FEWD

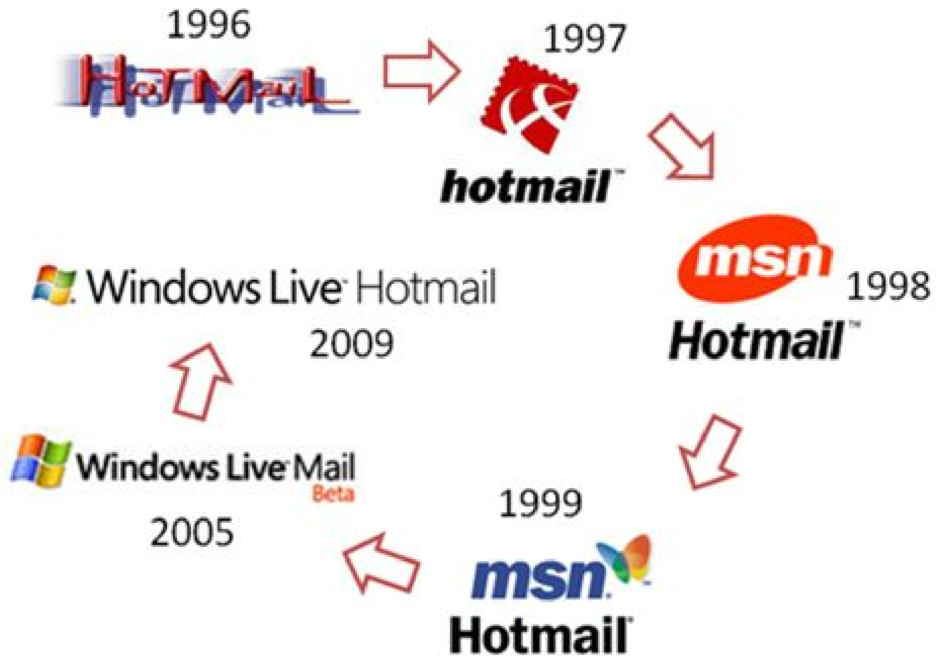
WELCOME TO FRONT-END WEB DEVELOPMENT

Please sit next to a different classmate and write your name on your name tag.

Wi-fi: GA-Guest
pw: yellowpencil

HoTMaiL

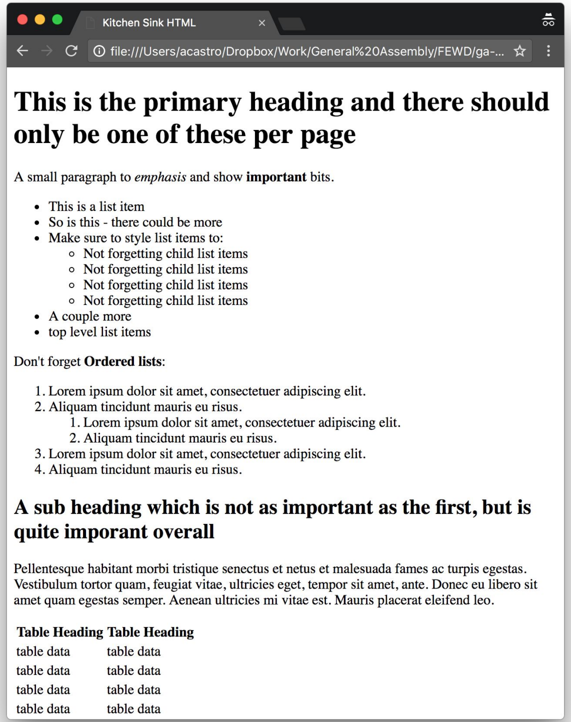
The name “Hotmail” was chosen out of many possibilities ending in *“-mail”* as it included the letters HTML. To emphasize this, the original type casing was “HoTMaiL”.



LESSON 04 BOX MODEL & LAYOUT RECAP

CSS Reset and Normalize

Default rendering



Kitchen Sink HTML

file:///Users/acastro/Dropbox/Work/General%20Assembly/FEWD/ga-...

This is the primary heading and there should only be one of these per page

A small paragraph to *emphasis* and show **important** bits.

- This is a list item
- So is this - there could be more
- Make sure to style list items to:
  - Not forgetting child list items
  - Not forgetting child list items
  - Not forgetting child list items
  - Not forgetting child list items
- A couple more
- top level list items

Don't forget **Ordered lists**:

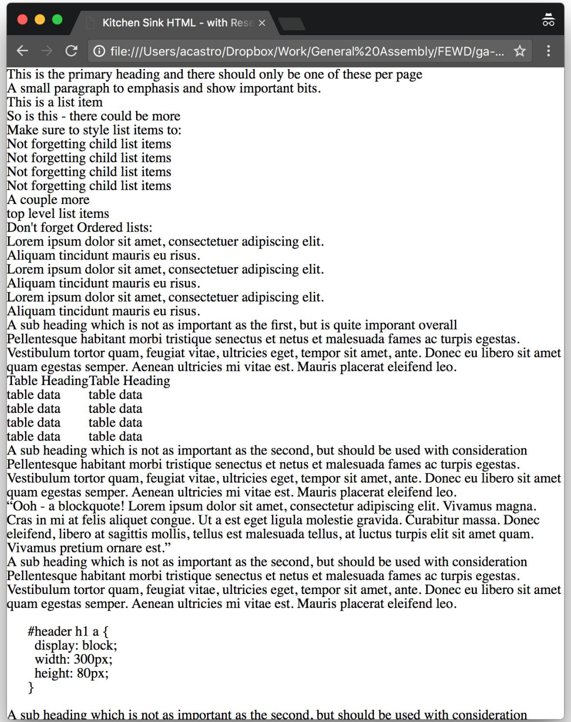
1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
2. Aliquam tincidunt mauris eu risus.
   1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
   2. Aliquam tincidunt mauris eu risus.
3. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
4. Aliquam tincidunt mauris eu risus.

A sub heading which is not as important as the first, but is quite imporant overall

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

Table Heading	Table Heading
table data	table data
table data	table data
table data	table data
table data	table data

With reset.css



Kitchen Sink HTML - with Rese

file:///Users/acastro/Dropbox/Work/General%20Assembly/FEWD/ga-...

This is the primary heading and there should only be one of these per page
A small paragraph to emphasis and show important bits.
This is a list item
So is this - there could be more
Make sure to style list items to:
Not forgetting child list items
Not forgetting child list items
Not forgetting child list items
Not forgetting child list items
A couple more
top level list items
Don't forget Ordered lists:
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Aliquam tincidunt mauris eu risus.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Aliquam tincidunt mauris eu risus.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Aliquam tincidunt mauris eu risus.
A sub heading which is not as important as the first, but is quite imporant overall
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

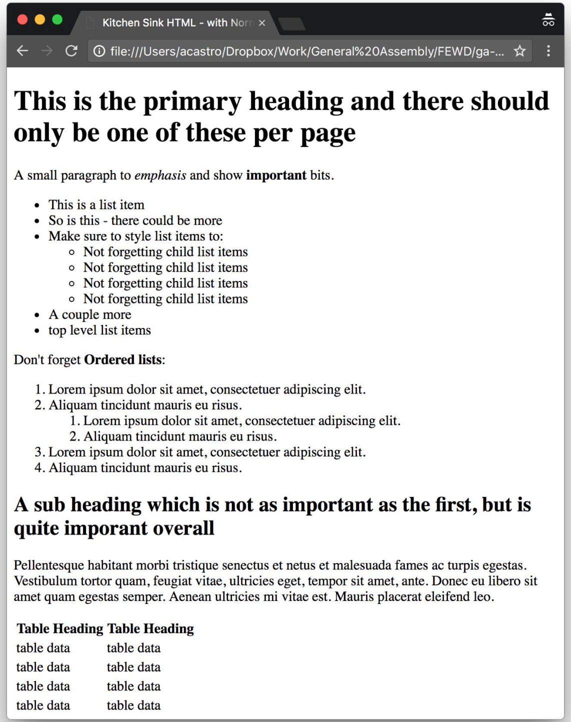
Table Heading	Table Heading
table data	table data
table data	table data
table data	table data
table data	table data

A sub heading which is not as important as the second, but should be used with consideration
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.
“Ooh - a blockquote! Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus magna. Cras in mi at felis aliquet congue. Ut a est eget ligula molestie gravida. Curabitur massa. Donec eleifend, libero at sagittis mollis, tellus est malesuada tellus, at luctus turpis elit sit amet quam. Vivamus pretium ornare est.”
A sub heading which is not as important as the second, but should be used with consideration
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

```
    #header h1 a {
      display: block;
      width: 300px;
      height: 80px;
    }
```

A sub heading which is not as important as the second, but should be used with consideration

With normalize.css



Kitchen Sink HTML - with Norr

file:///Users/acastro/Dropbox/Work/General%20Assembly/FEWD/ga-...

This is the primary heading and there should only be one of these per page

A small paragraph to *emphasis* and show **important** bits.

- This is a list item
- So is this - there could be more
- Make sure to style list items to:
  - Not forgetting child list items
  - Not forgetting child list items
  - Not forgetting child list items
  - Not forgetting child list items
- A couple more
- top level list items

Don't forget **Ordered lists**:

1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
2. Aliquam tincidunt mauris eu risus.
   1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
   2. Aliquam tincidunt mauris eu risus.
3. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
4. Aliquam tincidunt mauris eu risus.

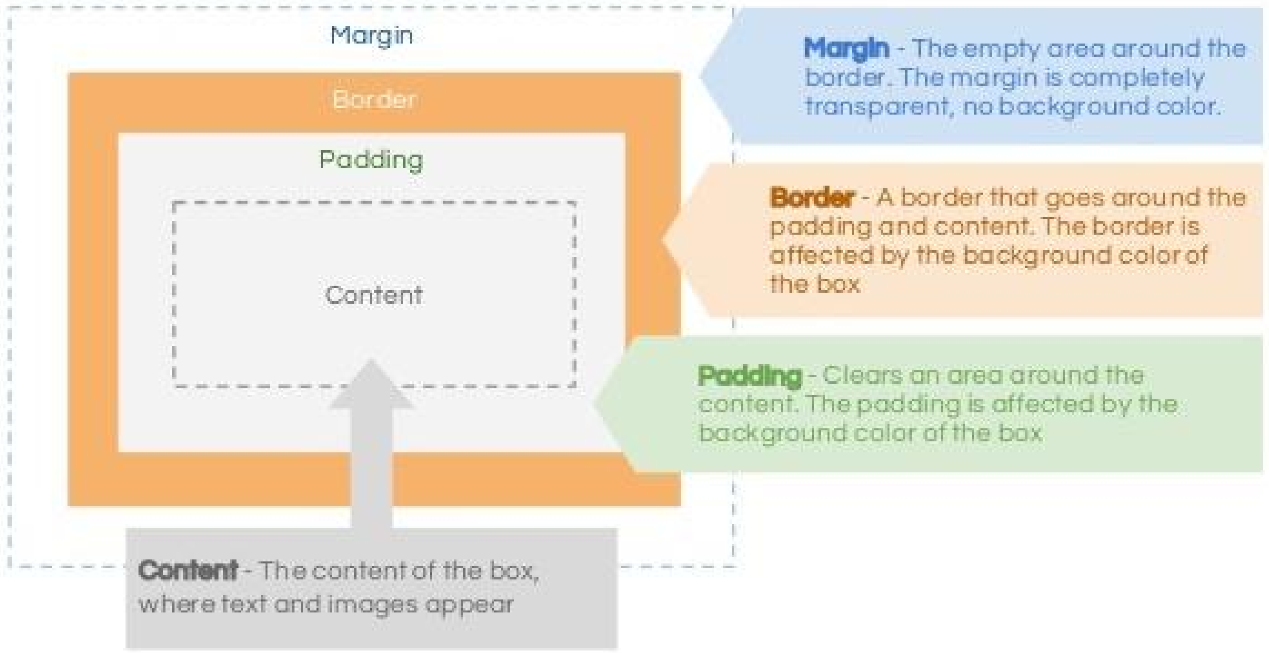
A sub heading which is not as important as the first, but is quite imporant overall

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

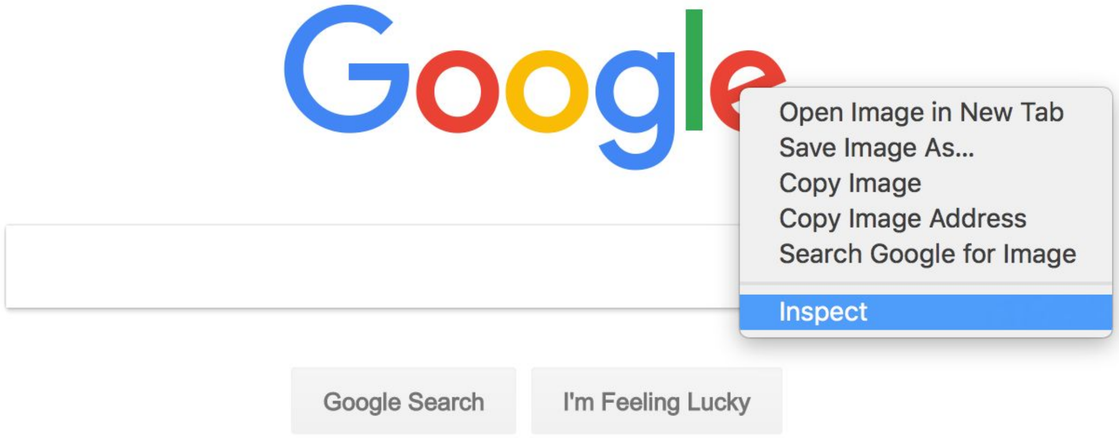
Table Heading	Table Heading
table data	table data
table data	table data
table data	table data
table data	table data

CSS Box Model

In an HTML document, each element is represented as a rectangular box, with the box's content, padding, border, and margin built up around one another **like the layers of an onion**.


Margin
Border
Padding
Content
Margin - The empty area around the border. The margin is completely transparent, no background color.
Border - A border that goes around the padding and content. The border is affected by the background color of the box
Padding - Clears an area around the content. The padding is affected by the background color of the box
Content - The content of the box, where text and images appear

Introducing the Developer Tools' Inspector



Classes & ID

Useful for:

- Classes and IDs are **selectors**
- id is used to define **one unique element**
- Classes can define more than one element

```
/* ID */
#main-content {
  color: black;
}

/* class *
.messages {
  color: red;
}
```

CSS Selectors – Basic

Selector	Description	Example

`element`	**Type** selector. Matches an element.	`p { color: red }`
		`/* matches paragraphs */`

`.class`	**Class** selector. Matches the value of a class attribute.	`.warning { color: red }`
		`/* matches elements containing class="warning" */`

`#id`	**ID** selector. Matches the value of an id attribute.	`#warning { color: red }`
		`/* matches elements containing id="warning" */`

`*`	**Universal** selector. Matches everything.	`* { color: red }`
		`/* matches everything */`

CSS

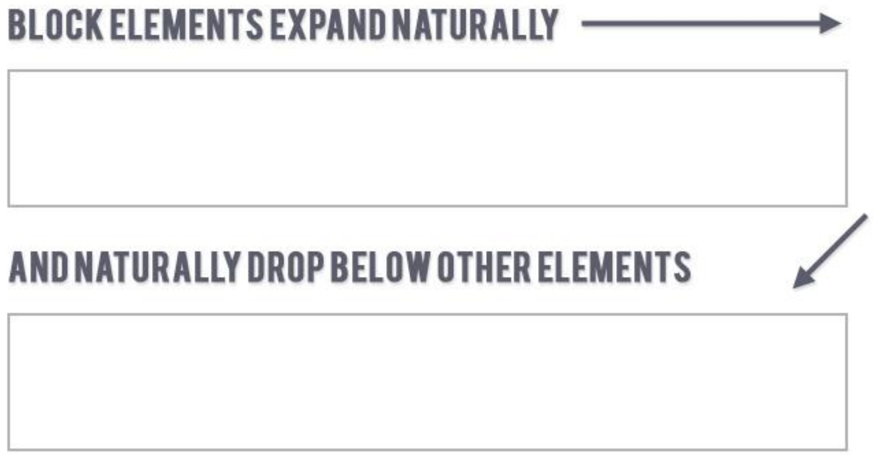
LESSON 05 ADVANCED CSS & LAYOUT LAB

Understanding block-level vs inline elements

Block-level elements:

- If no **width** is set, **will expand naturally** to fill its parent container
- If no **height** is set, **will expand naturally** to fit its child elements
- Can have margins and padding
- By default, will be placed below previous elements in the markup

Examples of block-level elements:
`<p>`, `<div>`, `<form>`, `<header>`, `<nav>`, `<ul>`, `<li>` and `<h1>`


BLOCK ELEMENTS EXPAND NATURALLY
AND NATURALLY DROP BELOW OTHER ELEMENTS

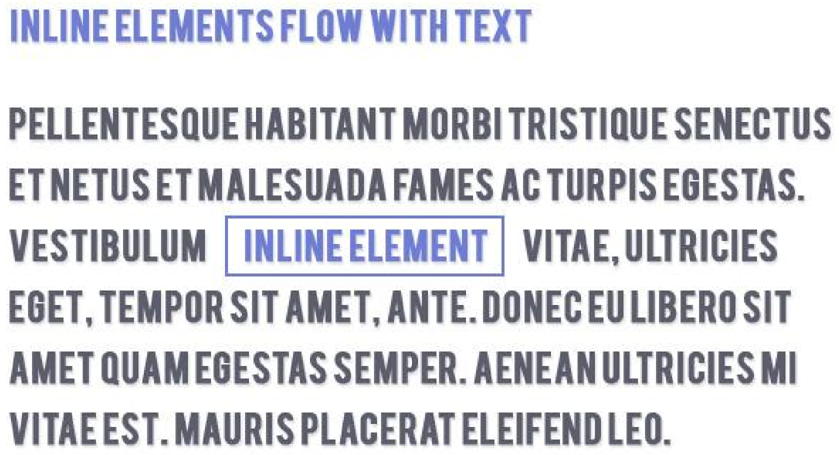
Understanding block-level vs inline elements

Inline elements:

- Flows along with text content
- Will not clear previous content to drop to the next line like block elements
- Will ignore top and bottom margin settings, but will apply left and right margins, and any padding
- Will ignore the width and height properties

Examples of inline elements:

`<a>`, `<span>`, `<b>`, `<em>`, `<i>`, `<cite>`, `<mark>` and `<code>`


INLINE ELEMENTS FLOW WITH TEXT

PELLENTESQUE HABITANT MORBI TRISTIQUE SENECTUS ET NETUS ET MALESUADA FAMES AC TURPIS EGESTAS. VESTIBULUM INLINE ELEMENT VITAE, ULTRICIES EGET, TEMPOR SIT AMET, ANTE. DONEC EU LIBERO SIT AMET QUAM EGESTAS SEMPER. AENEAN ULTRICIES MI VITAE EST. MAURIS PLACERAT ELEIFEND LEO.

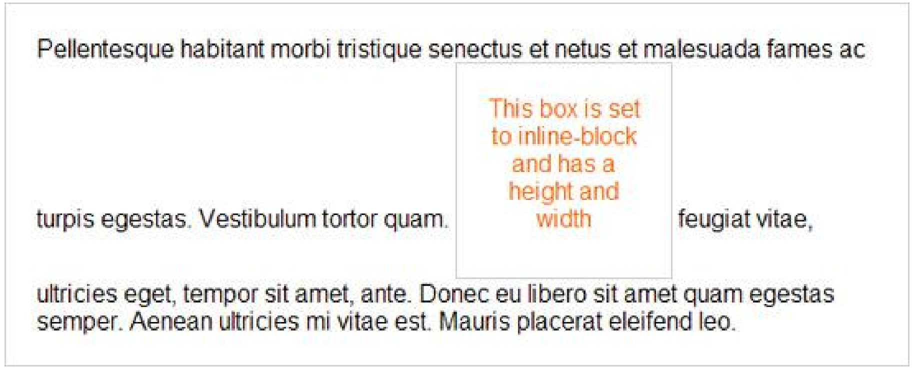
Introducing inline-block

Inline-block elements:

- Allow other elements to sit to their left and right
- Can have margins and padding
- Can have explicit height and width

Inline-block level elements are defined:

```
element {
  Display: inline-block;
}
```



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam. This box is set to inline-block and has a height and width feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

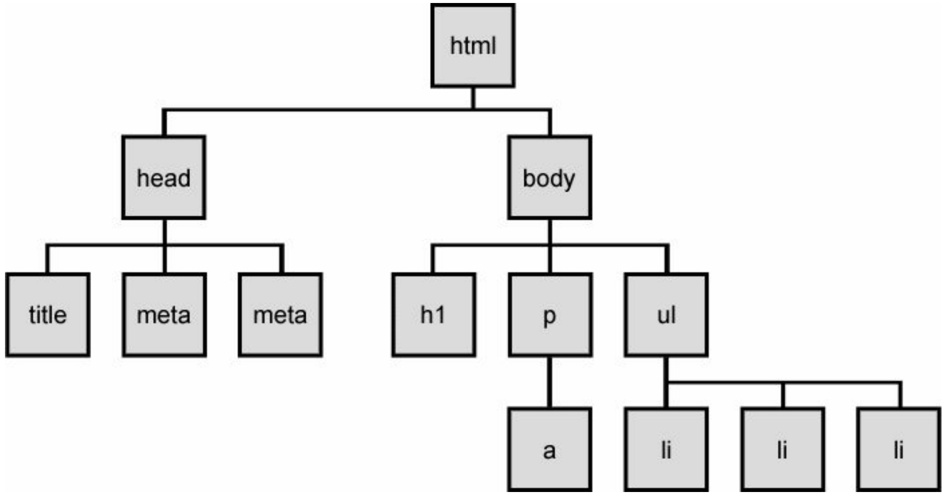
The DOM Tree

The DOM tree is a **structural representation of a web page**.

The **root** of every tree is the `html` element.

The following relationships exist among elements in the tree:

- Parent
- Child
- Sibling
- Ancestor
- Descendant



The DOM Tree

Parent: the element connected above another element

- `body` is the parent of `h1, p` and `ul`

Child: the element connected below another element

- `li` is a child of `ul`

Sibling: elements with the same parent

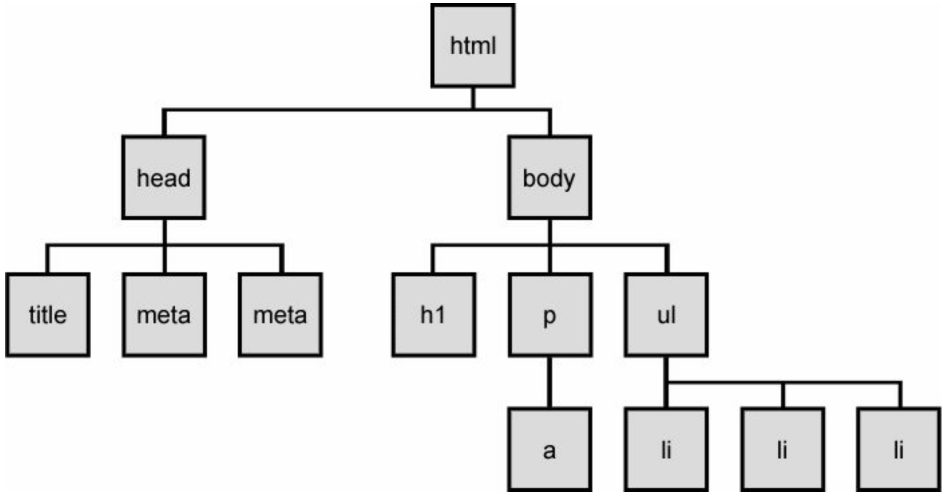
- `h1`, `p` and `ul` are siblings

Ancestor: Any element that precedes a given element

- `body` is an ancestor of `h1` and `p` but also `a` and `li`

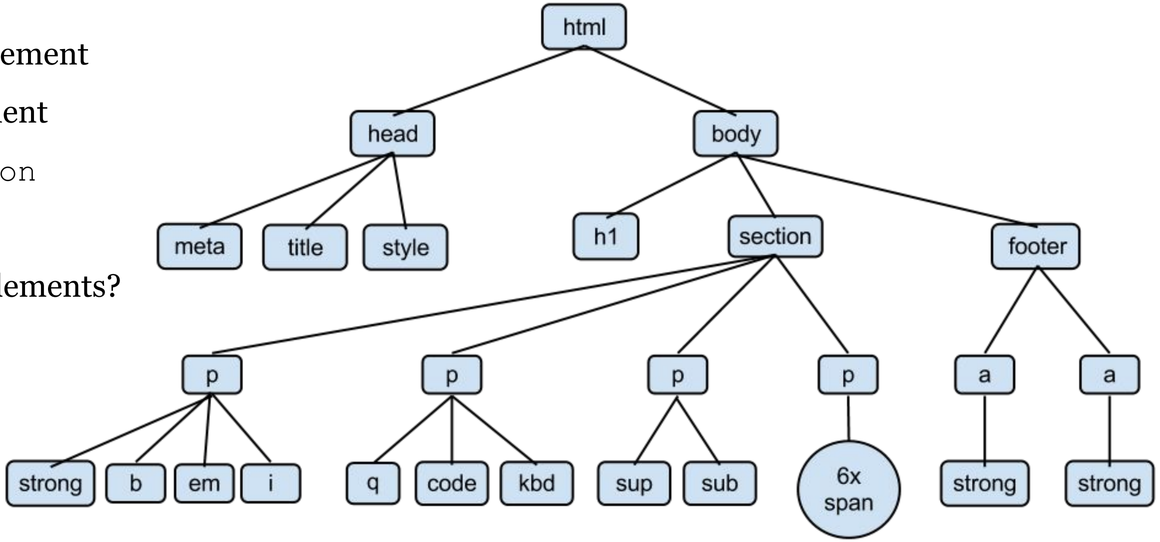
Descendant: Any given element that has another element as an ancestor

- `li` is an descedant of `body`



Playing in the DOM Tree

1. Name all the **ancestors** of the `kbd` element
2. Name all the **descendants** of the `footer` element
3. Name all the **siblings** of the `code` element
4. Name the **parent** of the `title` element
5. Name all the **children** of the `section` element
6. Which element has the most **child** elements?



Sketching a DOM Tree

Sketch out DOM tree for a document that has elements that satisfy each of the following relationships:

- An `html` element at the root of the element tree
- The `head` has the required `meta` and `title` elements as children and a `style` child element for CSS
- The `body` has a `header` element, two `section` elements, and a `footer` element, in that order, as children
- The `header` has an `h1` element followed by a `nav` element as children
- The `nav` element has five `a` elements as children
- Both `section` elements have the same children: an `h2` element followed an `h3` element followed by an `ol` element followed by another `h3` element and then another `ol` element
- All the `ol` elements have four `li` elements as children
- The `footer` has two `a` elements as children, and each of the `a` elements has a single `strong` element as a child

CSS Selectors – Basic

Selector	Description	Example
`element`	**Type** selector. Matches an element.	`p { color: red }` `/* matches paragraphs */`
`.class`	**Class** selector. Matches the value of a class attribute.	`.warning { color: red }` `/* matches elements containing class="warning" */`
`#id`	**ID** selector. Matches the value of an id attribute.	`#warning { color: red }` `/* matches elements containing id="warning" */`
`*`	**Universal** selector. Matches everything.	`* { color: red }` `/* matches everything */`

CSS Selectors – Basic (continued)

Selector	Description	Example
`element, element`	Selector **grouping**. Matches multiple elements separated by a comma	`h1, h2 { color: red }` `/* matches both h1 and h2 */`
`element.class`	**Combined Class** selector. Matches a specific element with the value of a class attribute	`p.warning { color: yellow }` `/* matches only paragraphs containing class="warning" */` `p.warning.urgent { color: red }` `/* matches only paragraphs containing class="warning urgent" */`

CSS Selectors – Combinators

Selector	Description	Example
`selector selector`	**Descendant** combinator. Matches elements that are descendants of another element.	`section p { color: red }` `/* matches a paragraph inside a section element */`
`selector > selector`	**Child** combinator. Matches elements that are children of another element.	`.warning > p { color: red }` `/* matches paragraphs that are children of elements containing class="warning" */`
`selector + selector`	**Adjacent sibling** combinator. Matches elements that immediately follow another element.	`h1 + p { color: red }` `/* matches the first paragraph to follow a top-level heading */`
`selector ~ selector`	**General sibling** combinator. Matches elements that follow another element.	`h2 ~ p { color: red }` `/* matches every paragraph that follows a second-level heading */`