level 2

TRY jQuery

Searching the DOM

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

Using the descendant selector

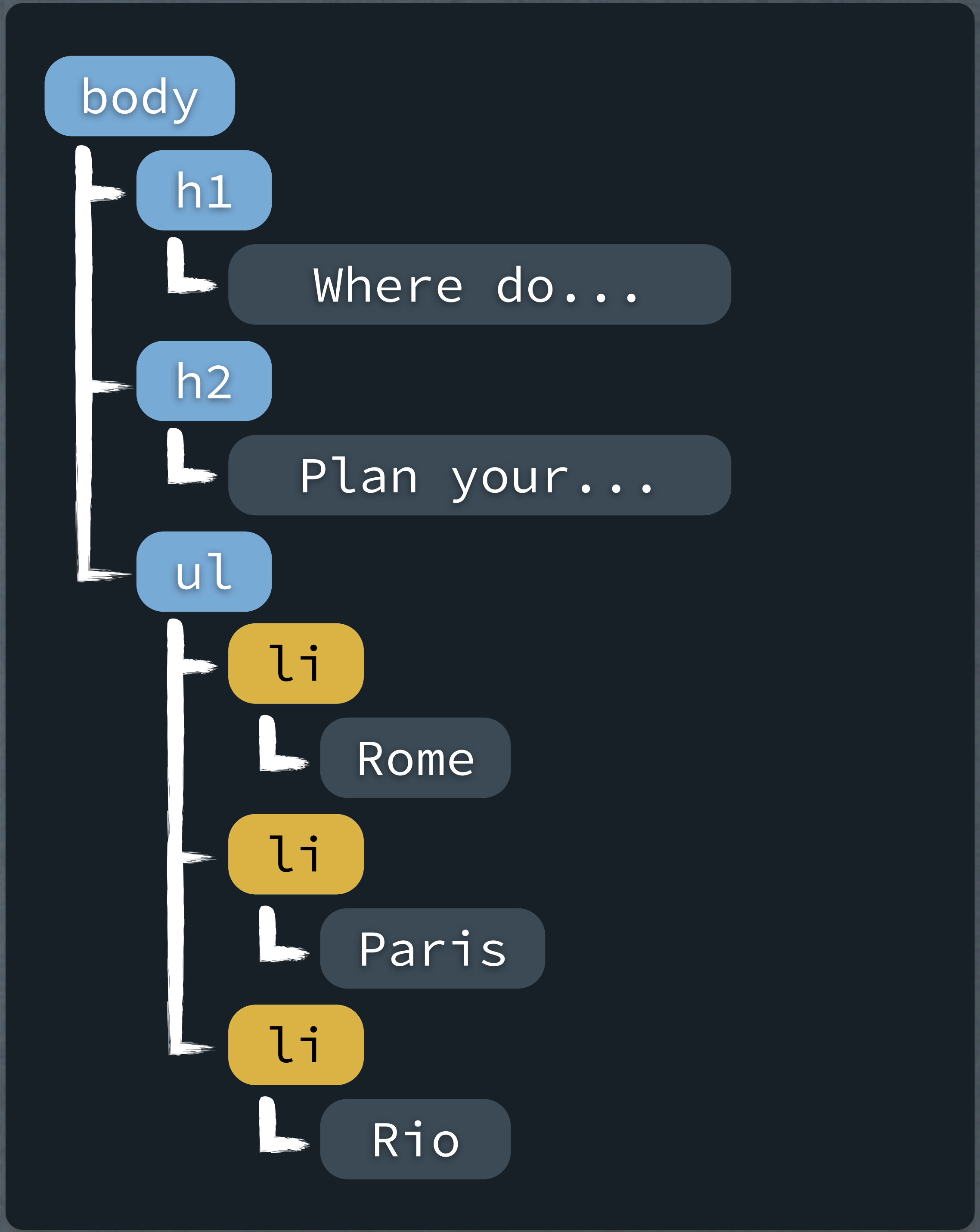
HTML document

```
<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*



Selecting direct children

HTML document

```html
<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>?

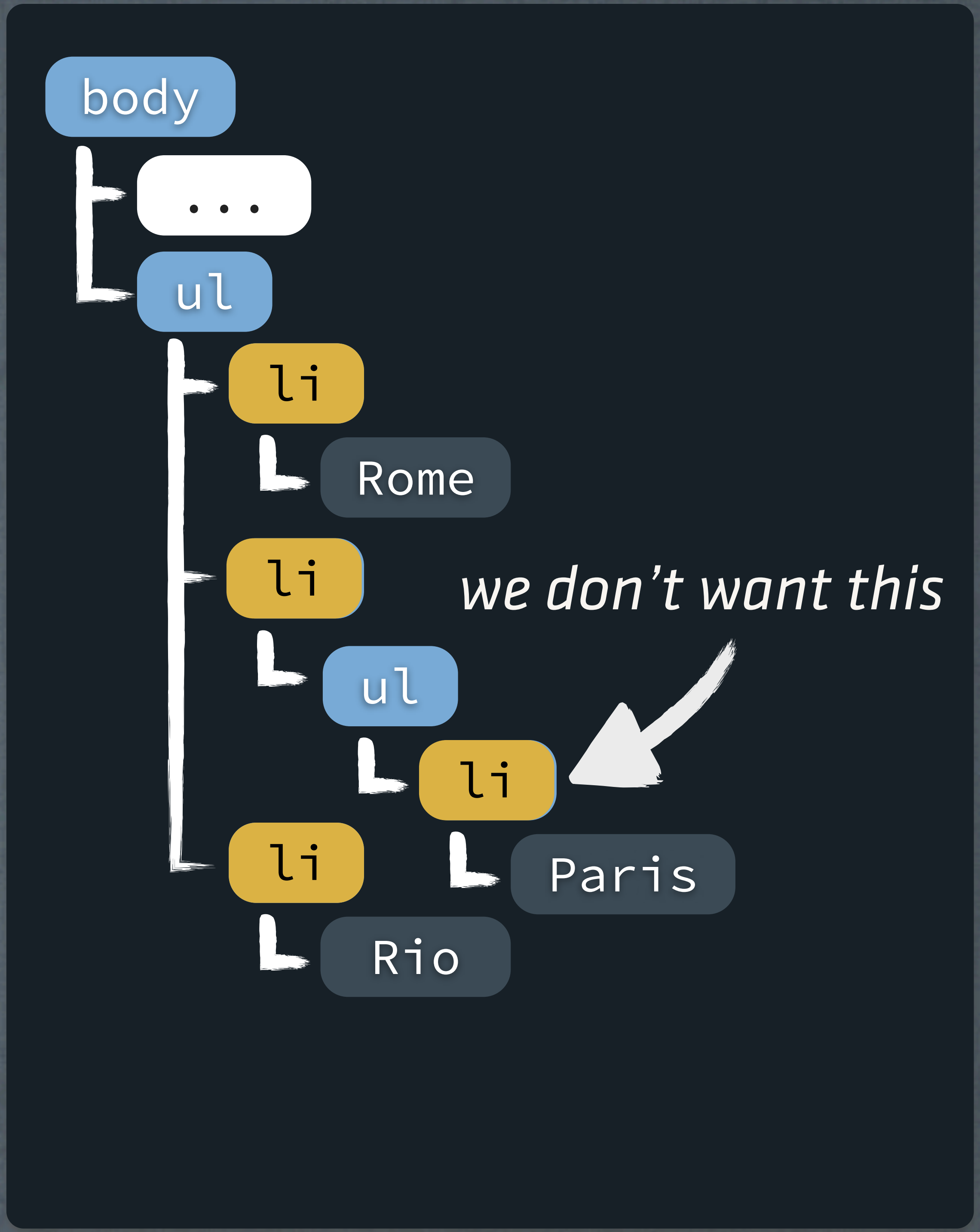
🔍 descendant selector?

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");
```



we don't want this

2.1 Searching the DOM

TRY jQuery

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>
</div>
```

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

🔍 child selector

Selecting only direct children

HTML document

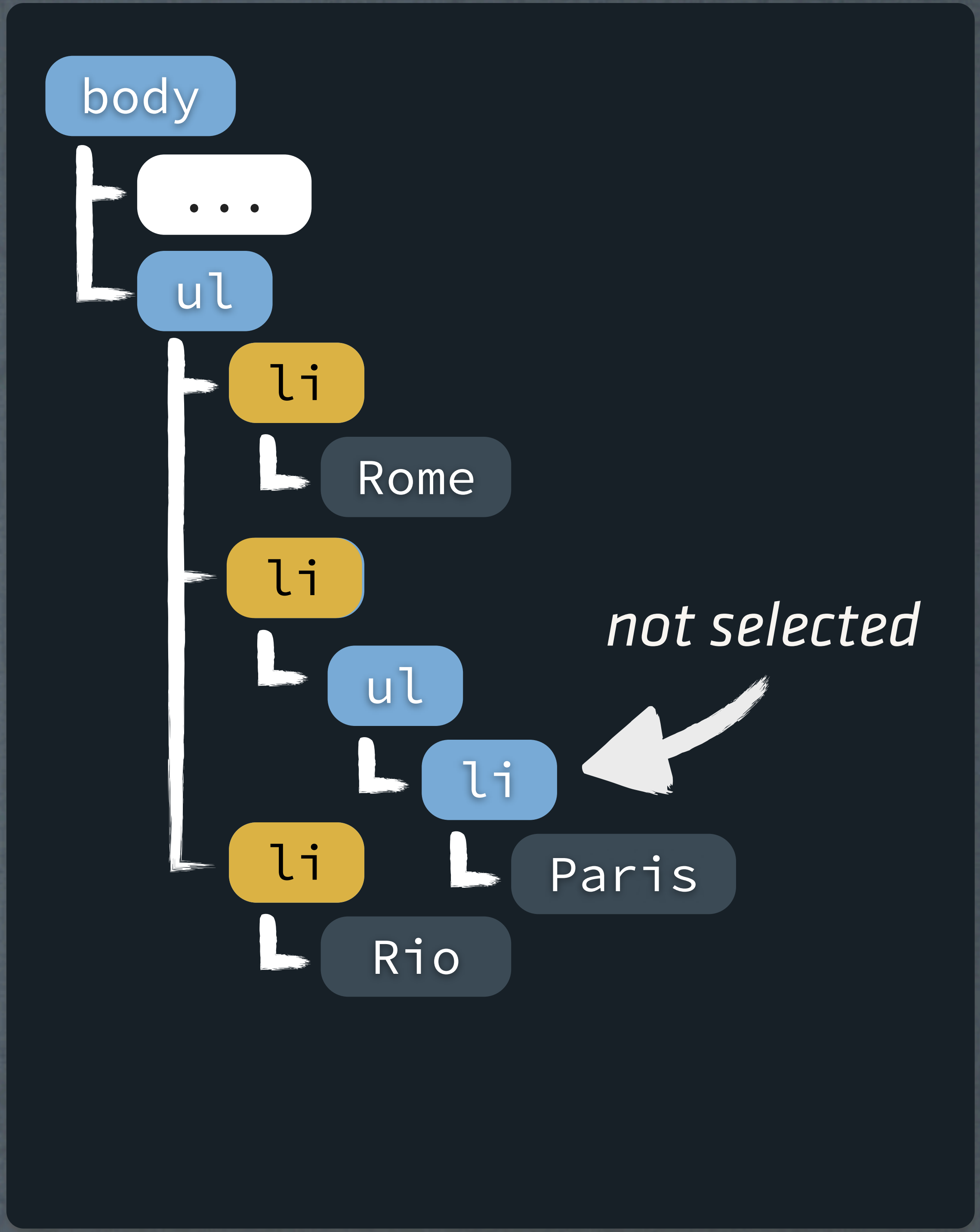
```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

the sign matters

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

parent

child



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

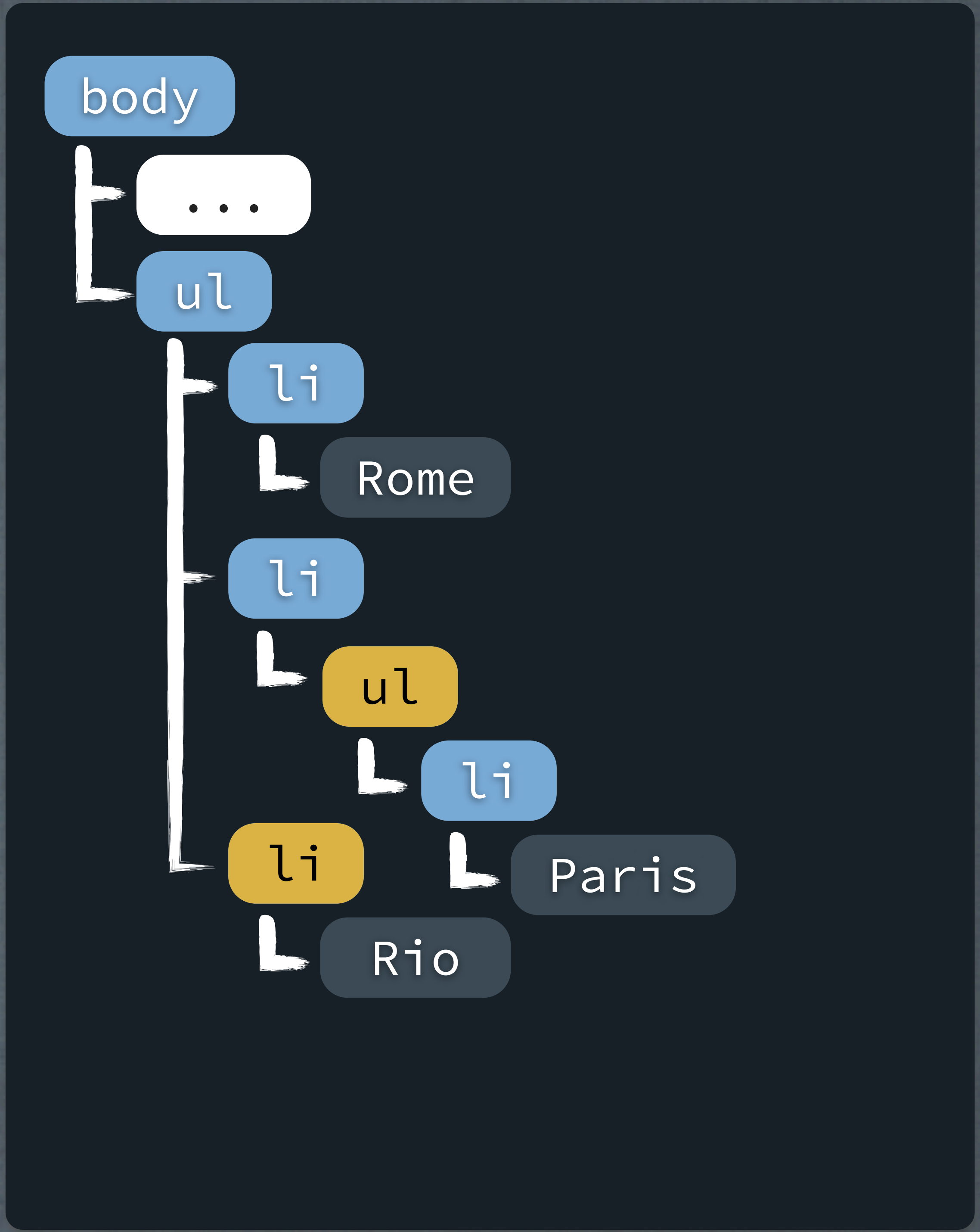
Selecting multiple elements

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");
```

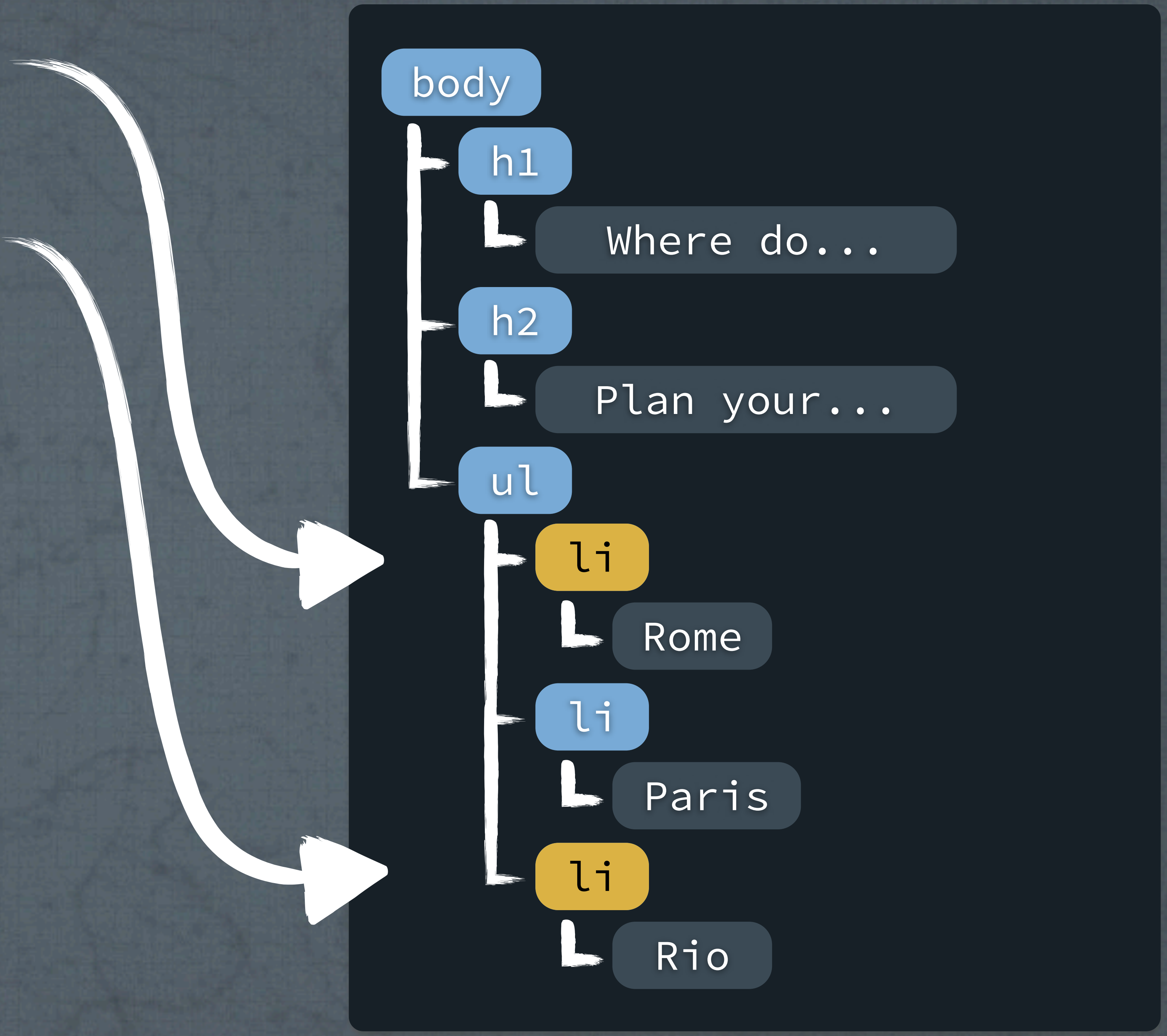


CSS-like pseudo classes

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

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

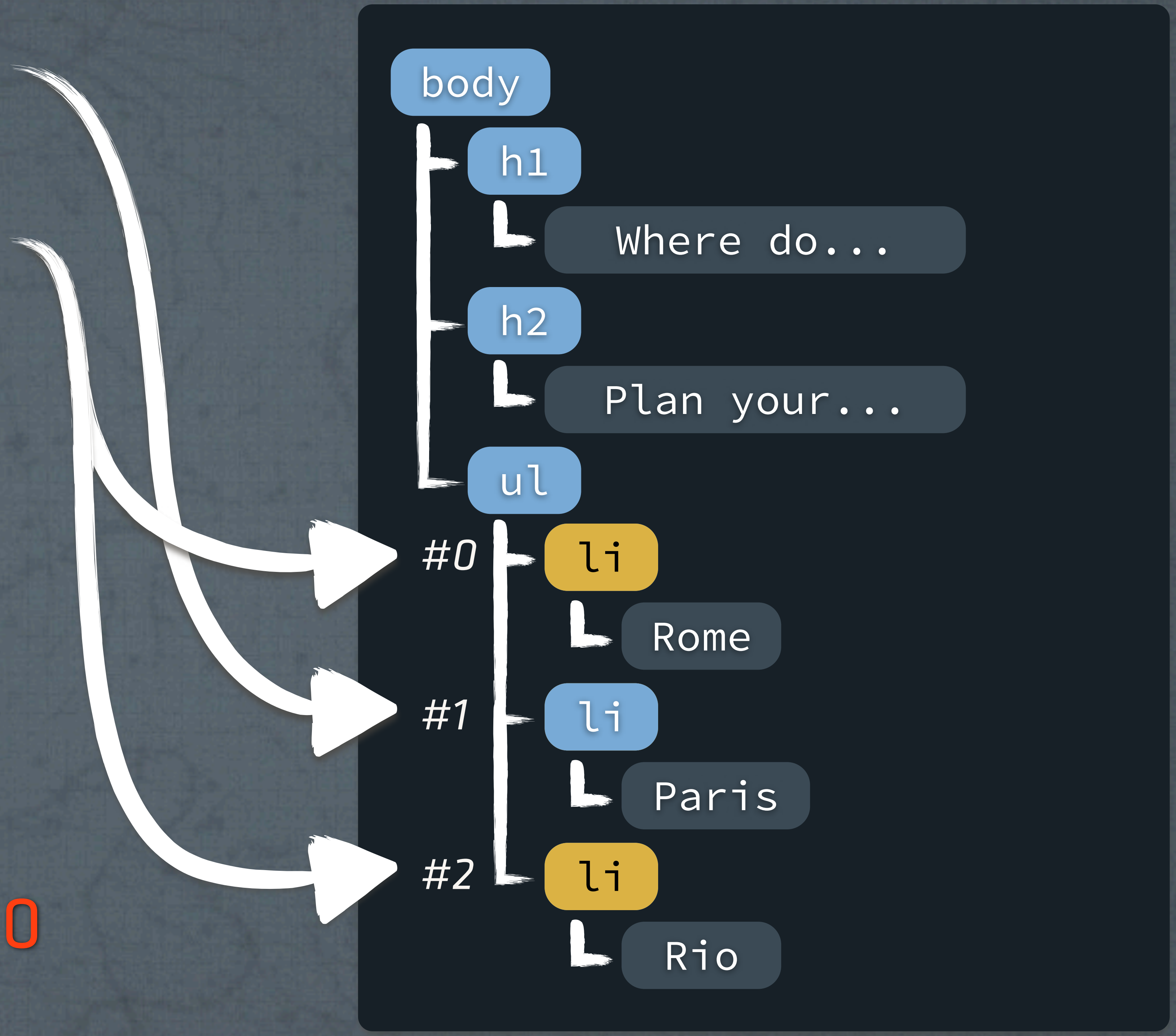
filter



CSS-like pseudo classes

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

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



⚠ watch out, the index starts at 0

2.1 Searching the DOM

Traversing

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

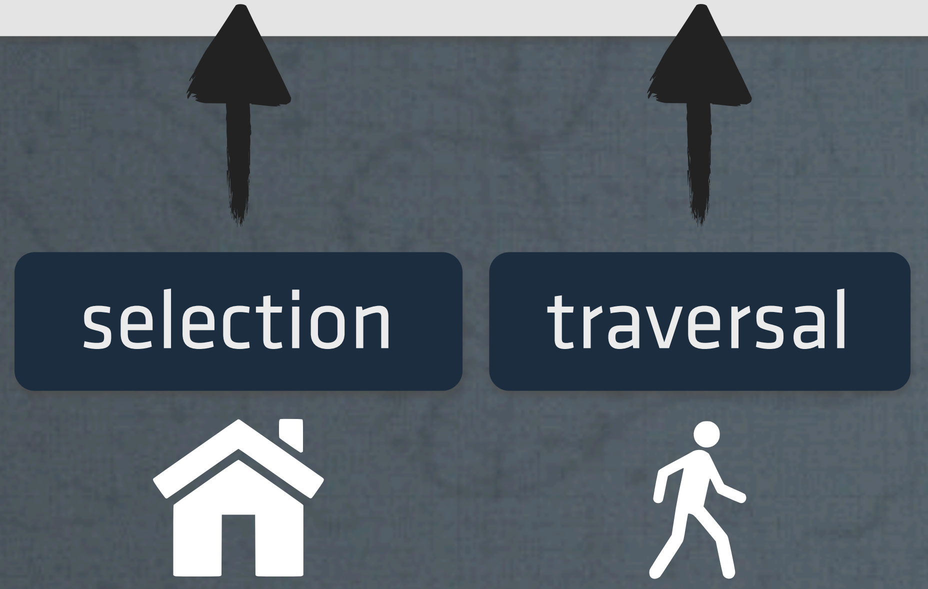
Filtering by traversing

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

!

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

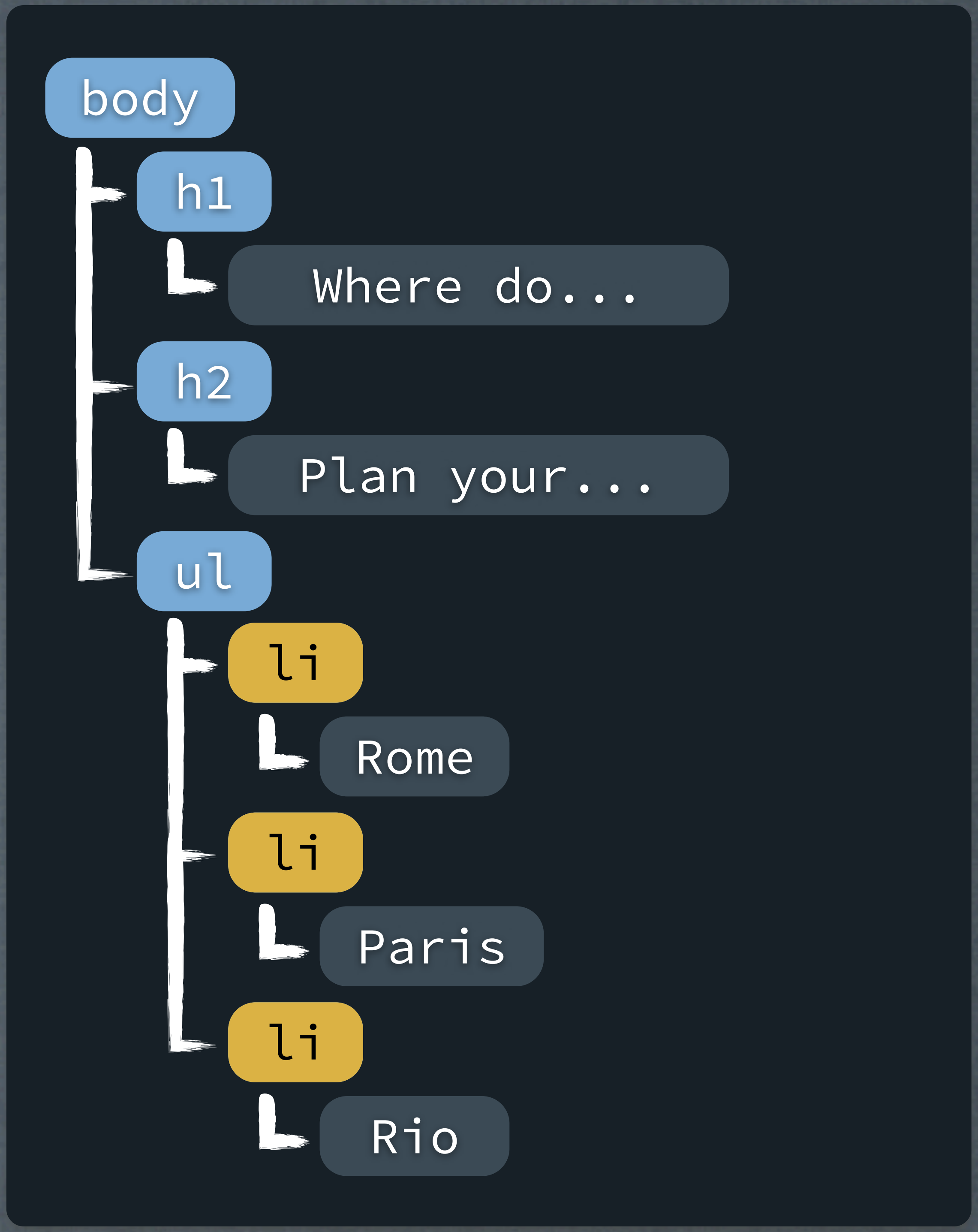
✓



selection

traversal

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



body

h1

Where do...

h2

Plan your...

ul

li

Rome

li

Paris

li

Rio

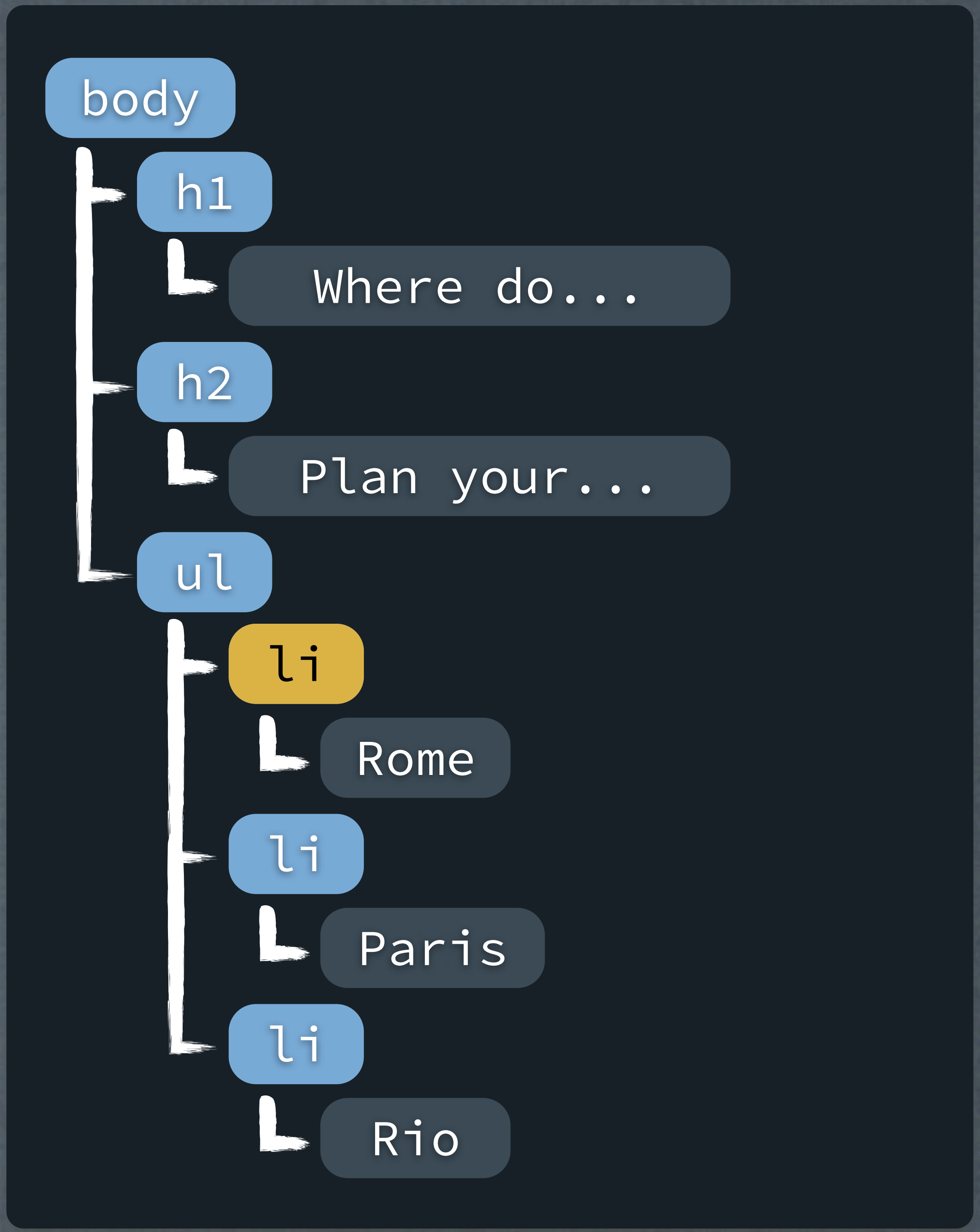
Filtering by traversing

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

!

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

✓



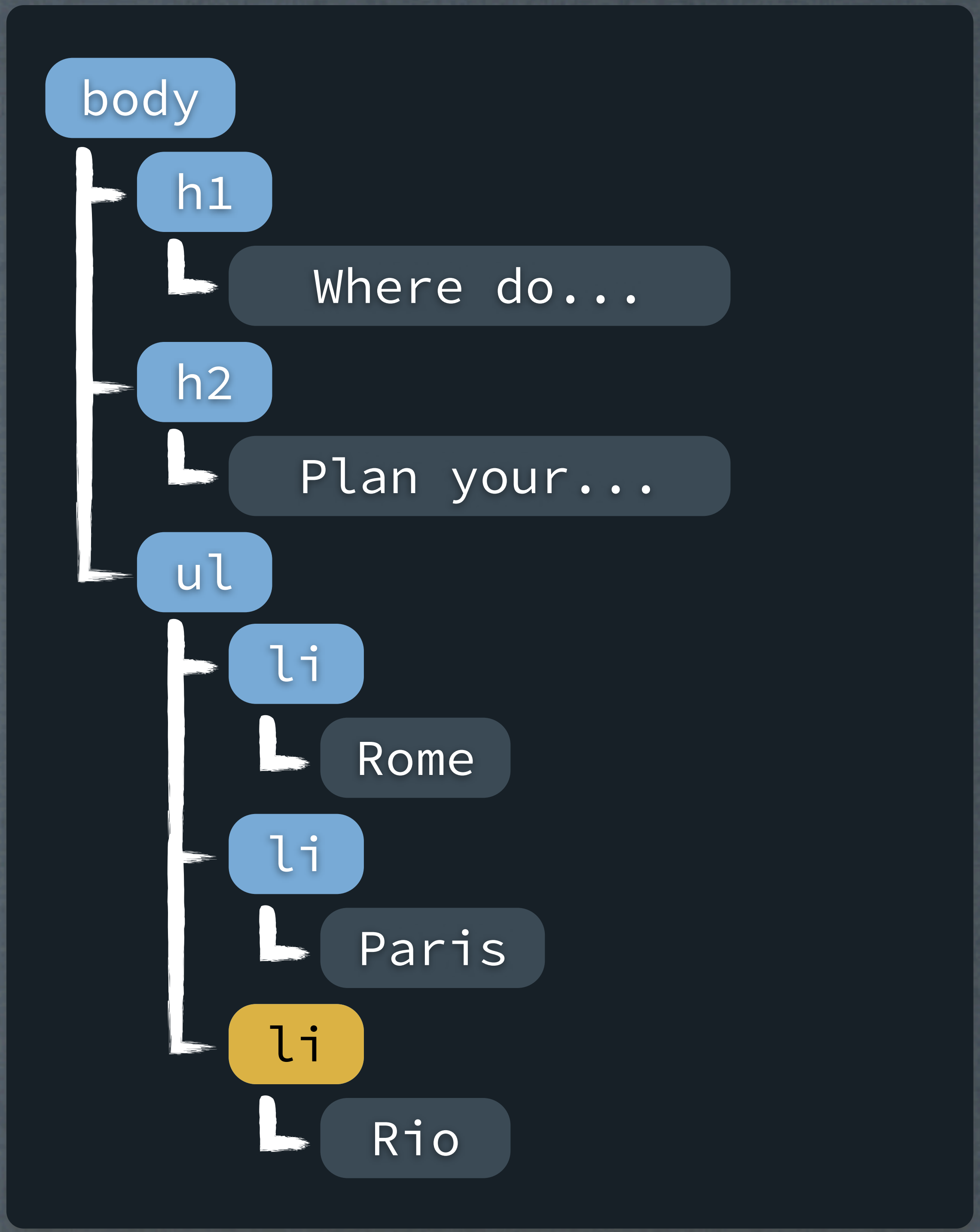
Filtering by traversing

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

!

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

✓



body
h1
Where do...
h2
Plan your...
ul
li
Rome
li
Paris
li
Rio

Walking the DOM

HTML document

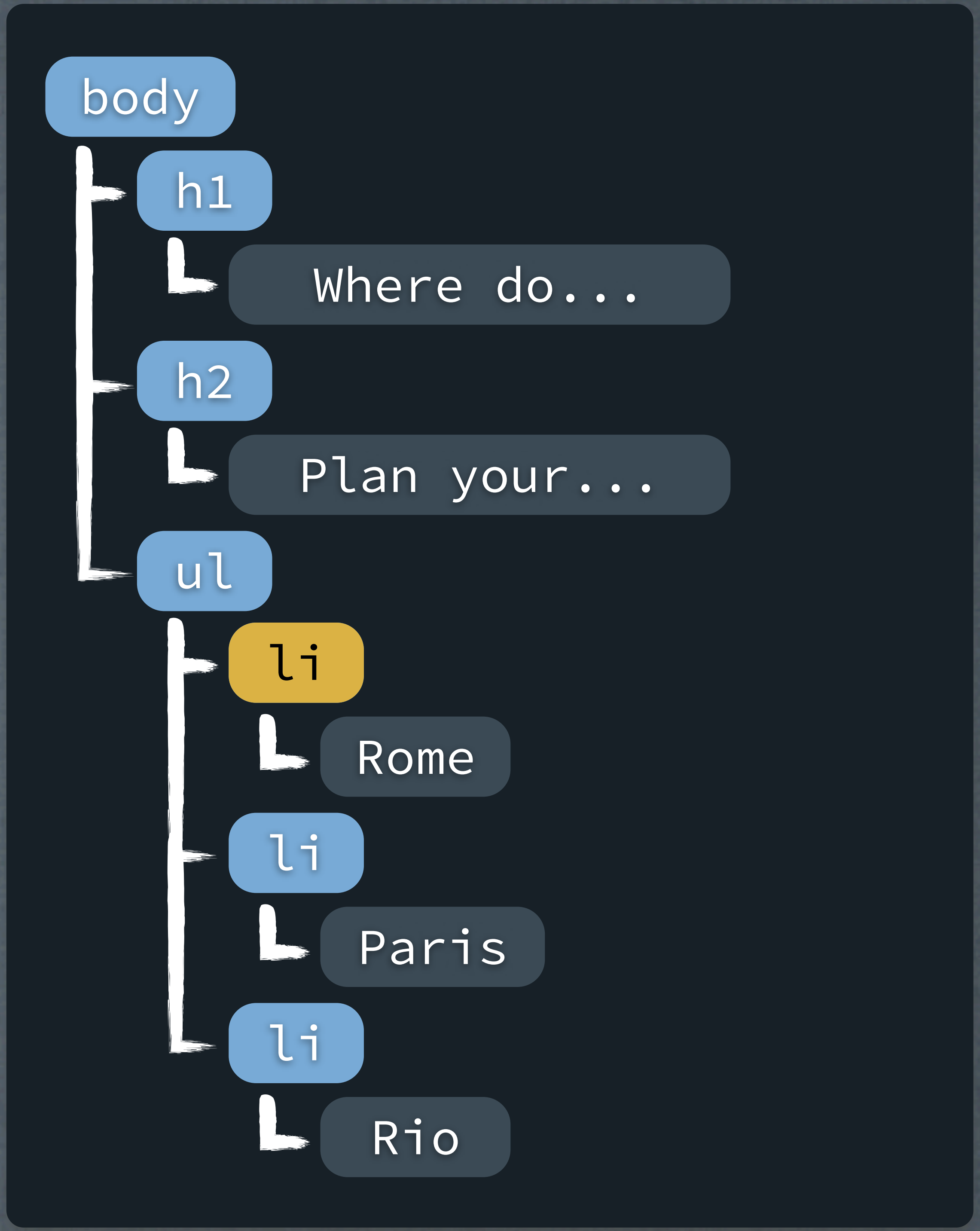
```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

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

traversing

Walking the DOM

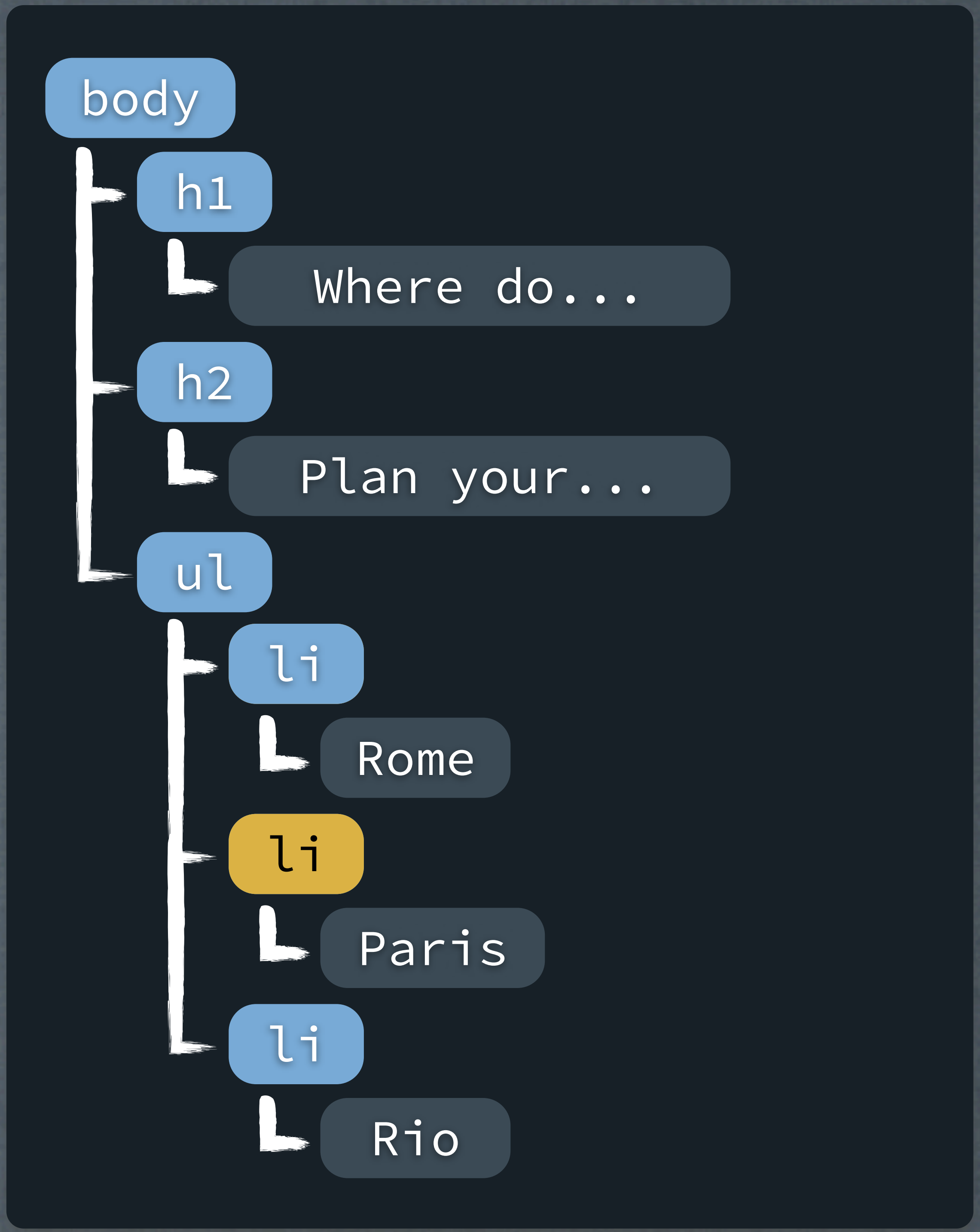
```
$("li").first();
```



Walking the DOM

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

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

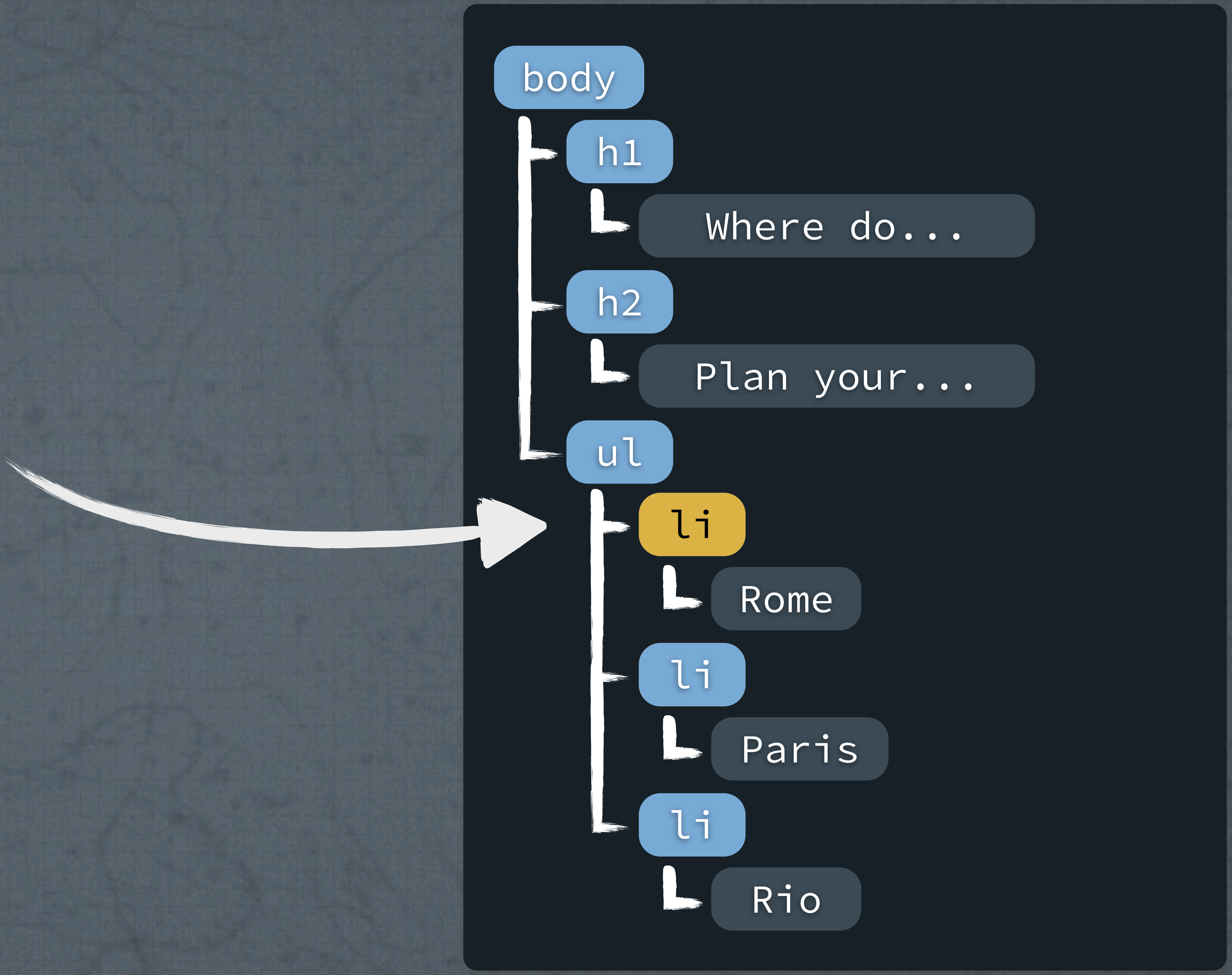


Walking the DOM

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

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

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



2.2 **Traversing**

Walking up the DOM

HTML document

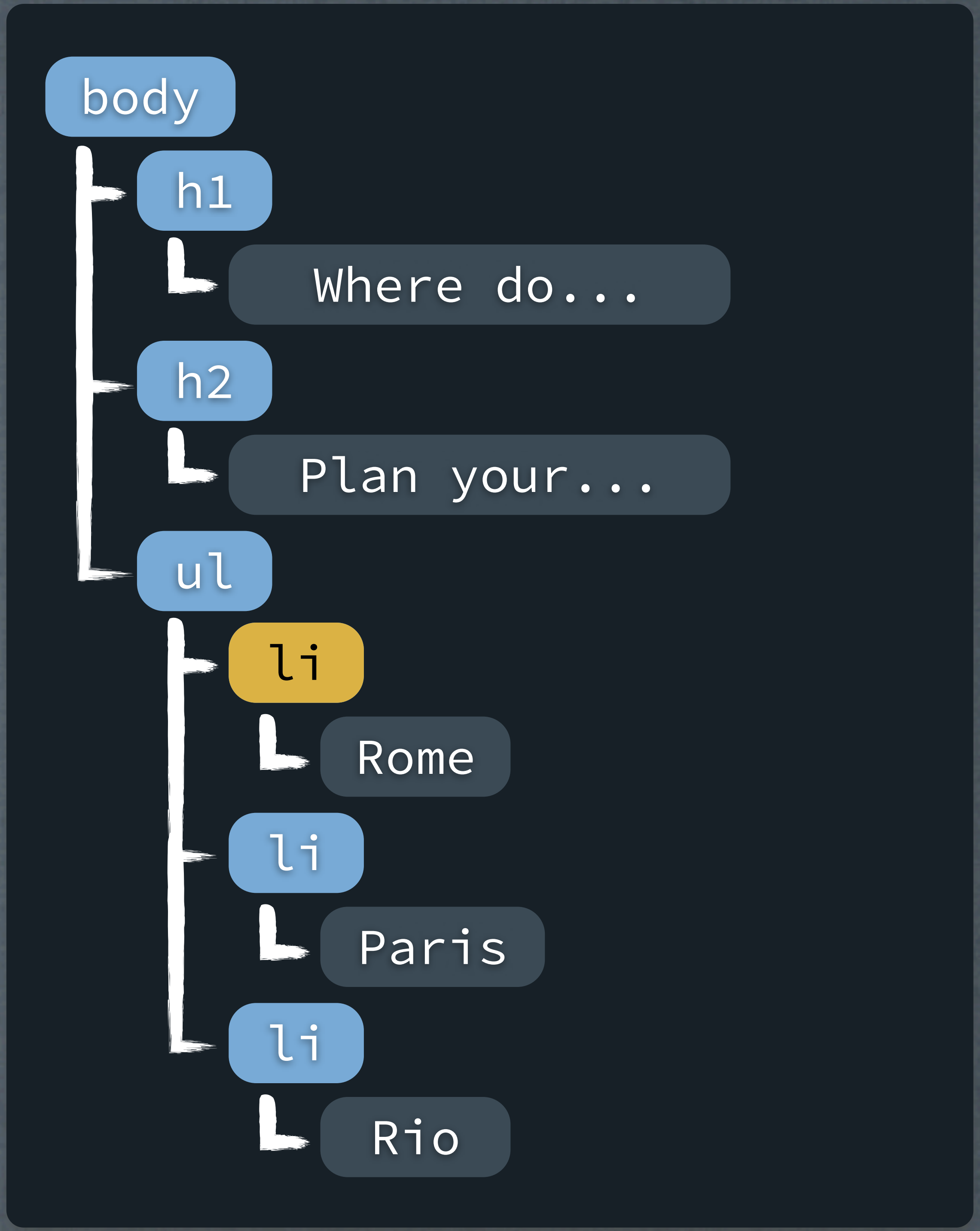
```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

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

traversing up

Walking up the DOM

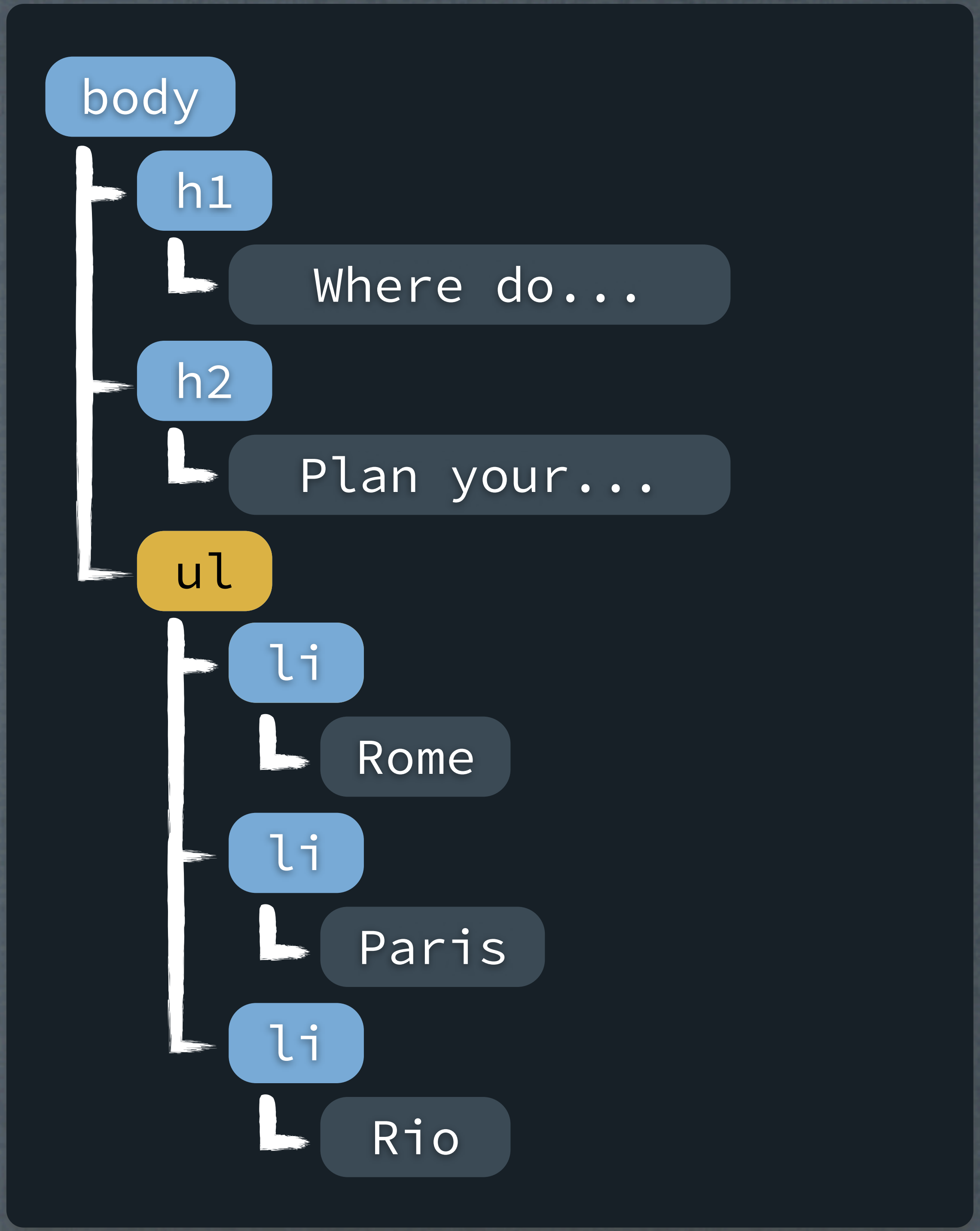
```
$("li").first();
```



Walking up the DOM

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

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



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>
```

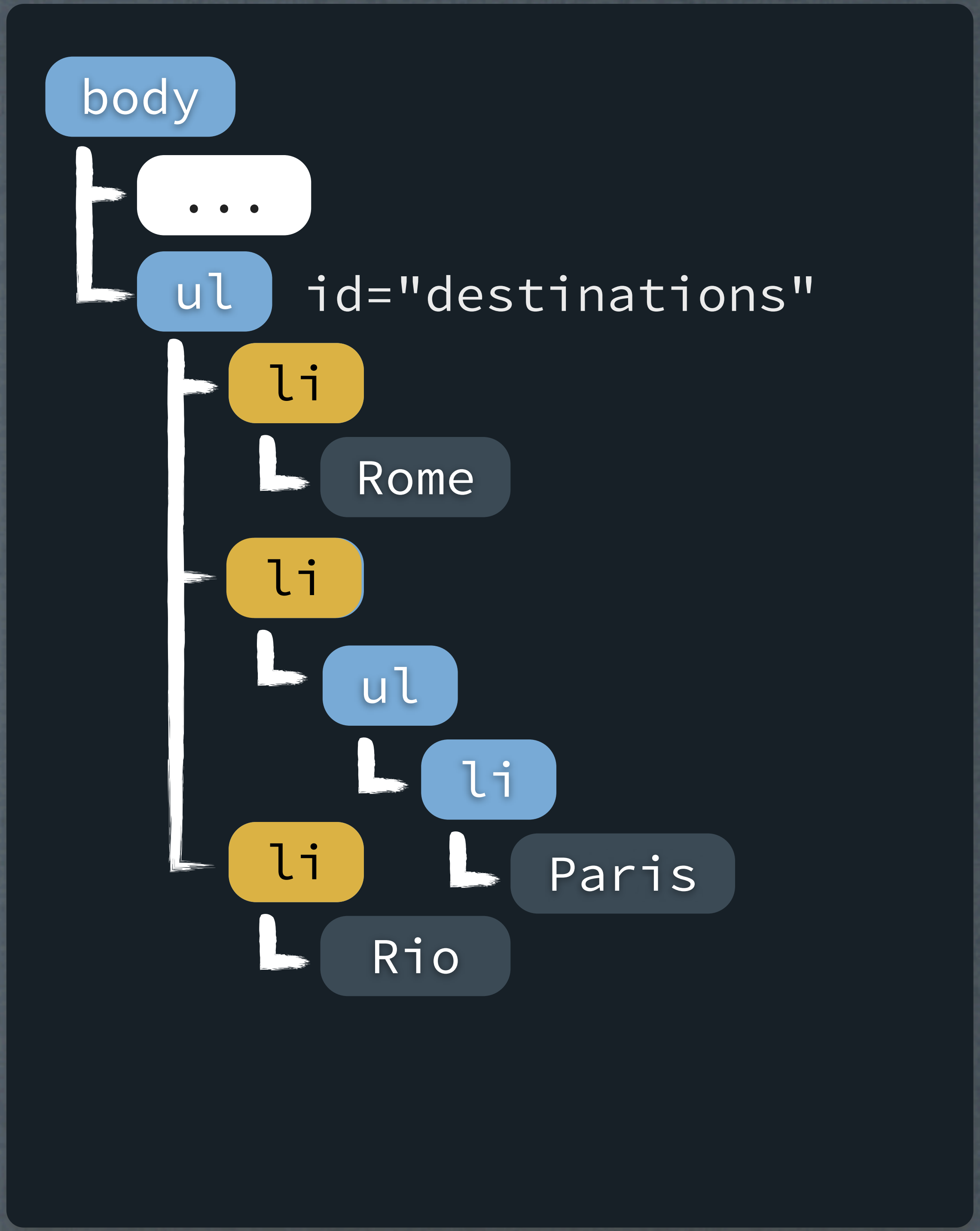
With a parent that has many children who in turn have their own children, how could we find only the first generation of children?

traversing down

Walking the DOM up and down

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

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



2.2 Traversing