

# level 2

TRYjQuery



# Searching the DOM

TRYjQuery

# Selecting descendants

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

How do we find the `<li>` elements that are inside of the `<ul>` with a “destinations” ID?



descendant selector

## 2.1 Searching the DOM

TRYjQUERY

# Using the descendant selector

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

*the space matters*

```
$("#destinations li");
```

*parent*      *descendant*



## 2.1 Searching the DOM

TRYjQUERY

# Selecting direct children

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

How do we find only the `<li>` elements that are children of the “destinations” `<ul>`?



descendant selector?

## 2.1 Searching the DOM

TRYjQUERY

# Selecting direct children

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

```
$("#destinations li");
```



## 2.1 Searching the DOM

TRYjQUERY

# Selecting only direct children

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

How do we find only the `<li>` elements that are **direct** children of the “destinations” `<ul>` then?



child selector

## 2.1 Searching the DOM

TRYjQUERY

# Selecting only direct children

## HTML document

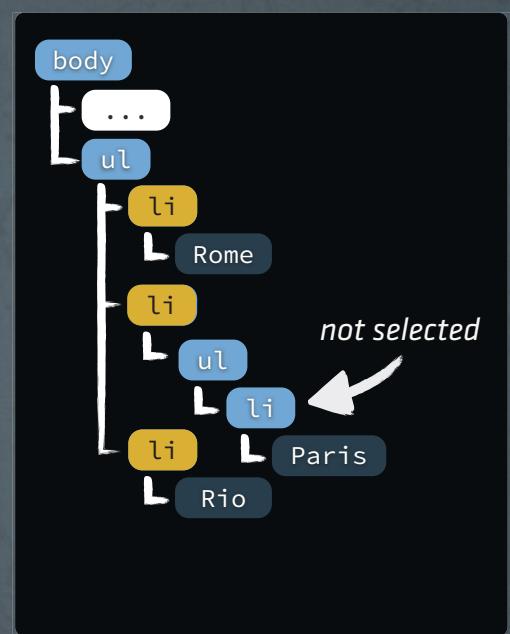
```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

`$("#destinations > li");`

parent

child

*the sign matters*



TRYjQUERY

# Selecting multiple elements

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

How do we find only elements with either a “promo” class or a “france” ID



multiple selector

## 2.1 Searching the DOM

TRYjQuery

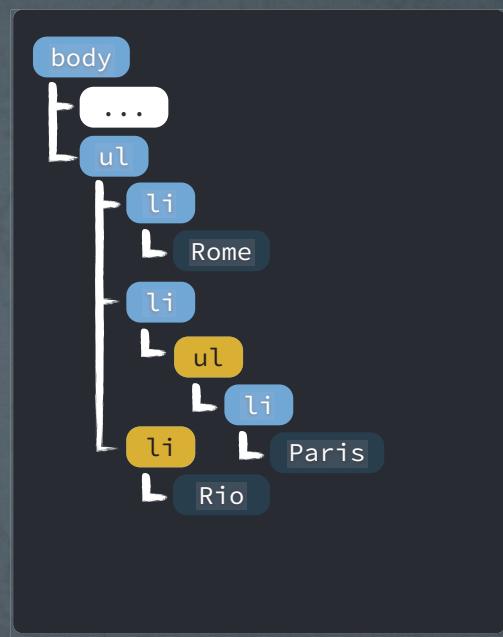
# Selecting multiple elements

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

*the comma matters*

```
$(".promo, #france");
```



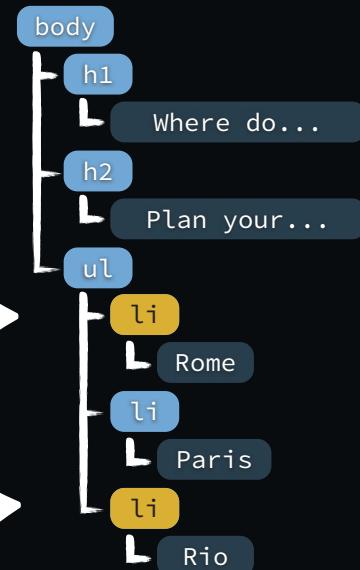
TRYjQuery

## CSS-like pseudo classes

```
$("#destinations li:first");
```

```
$("#destinations li:last");
```

*filter*



## 2.1 Searching the DOM

TRYjQuery

## CSS-like pseudo classes

```
$("#destinations li:odd");
```

```
$("#destinations li:even");
```

watch out, the index starts at 0



## 2.1 Searching the DOM

TRYjQuery



# Traversing

TRYjQuery

## Walking the DOM by traversing it

### HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

Can we find all the `<li>` elements that the “destinations” list contains without using a descendant selector?



filter by traversing

## 2.2 Traversing

TRYjQuery

## Filtering by traversing

```
$("#destinations li");
```



```
$("#destinations").find("li");
```



selection



traversal



It takes a bit more code, but it's faster.



TRYjQuery

## 2.2 Traversing

```
$("#li:first");
```



```
$("#li").first();
```



TRYjQuery

## 2.2 Traversing

## Filtering by traversing

```
$(“li:last”);
```



```
$(“li”).last();
```



TRYjQuery

## 2.2 Traversing

## Walking the DOM

### HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

Can we find the middle list item, knowing there is no filter to find it unlike :first or :last?



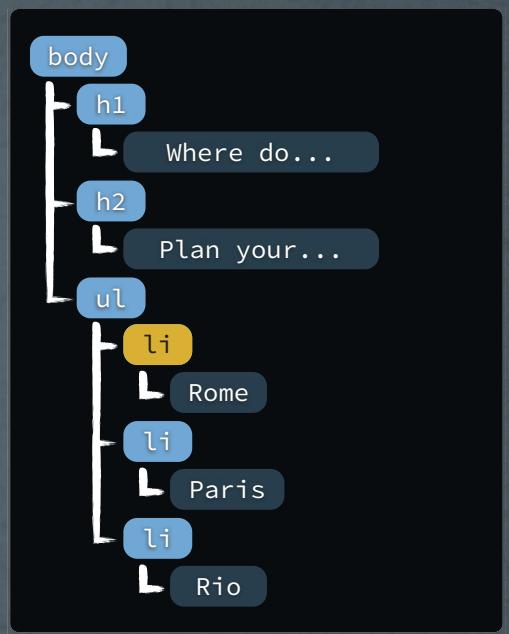
traversing

## 2.2 Traversing

TRYjQuery

## Walking the DOM

```
$( "li" ).first();
```



TRYjQuery

## 2.2 Traversing

## Walking the DOM

```
$( "li" ).first();
```

```
$( "li" ).first().next();
```



TRYjQuery

## 2.2 Traversing

# Walking the DOM

```
$(“li”).first();
```

```
$(“li”).first().next();
```

```
$(“li”).first().next().prev();
```



TRYjQuery

## 2.2 Traversing

# Walking up the DOM

### HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

If we started by selecting a child, can we figure out what element is its direct parent?



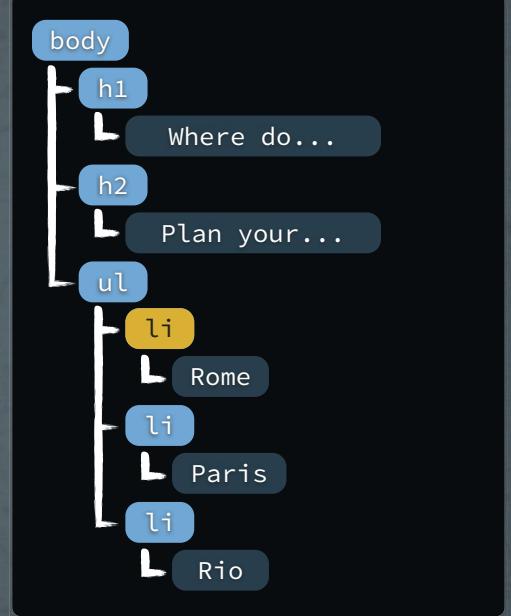
traversing up

## 2.2 Traversing

TRYjQuery

## Walking up the DOM

```
$( "li" ).first();
```



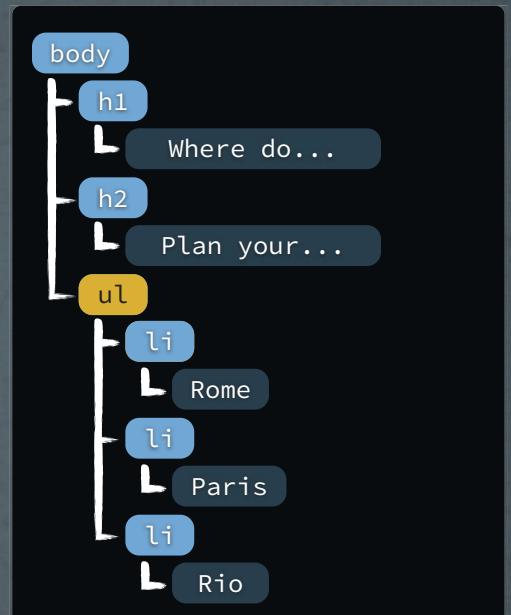
TRYjQuery

## 2.2 Traversing

### Walking up the DOM

```
$( "li" ).first();
```

```
$( "li" ).first().parent();
```



TRYjQuery

## 2.2 Traversing

# Walking down the DOM

## HTML document

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <ul id="france">
    <li>Paris</li>
  </ul>
  <li class='promo'>Rio</li>
</ul>
```

With a parent that has many children who in turn have their own children, how could we find only the first generation of children?



traversing down

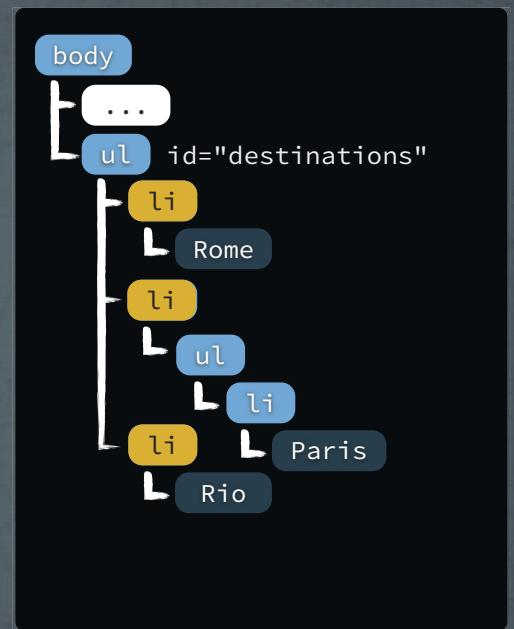
## 2.2 Traversing

TRYjQuery

# Walking the DOM up and down

```
$("#destinations").children("li");
```

children(), unlike find(), only selects direct children



## 2.2 Traversing

TRYjQuery