COMP309
Web-based Technology

Dr. N. B. Gyan

Central University, Miotso. Ghana

Recap

Discussion of *possible* solution to Project 1

CSS (Pt. II)

How to apply your CSS to your HTML

There are three different ways to apply CSS to an HTML document that you'll commonly come across, some more useful than others.

- External stylesheet
- Internal stylesheet
- Inline stylesheet

External Stylesheet

- An *external stylesheet* is when you have your CSS written in a separate file with a **.css** extension, and you reference it from an HTML **<link>** element.

```
1 <!DOCTYPE html>
2 <html>
3  <head>
4   <meta charset="utf-8">
5   <title>My CSS experiment</title>
6   <link rel="stylesheet" href="style.css">
7  </head>
8 </html>
```

External Stylesheet

- This method is arguably the best, as you can use one stylesheet to style multiple documents, and would only need to update the CSS in one place if changes were needed.

Internal Stylesheet

- An *internal stylesheet* is where you don't have an external CSS file, but instead place your CSS inside a **<style>** element, contained inside the HTML *head*.

Internal Stylesheet

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>My CSS experiment</title>
  <style>
   h1 {
    color: blue;
    background-color: yellow;
    border: 1px solid black;
   }
```

Internal Stylesheet

```
12      p {
13        color: red;
14      }
15    </style>
16  </head>
17  <body>
18    <h1>Hello World</h1>
19    <p>This is my first CSS example</p
20  </body>
21 </html>
```

Internal Stylesheet

- This can be useful in some circumstances (maybe you're working with a content management system where you can't modify the CSS files directly).
- But it isn't quite as efficient as external stylesheets — in a website, the CSS would need to be repeated across every page, and updated in multiple places if changes were required.

Inline Styles

- Inline styles are CSS declarations that affect one element only, contained within a style attribute:

```
1 <!DOCTYPE html>
2 <html>
3  <head>
4   <meta charset="utf-8">
5   <title>My CSS experiment</title>
6  </head>
7  <body>
```

Inline Styles

```
8    <h1 style="color:blue;
9               background-color:yellow;
10              border:1px solid black;">
11     Hello World
12   </h1>
13
14   <p style="color:red;">
15    This is my first CSS example.</p>
16   </p>
17  </body>
18 </html>
```

Inline Styles

- Do not do this, *unless* you really have to! It is actually bad for maintenance (you might have to update the same information multiple times per document), and it also mixes your presentational CSS information with your HTML structural information, making the CSS harder to read and understand.

Inline Styles

- Keeping your different types of code separated and pure makes for a much easier job for all who work on the code.
- The only time you might have to resort to using inline styles is when your working environment is *really* restrictive.

Comments

- As with HTML, you are encouraged to make comments in your CSS, to help you understand how your code works when coming back to it after several months, and to help others understand it.
- Comments are also useful for temporarily commenting out certain parts of the code for testing purposes, for example if you are trying to find which part of your code is causing an error.
- Comments in CSS begin with /* and end with */.

Shorthand

- Some properties like **font**, **background**, **padding**, **border**, and **margin** are called *shorthand properties* — this is because they allow you to set several property values in a single line, saving time and making your code neater in the process.

Shorthand

```
padding: 10px 15px 15px 5px;
```

is the same as

```
padding-top:    10px;
padding-right:  15px;
padding-bottom: 15px;
padding-left:   5px;
```

Shorthand

```
background: red url(bg-graphic.png)
            10px 10px repeat-x fixed;
```

is the same as

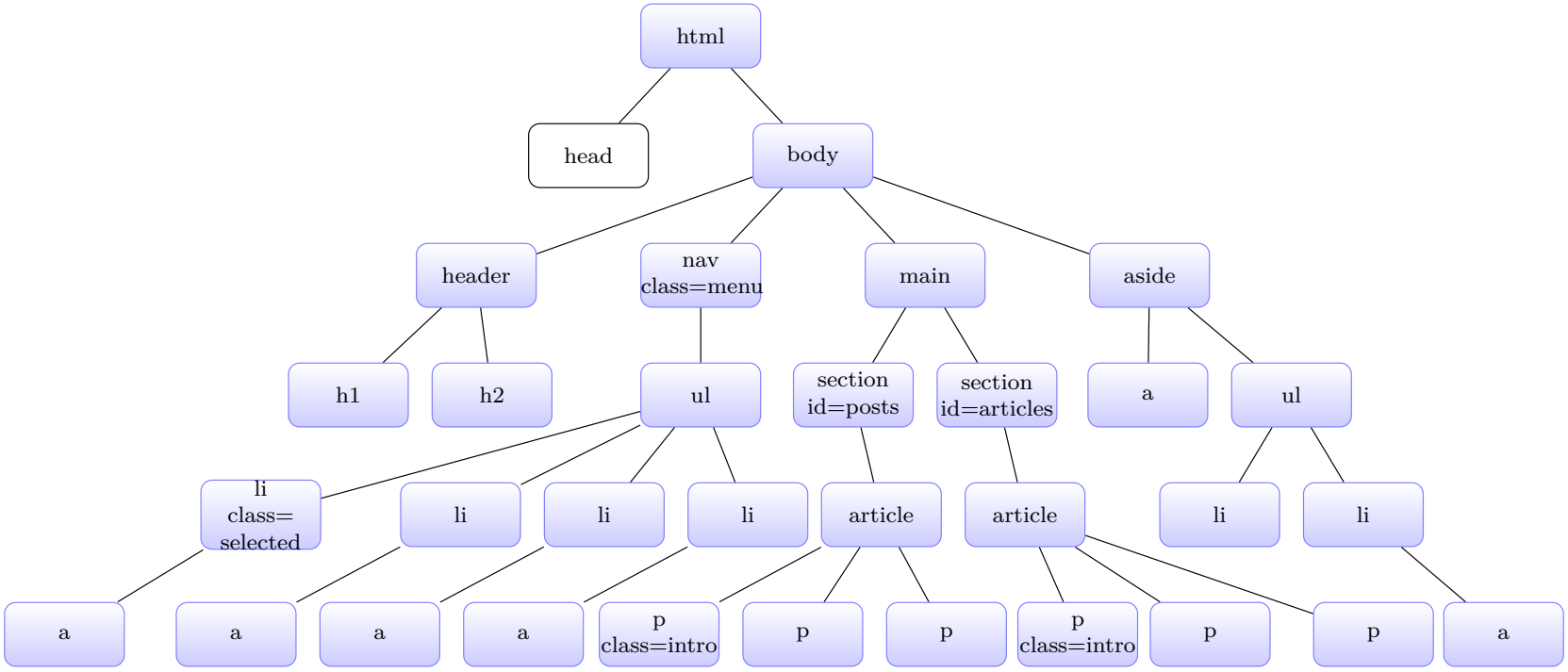
```
background-color: red;
background-image: url(bg-graphic.png);
background-position: 10px 10px;
background-repeat: repeat-x;
background-scroll:fixed;
```

Selectors & Properties

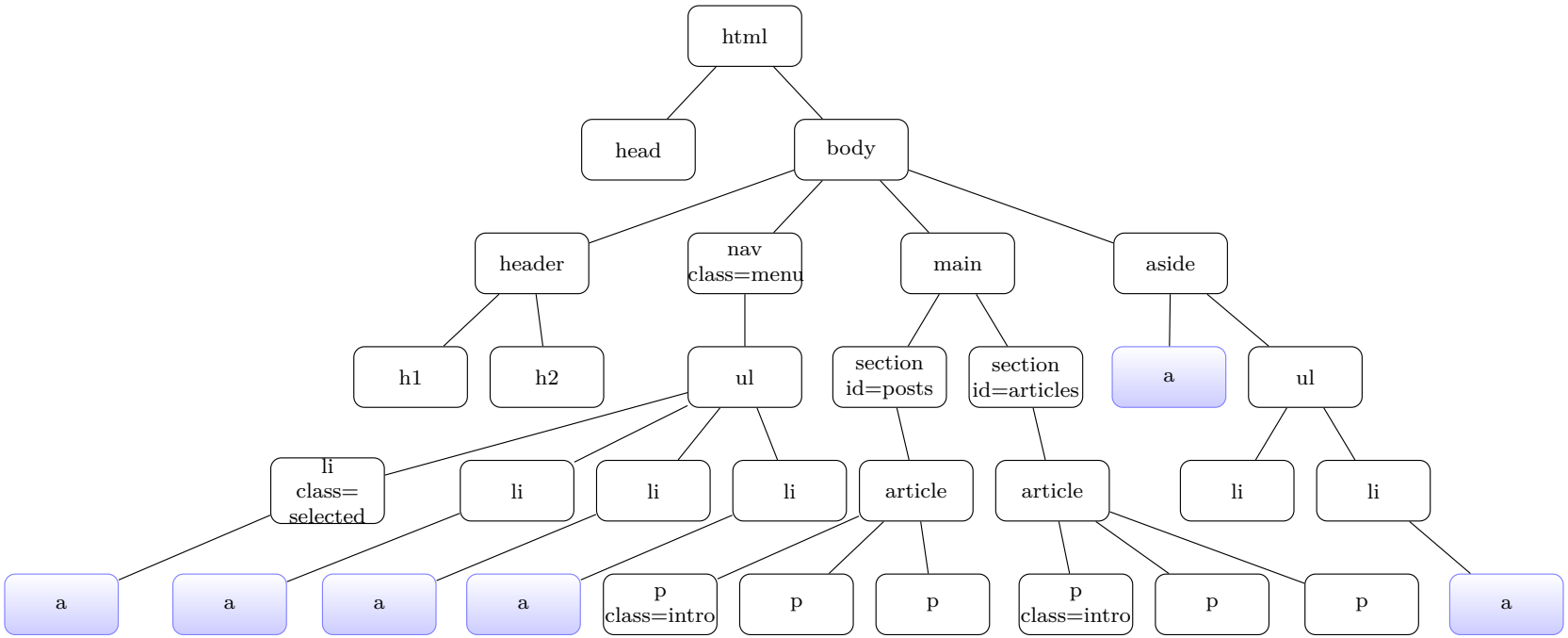
Selectors

- There are several types of selectors:
  - The **Universal**(*) selector.
  - **Type** selectors.
  - **Attribute** selectors.
  - **Class**(.) and **Id**(#) selectors.
  - Combinators
  - Grouping (,)

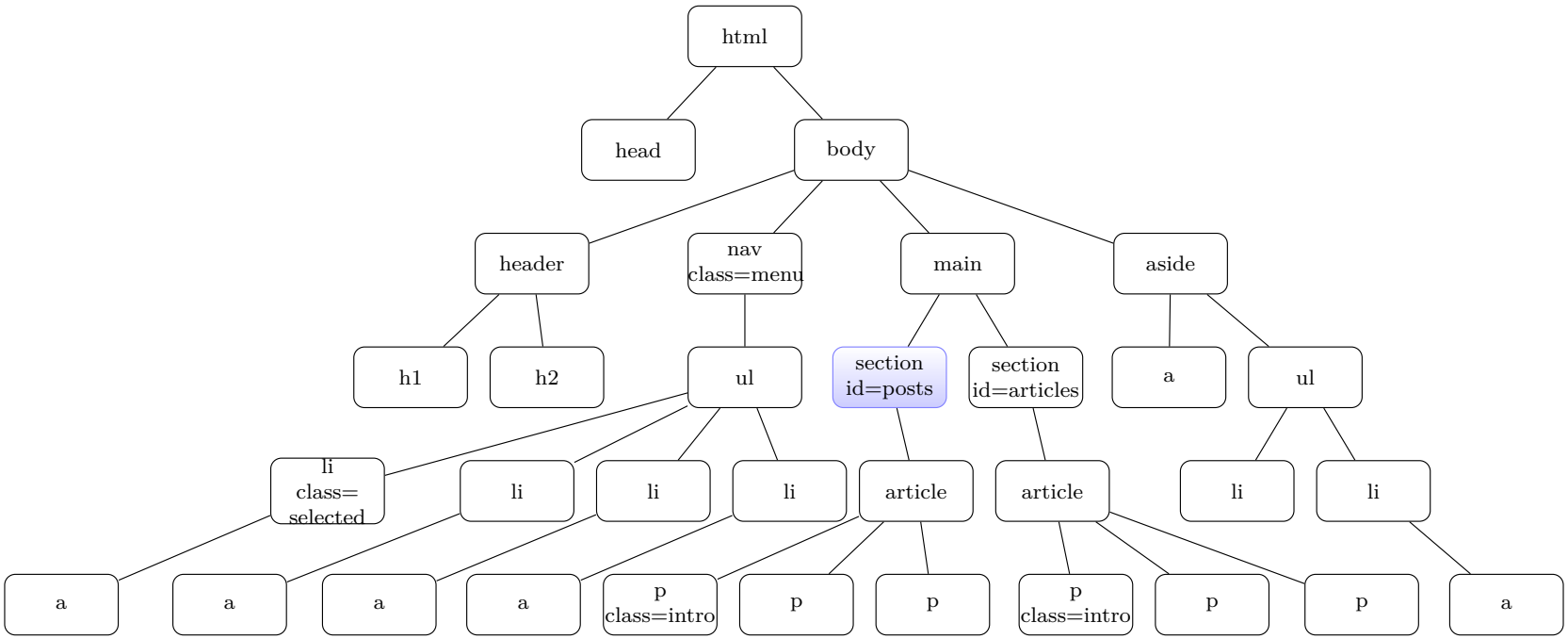
Universal selector: *



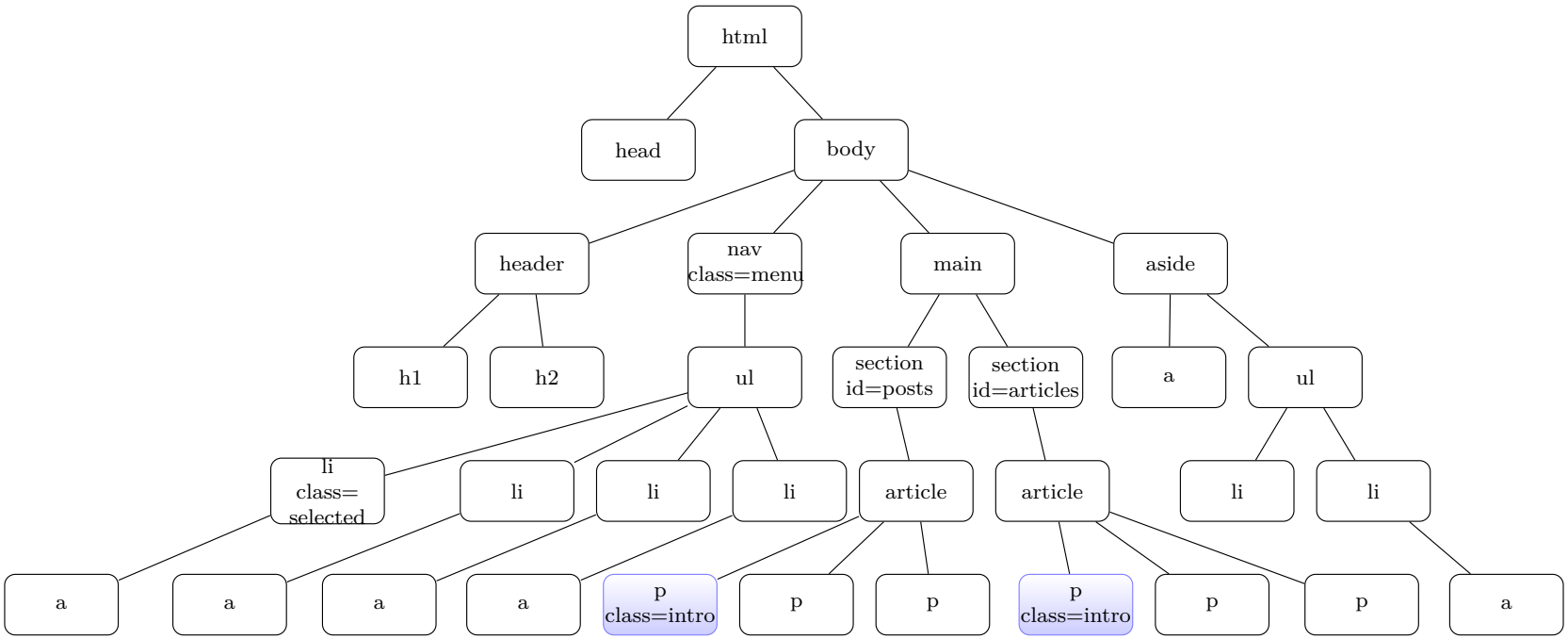
Type selector: a



Id selector: #posts



Class selector: .intro

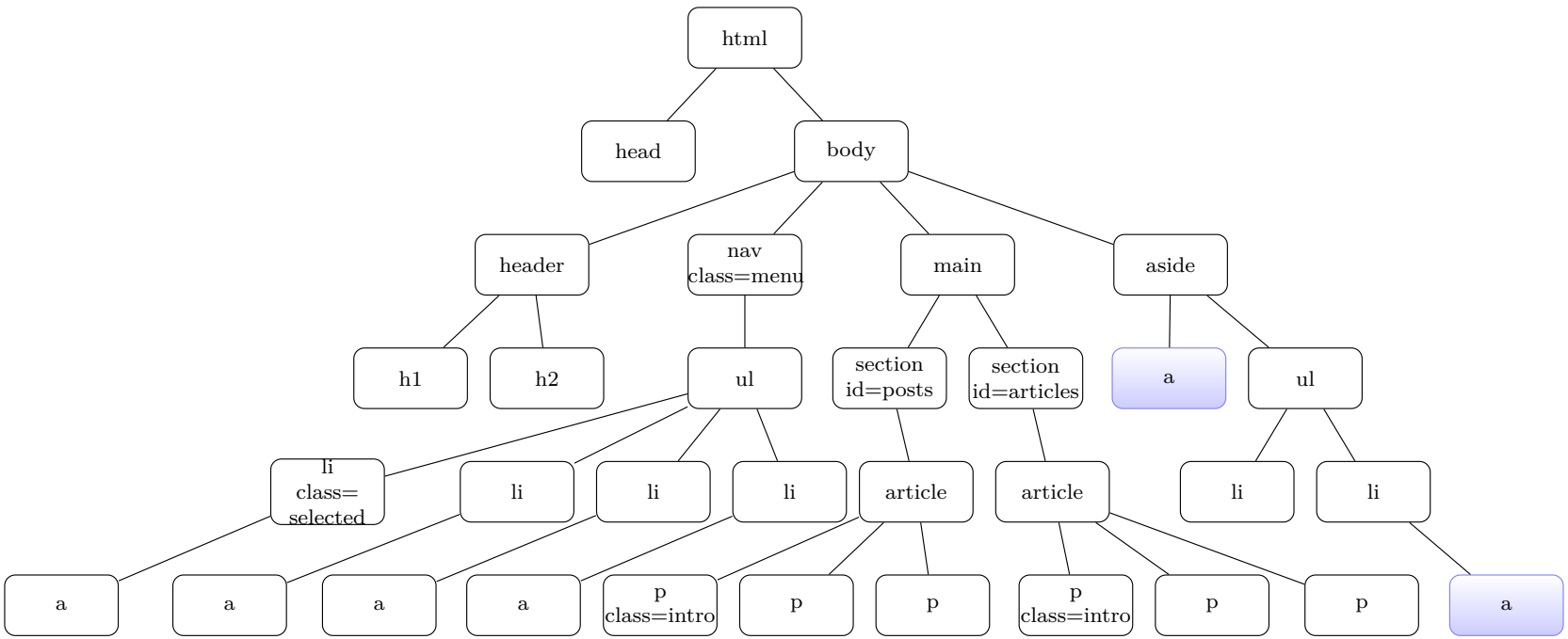


Combinators

Descendant combinator

Select all descendants (space)

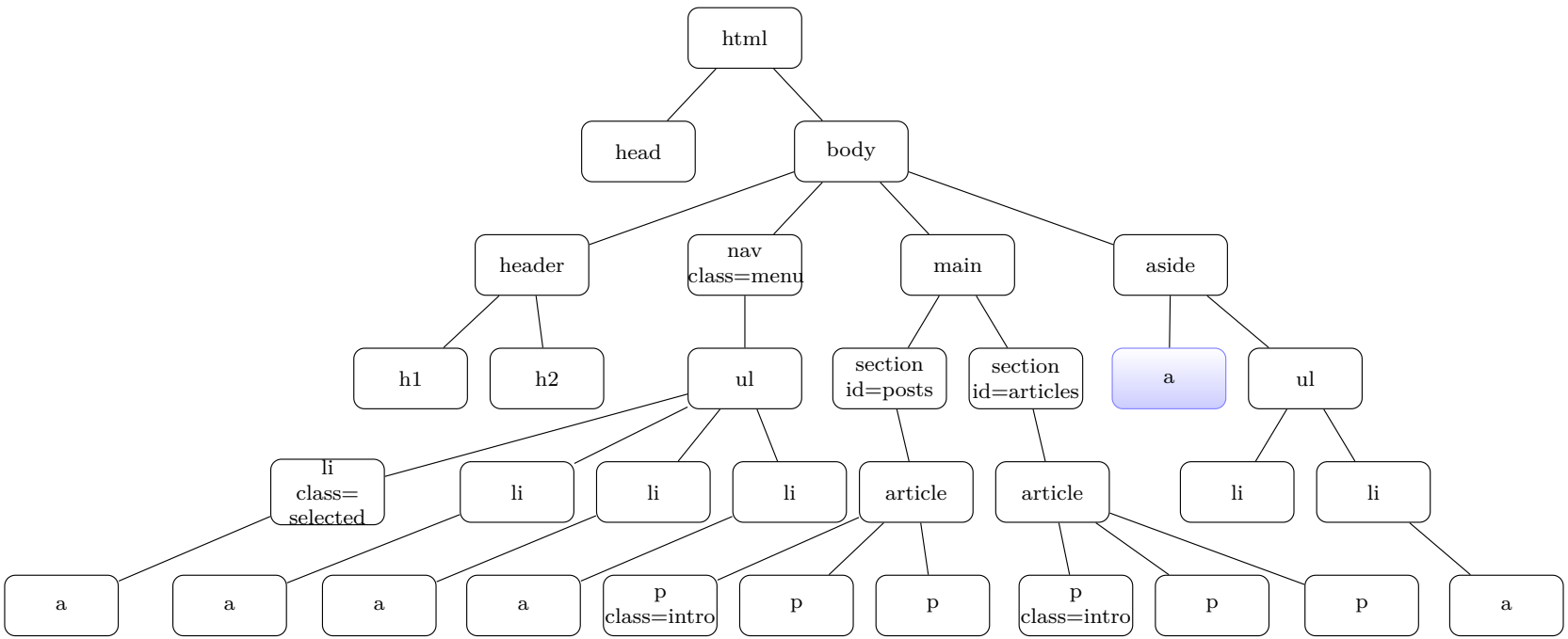
```
aside a
```



Descendant combinator

Select all children. Direct descendants only.

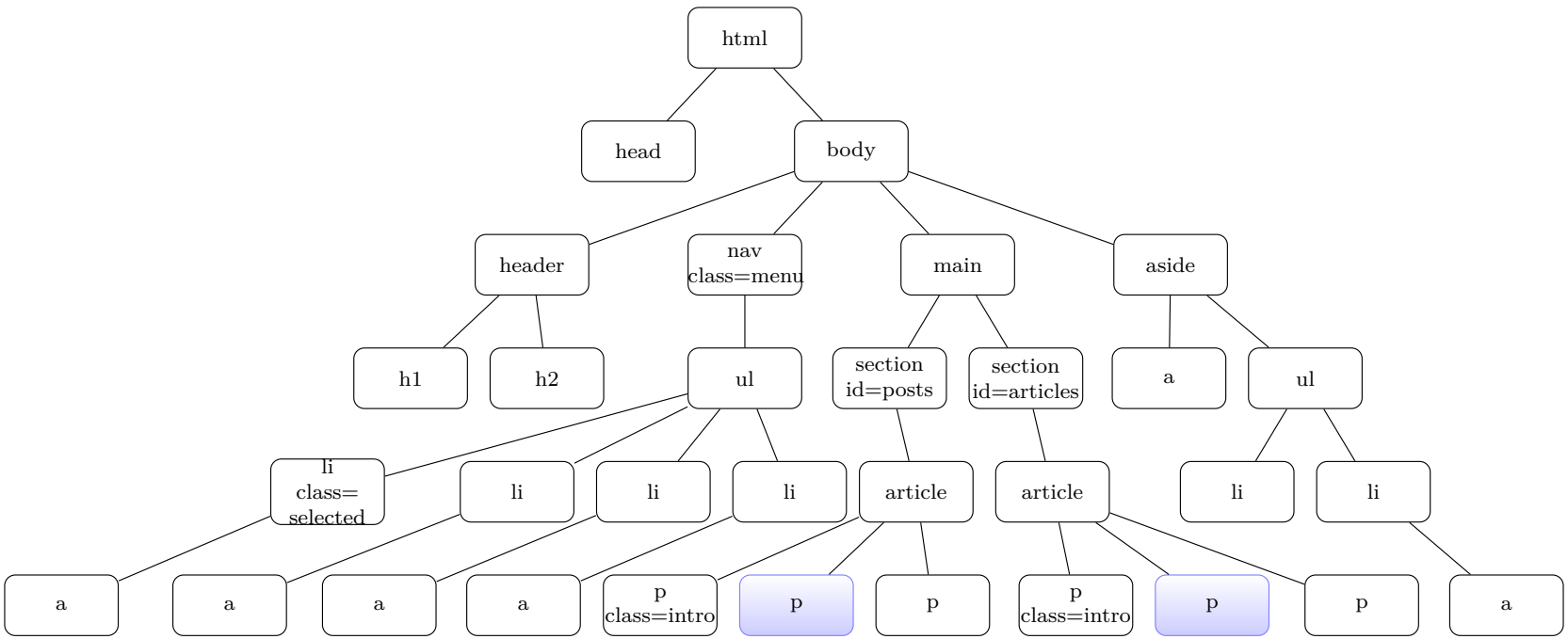
```
aside > a
```



Descendant combinator

Select next sibling (+). The next one similar element.

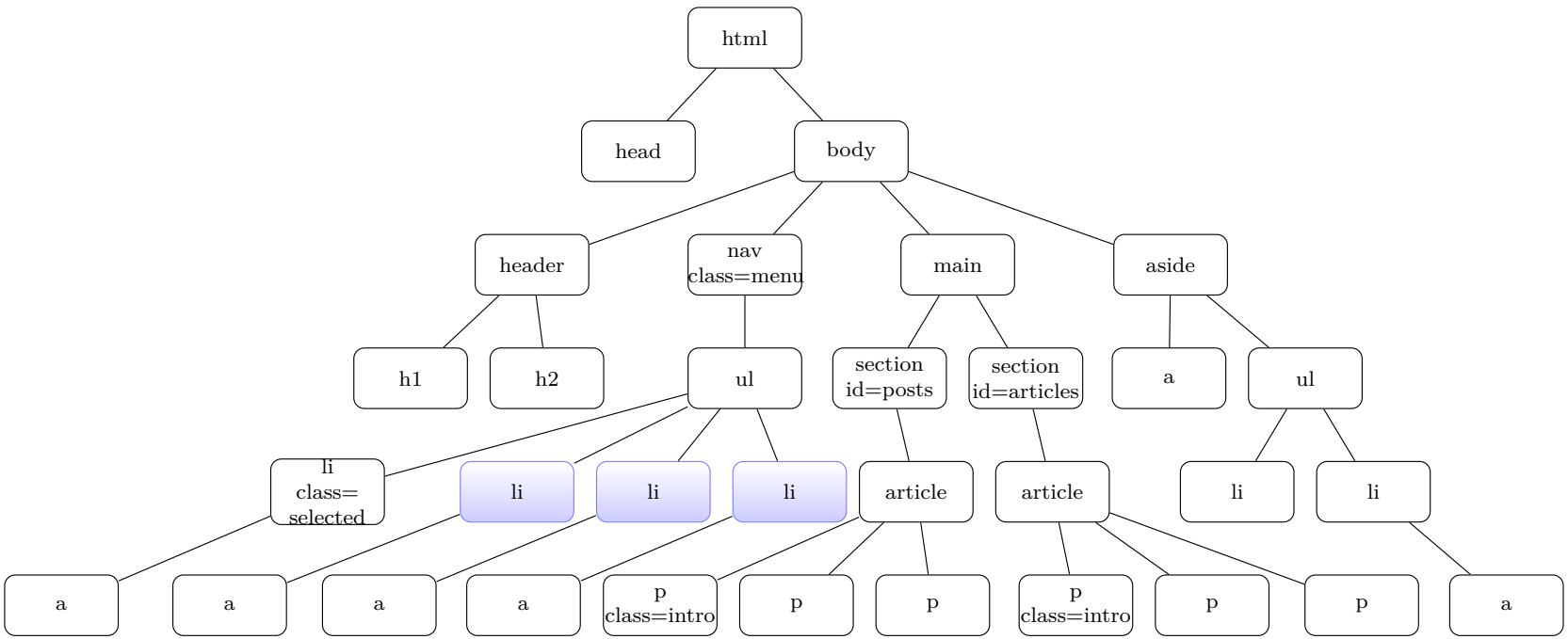
```
.intro + p
```



Descendant combinator

Select subsequent siblings (~).

```
.selected ~ li
```

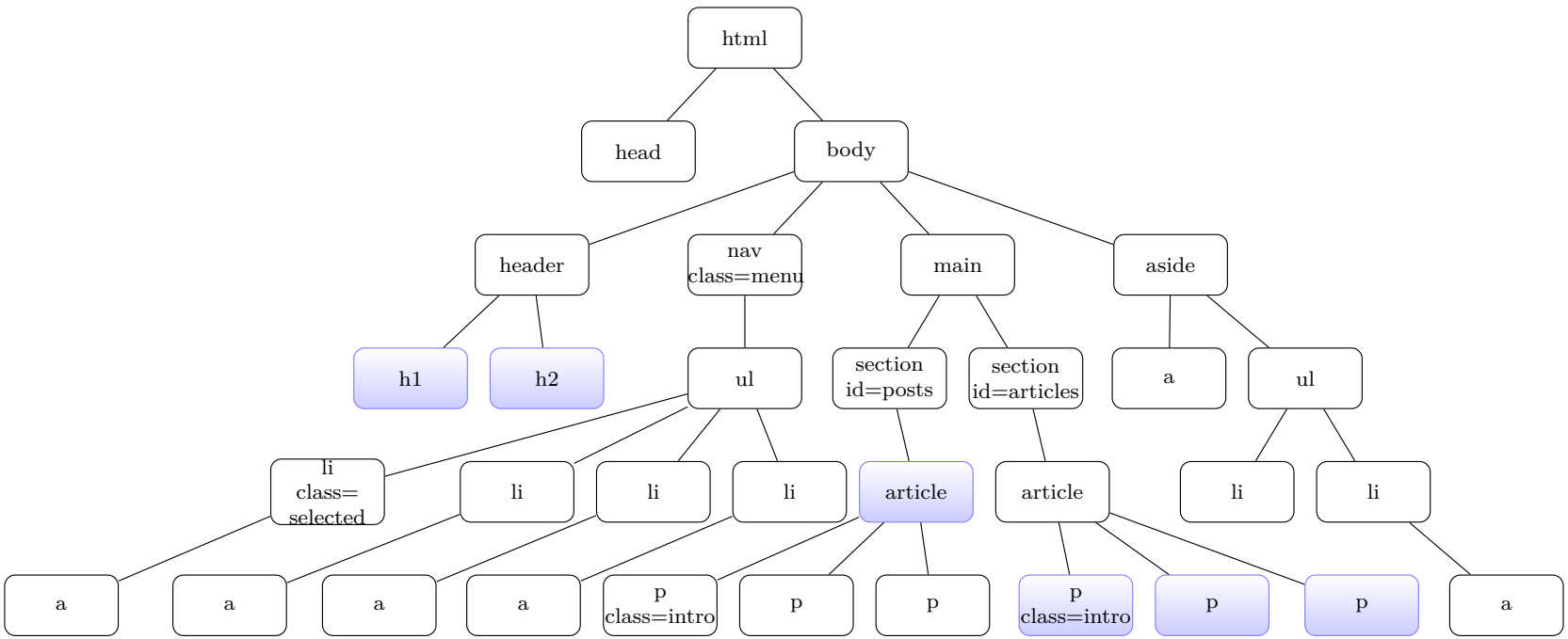


Grouping Selectors

Descendant combinator

Selector groups (,) are just a way to simplify CSS rules

```
header > *, main article, #articles p
```



Exercise: Learning Aid

To verify the that the selectors we have looked at are true as indicated in the slides, build the HTML tree and apply the rules.

Project 2

Find **Project 2** on **vcampus** and work on it.

See you next week, God willing 🙏🏾