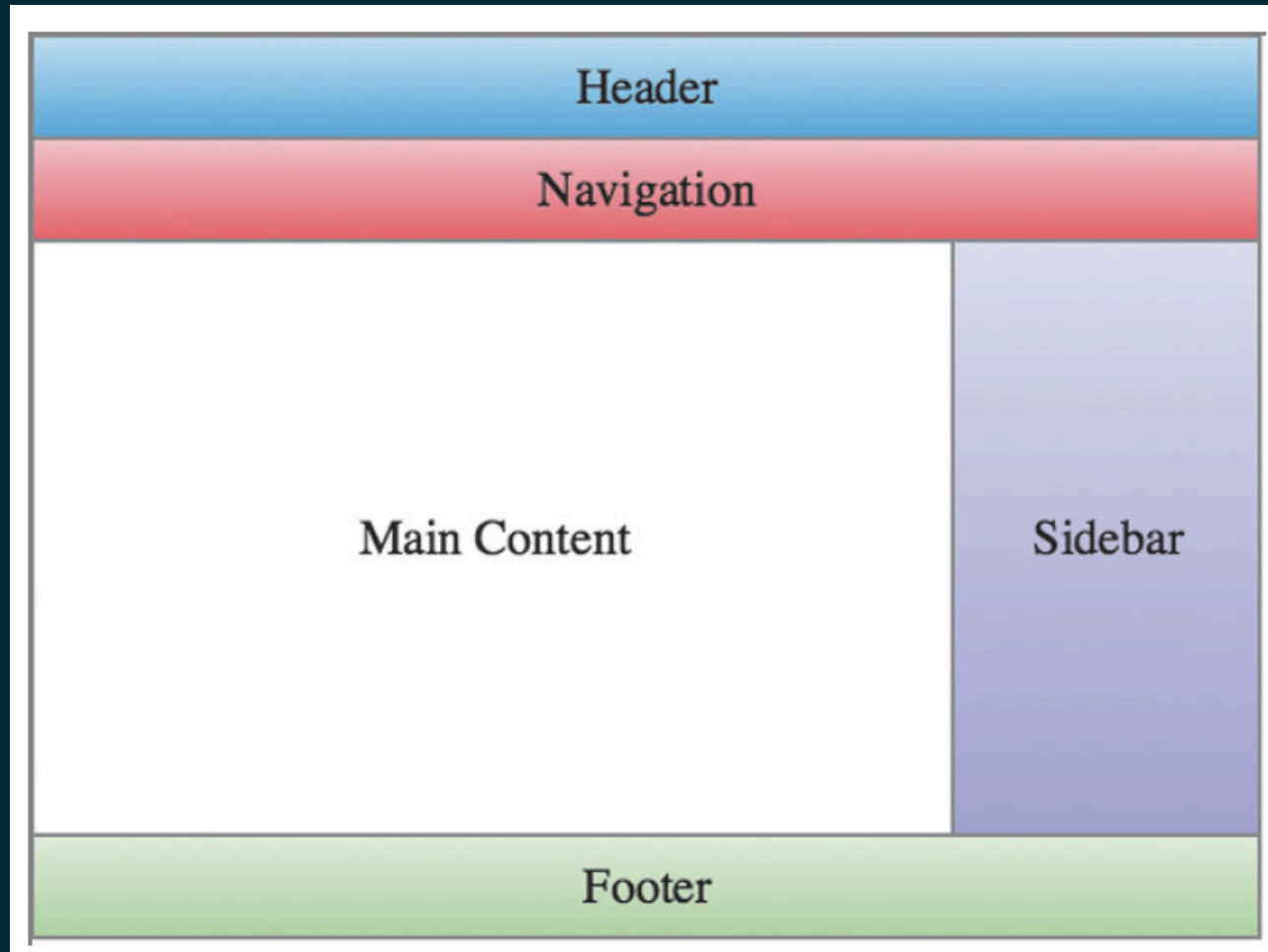# LEARN WEB

ale66

PAGE LAYOUTS

SO FAR

learned how to combine `HTML5+CSS` to obtain pleasing *static* pages.

However, the default HTML layout is almost never a good idea...

- blocks and padding help in separating text and make it stand out, but

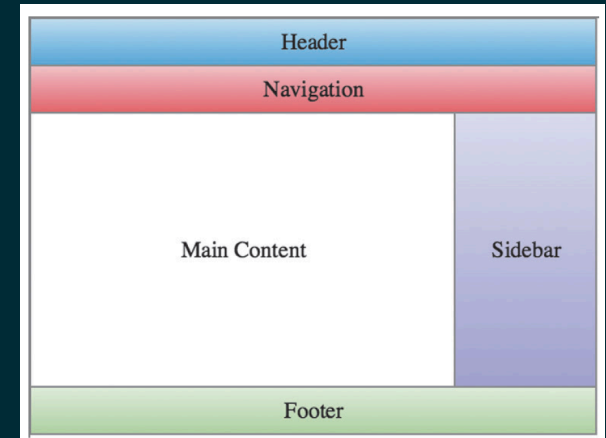- the CSS style may result in displaying over two lines

A MODERN, DEFAULT LAYOUT

A review of Ch. 11 of the textbook

github.com/ale66/learn-web

A MODERN, DEFAULT LAYOUT, 2

The Holy-Grail Layout

To help you learn the various web page layout techniques, I'm going to use a version of the so-called *holy-grail* layout that consists of the following parts:

- A header at the top of the page

- A navigation bar below the header

- Two full-height columns consisting of the main page content in the left column and a sidebar of related content in the right (or sometimes the left) column

- A footer at the bottom of the page

APPROACHES TO WEB PAGE LAYOUT

1. do nothing

2. floats

3. inside blocks

4. flexbox

5. Twitter bootstrap

github.com/ale66/learn-web

BEWARE

Another common page layout technology is a front-end library named Bootstrap (https://getbootstrap.com). The library comes with prefab HTML, CSS, and JavaScript components that enable you to get your projects off the ground quickly. Most modern web designers, however, eschew complex solutions such as Bootstrap in favor of writing their web page code themselves.

We are going to do just that...

HEADER

```
<body>
    <header>
        <img src="logo.tif" alt="Semantics Depot logo">
        <h1>Semantics Depot</h1>
    </header>
    etc.
</body>
```

github.com/ale66/learn-web

NAV

```
<body>
    <header>
        <img src="logo.tif" alt="Semantics Depot logo">
        <h1>Semantics Depot</h1>
    </header>
    <nav>
        <a href="#">Home</a>
        <a href="#">Blog</a>
        <a href="#">Contact</a>
        <a href="#">About Us</a>
    </nav>
    etc.
</body>
```
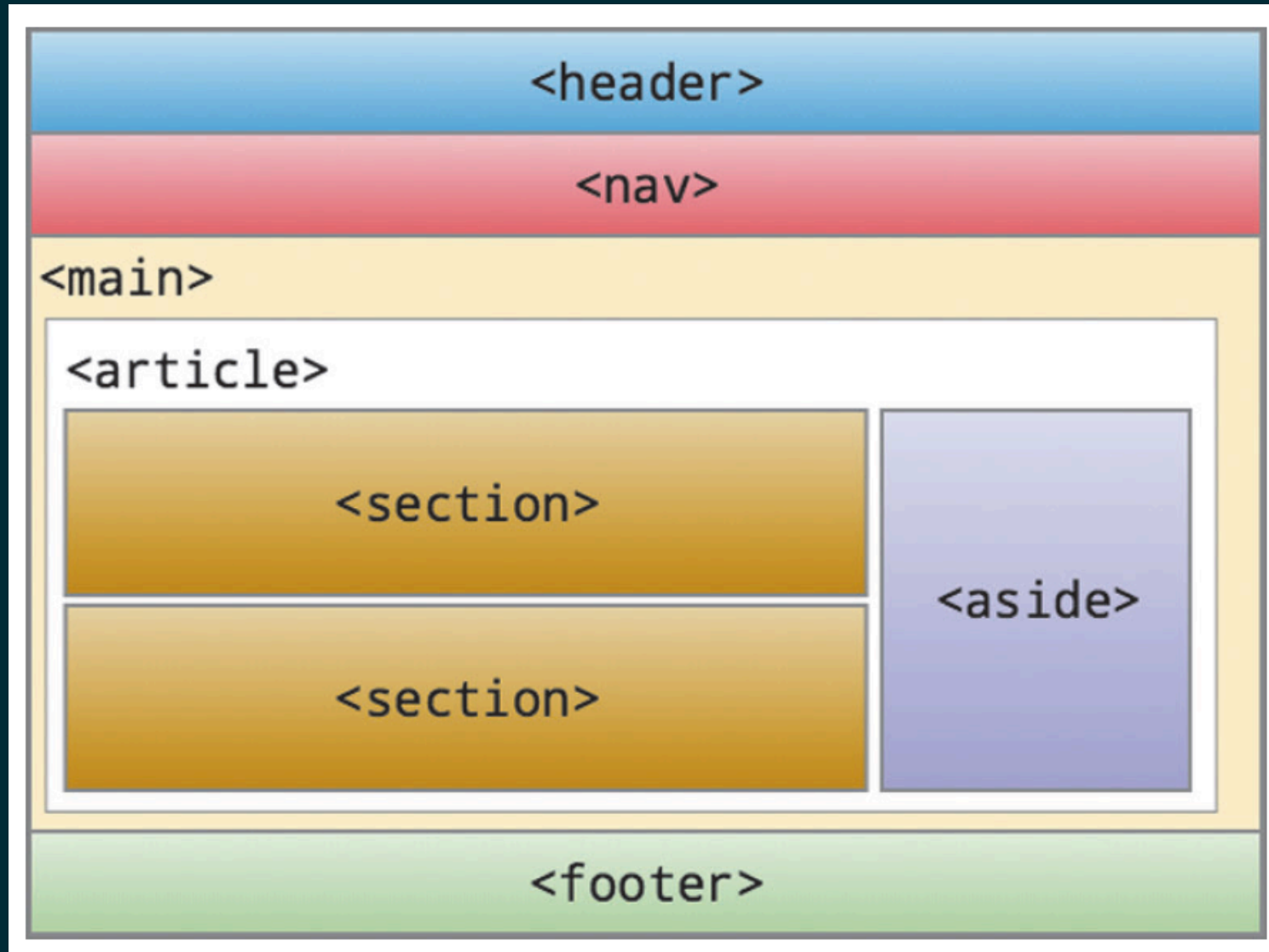
A NEW CONCEPT: ARTICLE

<article>

The `article` element is used to mark up a complete, self-contained composition. The model here is the newspaper or magazine article, but this element can also apply to a blog entry, a forum post, or an essay. Most pages have a single `article` element nested within the `main` element:

```
<body>
    <header>
        ...
    </header>
    <nav>
        ...
    </nav>
    <main>
        <article>
            Article content goes here
        </article>
    </main>
    etc.
</body>
```

It's perfectly acceptable, however, to have multiple `article` elements within a single `main` element. Note, too, that it's okay to nest a `header` element inside an `article` element if doing so is semantically appropriate:

```
<article>
    <header>
        <h2>Isn't It Semantic?</h2>
        <p>By Paul McFedries</p>
    </header>
    Article content goes here
</article>
```

github.com/ale66/learn-web

A NEW, CONTEMPORARY LAYOUT

A FLOATING HEADER

WEB PAGE

YOUR LOGO HERE **Site Title**

CSS

```css
header {
    border: 1px solid black;
    padding: .25em 0;
}
header img {
    float: left;
    padding-left: 1em;
}
h1 {
    float: left;
    padding-left: .5em;
}
.self-clear::after {
    content: "";
    display: block;
    clear: both;
}
```

Float the `img` element to the left.

Float the `h1` element to the left.

Clearfix to prevent the header from collapsing.

HTML

```html
<header class="self-clear">
    <img src="/images/your-logo-here.tif" alt="Our logo">
    <h1>Site Title</h1>
</header>
```

github.com/ale66/learn-web

A FLOATING FOOTER

WEB PAGE

YOUR LOGO HERE **Site Title**

HOME   ITEM   ITEM   ITEM

Article Title
Article paragraph 1
Article paragraph 2

Sidebar Title
Sidebar paragraph

Footer paragraph

CSS

No clear is needed with inline blocks.

```
footer {
    padding: 1em;
    background-color: #b6d7a8;
}
```

HTML

```
<footer>
    <p>Footer paragaph</p>
</footer>
```

SUMMARY

Summary

- You can make your pages more semantic by using the HTML5 page layout tags: `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<aside>`, and `<footer>`.

- To use a float-based layout, let the elements flow in the default manner; then, when you need two or more elements to appear beside each other, float them (usually to the left). Remember that when you want to resume the default page flow, clear the floats.

- To use an inline block-based layout, let the page elements flow in the default manner; then, when you need two or more elements to appear beside each other, display them as inline blocks.

github.com/ale66/learn-web

EXERCISES

- floats: wdpg.io/11-1-0

- inline blocks: wdpg.io/11-2-0

LAB

Participate to the in-class project by developing content and styling one or more *articles*