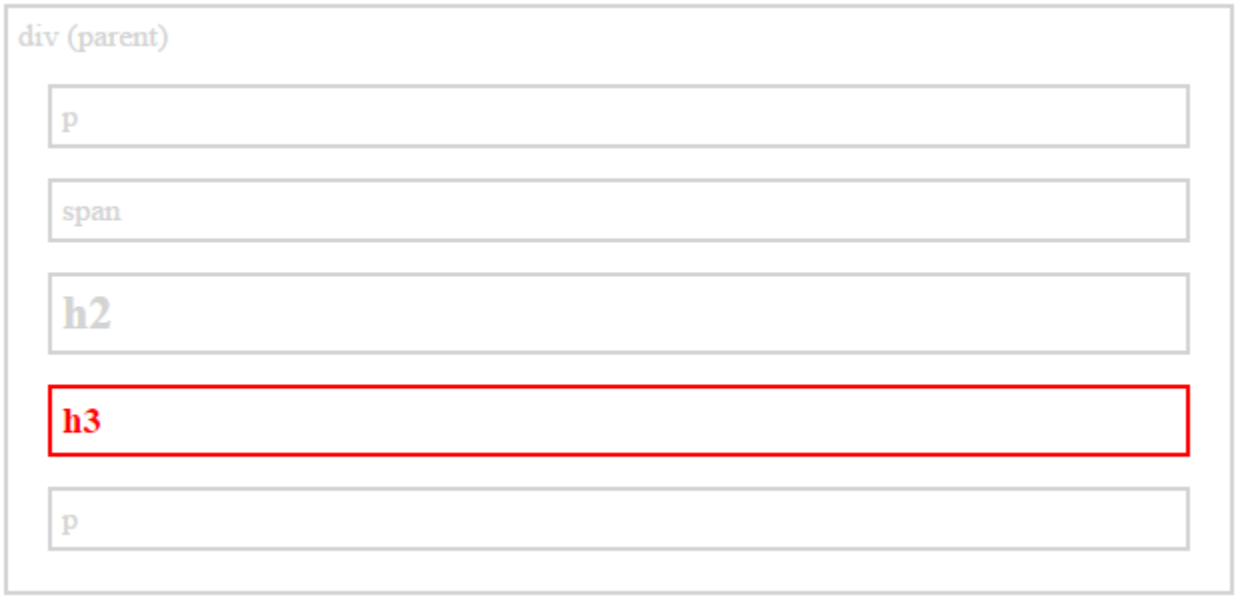
The following example returns the next sibling of <h2>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("h2").next().css({"color": "red", "border": "2px solid red"});
});
</script>
```

```
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



jQuery nextAll() Method

The nextAll() method returns all next sibling elements of the selected element.

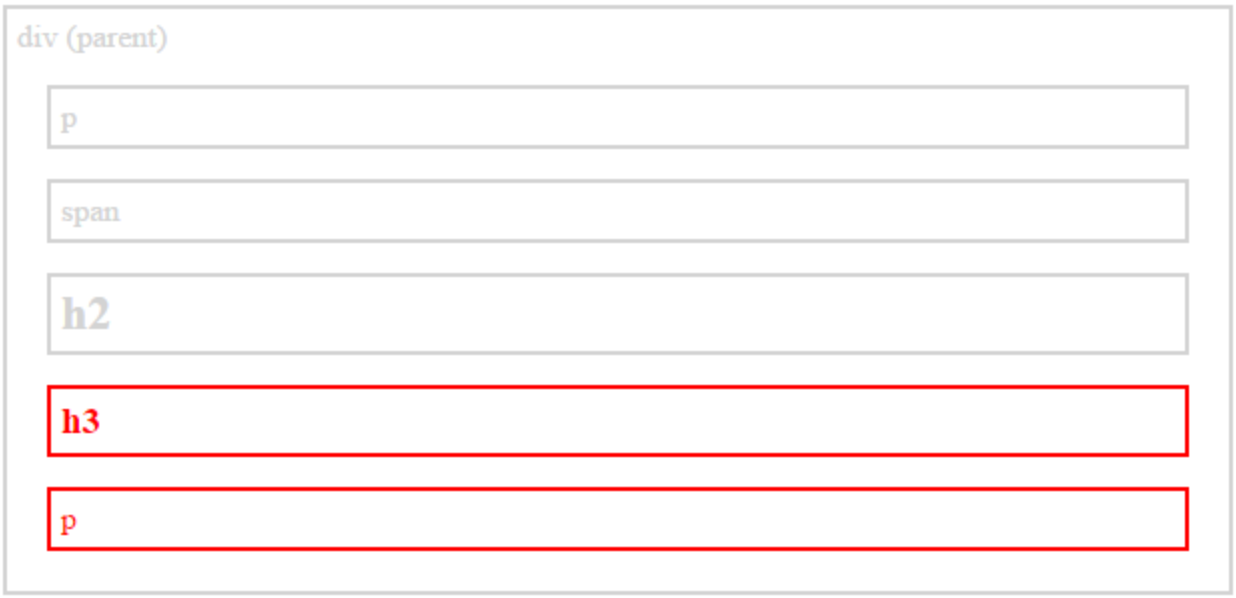
The following example returns all next sibling elements of <h2>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
```

```
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("h2").nextAll().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



jQuery nextUntil() Method

The nextUntil() method returns all next sibling elements between two given arguments.

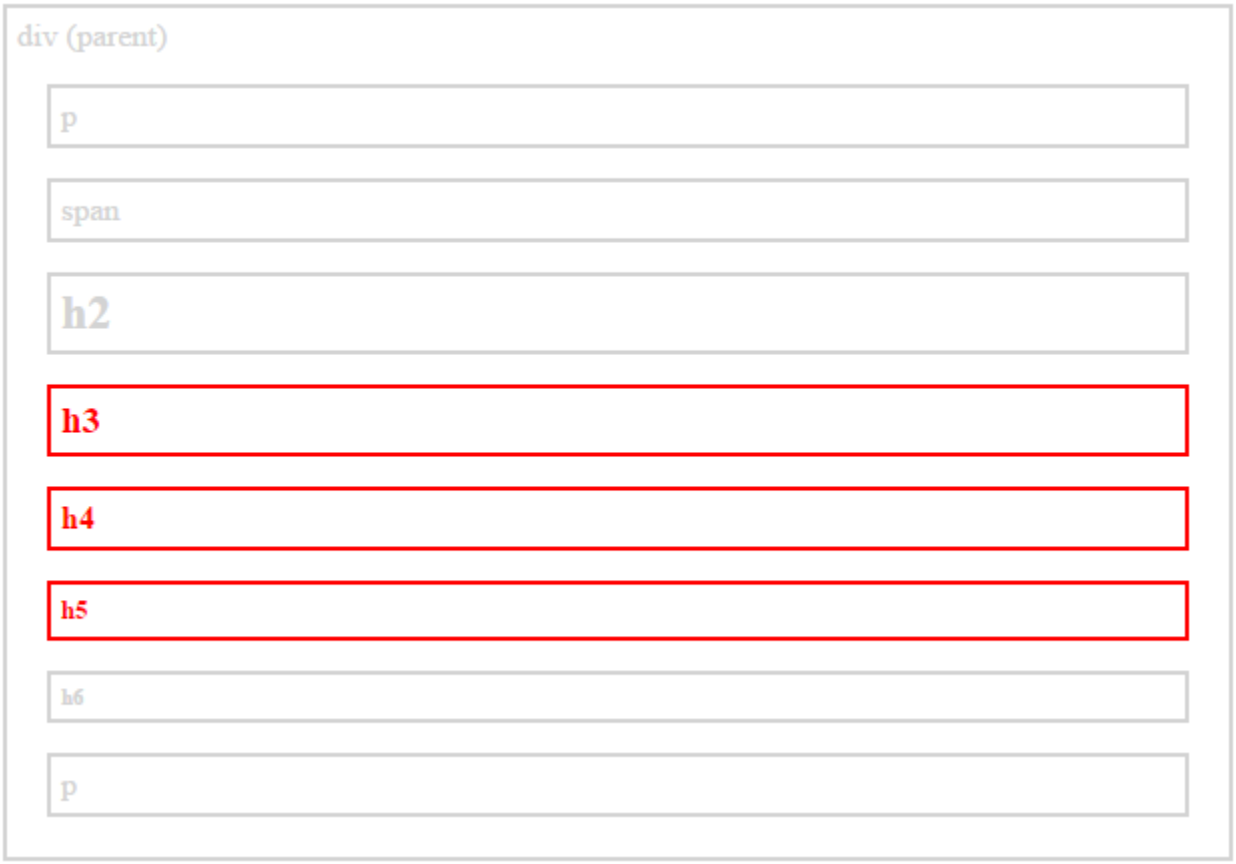
The following example returns all sibling elements between a <h2> and a <h6> element:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("h2").nextUntil("h6").css({"color": "red", "border": "2px solid red"});
});
</script>
```

```
</head>
<body class="siblings">

<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>

</body>
</html>
```



What is Traversing?

jQuery traversing, which means "move through", are used to "find" (or select) HTML elements based on their relation to other elements. Start with one selection and move through that selection until you reach the elements you desire.

The image below illustrates a family tree. With jQuery traversing, you can easily move up (ancestors), down (descendants) and sideways (siblings) in the family tree, starting from the selected (current) element. This movement is called traversing - or moving through - the DOM.

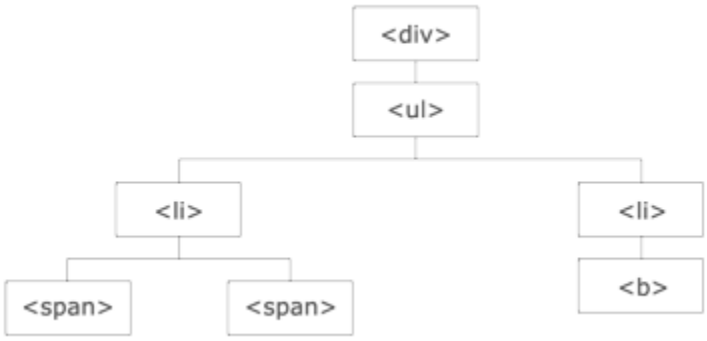


Illustration explained:

- The element is the **parent** of <ul>, and an **ancestor** of everything inside of it
- The <ul> element is the **parent** of both <li> elements, and a **child** of
- The left <li> element is the **parent** of , **child** of <ul> and a **descendant** of
- The element is a **child** of the left <li> and a **descendant** of <ul> and
- The two <li> elements are **siblings** (they share the same parent)
- The right <li> element is the **parent** of <b>, **child** of <ul> and a **descendant** of
- The <b> element is a **child** of the right <li> and a **descendant** of <ul> and

An ancestor is a parent, grandparent, great-grandparent, and so on.
A descendant is a child, grandchild, great-grandchild, and so on.
Siblings share the same parent.