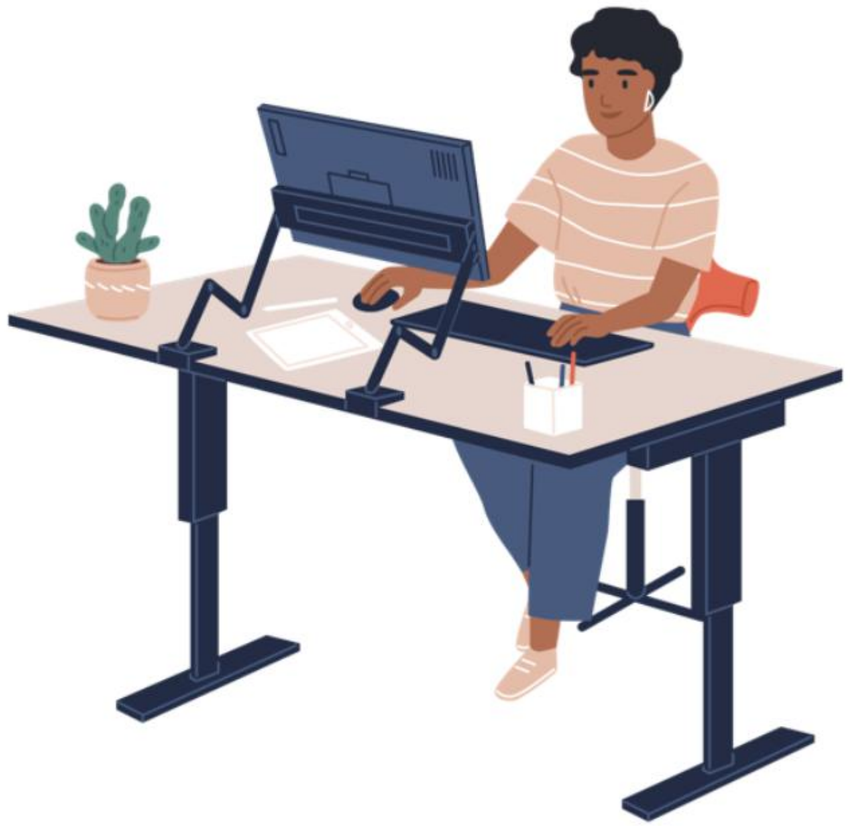
Slide Note

Learning Consolidation

Structure a Web Page Using Semantic HTML5 Elements



Slide Note

In this sprint, you learned to:

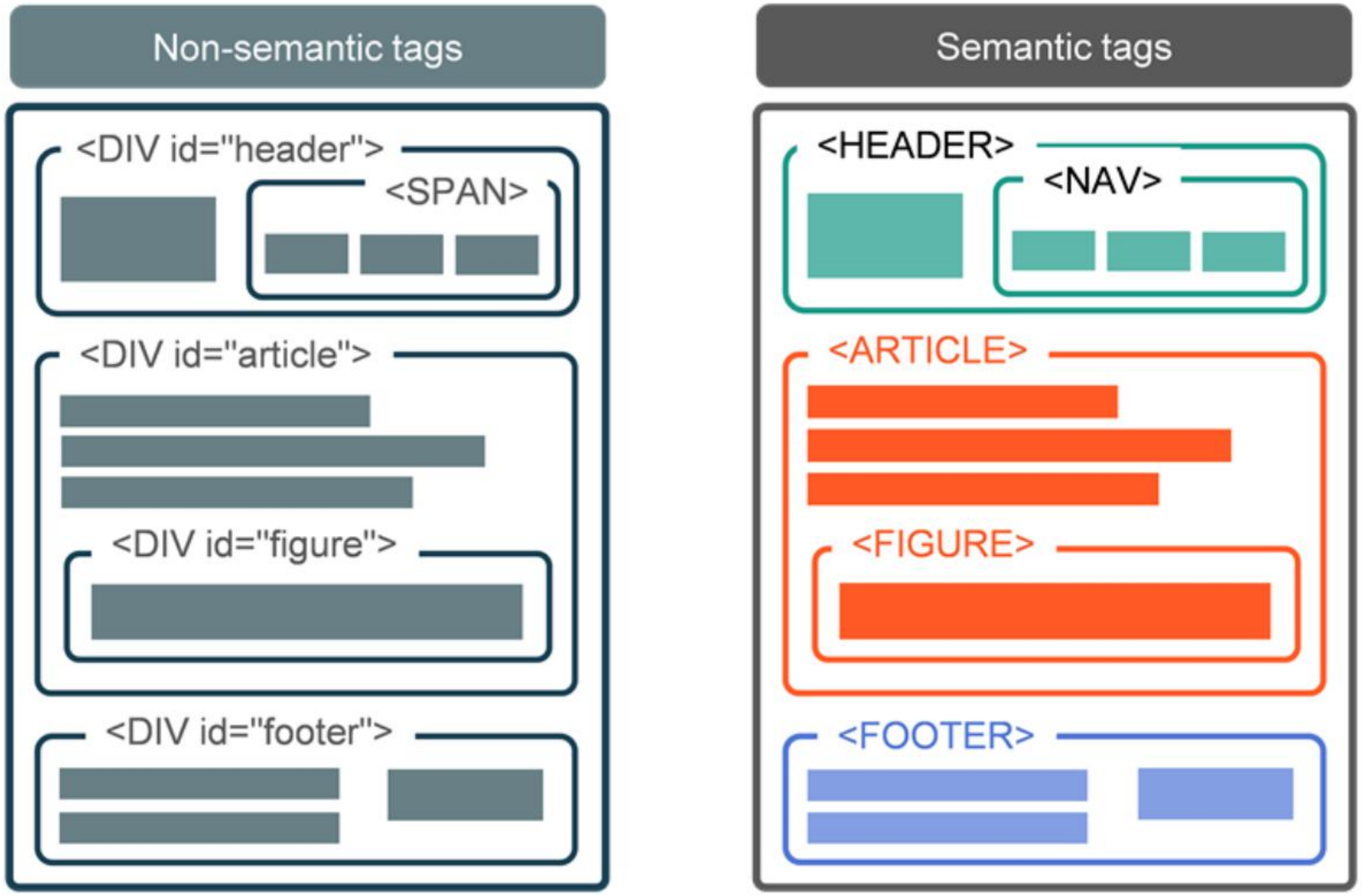
- Differentiate between non-semantic and semantic elements
- Identify the benefits of using semantic elements
- Explain the use of semantic markup by screen readers for page navigation
- Structure a web page using semantic HTML elements

Slide Note

Structure of the document will remain the same whether we use semantic or non-sematic tags like the image seen above.

You can write semantic markup by selecting and using HTML tags properly, and by selecting tags that convey something about the information marked by the tags.

Non-semantic elements like
and don't tell the browser anything about the meanings of the content of the element.



Slide Note

Semantic Tags: Before and After

- <h1>
- <h2>
- <title>
- <p>
-
-

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <time>

Slide Note

• Finding blocks of meaningful code is significantly easier than searching through endless divs with or without semantic or name spaced classes.

• Search engines will consider web page contents to identify important keywords and influence the page's search rankings.

• Screen readers use semantic elements as a signpost to help visually-challenged users navigate a page.

Screen Readers:

A screen reader is a piece of assistive technology that is frequently used by person with visual impairments or learning disabilities. It is also helpful for people learning English (or another language) and for the elderly.

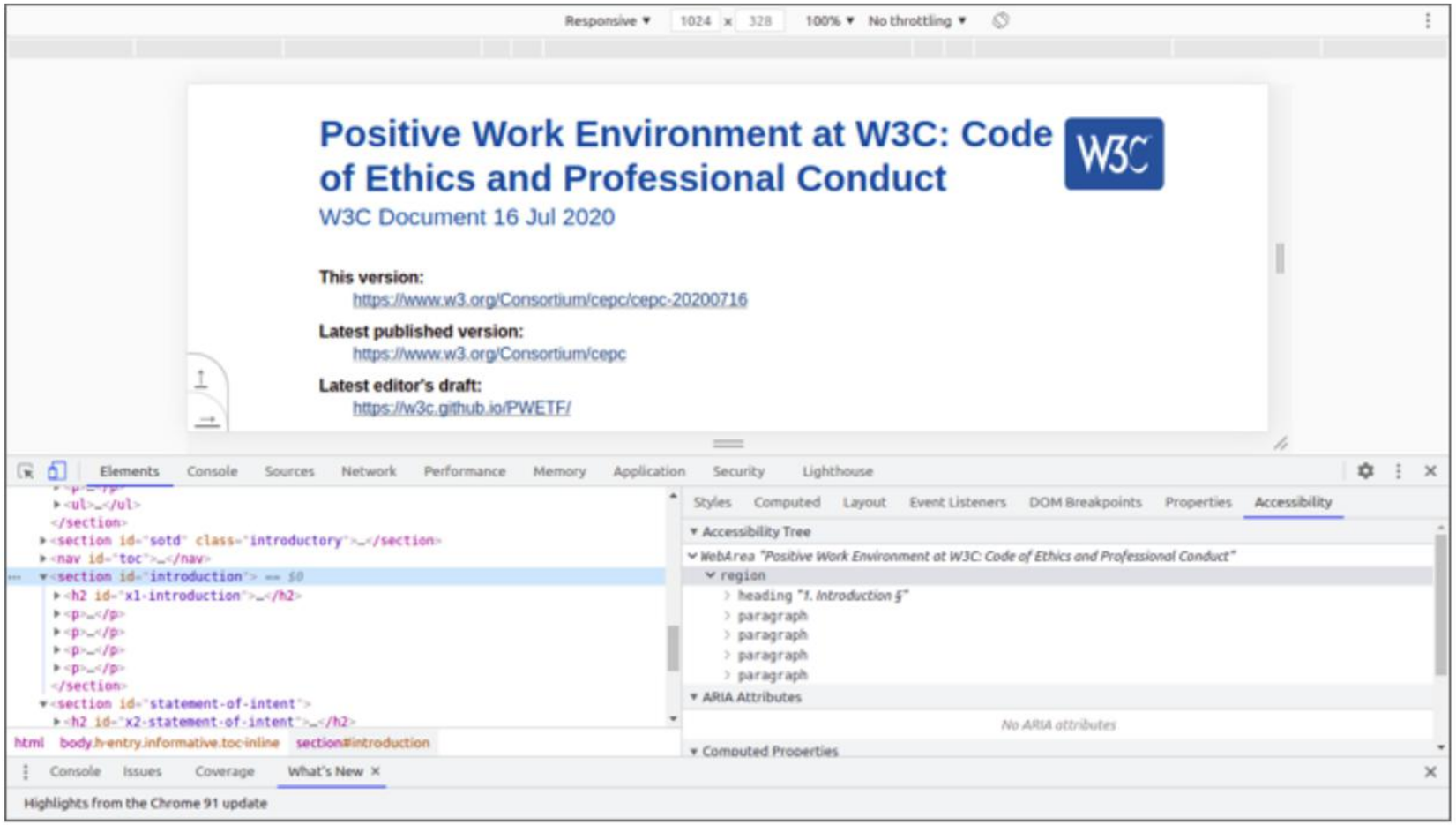
Typically, a screen reader will start at the top of a website or document and read any text (including alternate text for images). Some screen readers allow the user to preview information, like the navigation bar or all the headings on a page and skip the user to the desired section of the page. For this reason, using navigation styles like headings is part of creating accessible documents.

Benefits of Semantic Elements: SEO and Screen Reader

- Semantic elements make the process of searching keywords on web pages easier and faster.
- These elements are used for Search Engine Optimization (SEO) and screen readers.
- Semantic elements are:
  - Easier to read
  - Provide greater accessibility
  - Support default keyboard actions

Slide Note

Accessibility Check in Chrome Dev Tool



Slide Note

• **Easier to read** – The easier it is to read and understand the code, the easier it makes your job, especially when there are thousands of lines of code.

• **Greater accessibility** – Search engines and assistive technologies can better understand the context and content of the website, giving a better user experience.

• **Keyboard support** – Semantic elements provide default keyboard support.

• **Consistent code** – For non-semantic elements, there are many ways to create a header element by giving it various class names. Creating a standard semantic element can make it simpler overall.

• **Better on mobile** – Semantic HTML is arguably lighter in file size than non-semantic spaghetti code as well as more responsive.

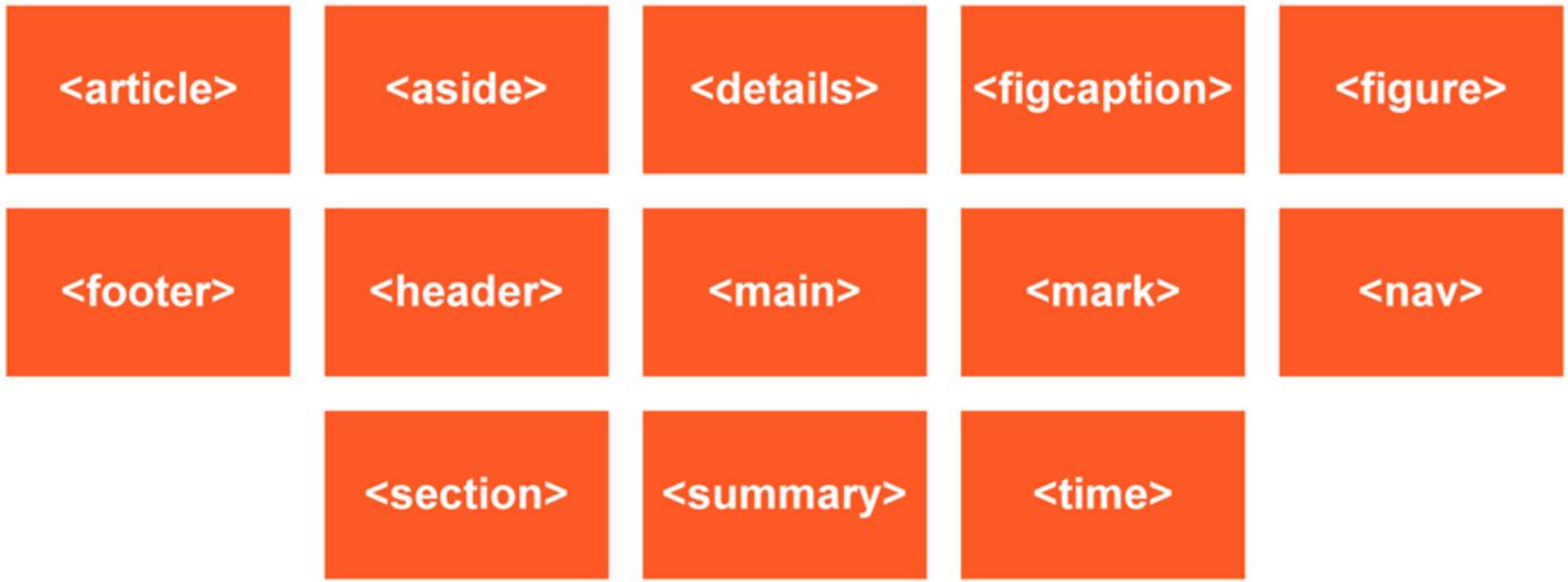
These elements suggest to the developer the type of data that will be populated.

Benefits of Semantic Elements: Web Development

- Semantic elements:
  - Ensure consistent code
  - Are lighter in file size than the non-semantic spaghetti code and easier to make responsive for mobile devices

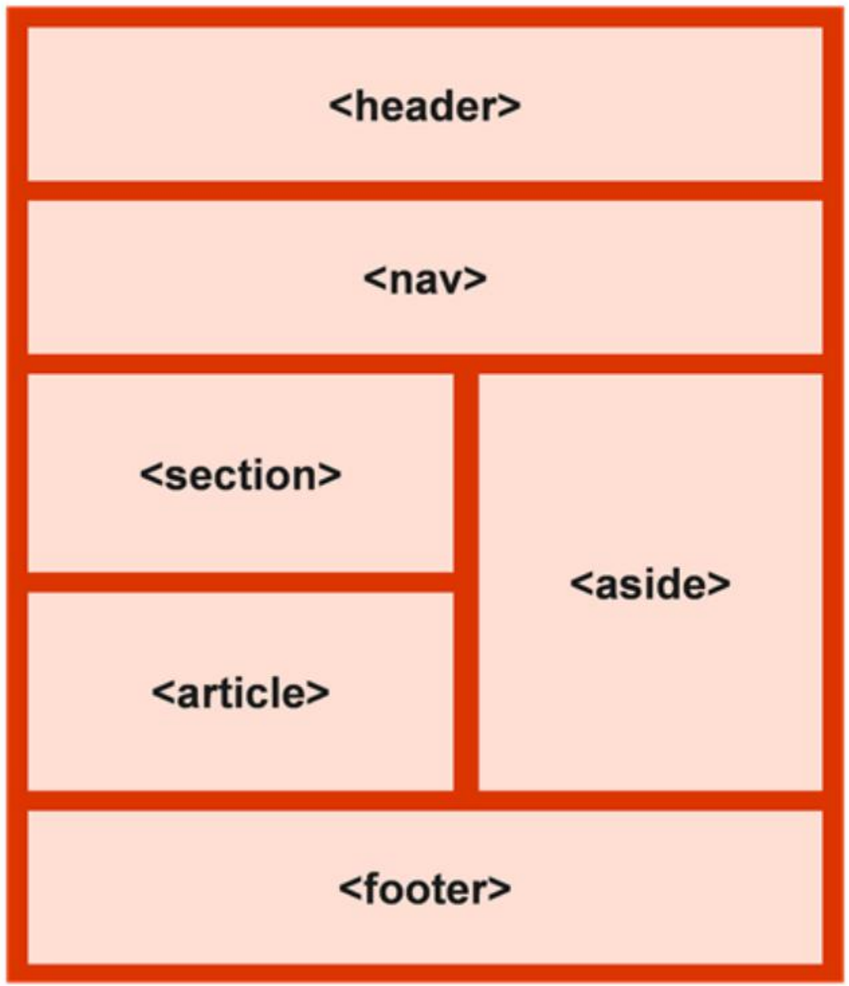
Slide Note

Popular Semantic Elements



Slide Note

Structure of a Web Page Using Semantic Elements



Slide Note

Accessible Rich Internet Applications (ARIA) Labels

ARIA is a spec from the W3C that was created to improve accessibility of web pages and applications by providing extra information to screen readers via HTML attributes.

- ARIA-label is used to provide an invisible label where a visible label cannot be used.
- It has no effect on how elements are displayed or behave in browsers.
- It does not add new functionality and is meant to act only as an extra descriptive layer for the screen readers.

Slide Note

ARIA attributes are divided into two categories: roles and states and properties.

An ARIA role is added via a role="" attribute and does not ever change for an element once it is set. There are four categories of ARIA roles:

- landmark
- document
- widget
- Abstract

ARIA states and properties are often used to support ARIA roles that exist on a page.

ARIA Properties often describe relationships with other elements, and for the most part, do not change once they're set.

Some of the popular properties are:

Aria-autocomplete

Aria-checked

Aria-controls

Aria-label

Aria-labelledby

Aria-disabled

Aria-describedby

Aria-expanded

Aria-hidden

Aria-required

Aria-selected

ARIA Attributes

- ARIA attributes are divided into two categories: roles, and states & properties
- An ARIA role is added via a **role="<ROLE TYPE>"** attribute, and does not ever change for an element once it is set

ARIA Role

ARIA State and Property

```
<nav class="mobile-class" role="navigation"
     aria-label="mobile-menu">
     List of Links
</nav>
```

Slide Note

Quick Check

What tag would you use to emphasize a text?

A. or

B. <emphasize>

C. <bold>

D. None of these

Slide Note

Quick Check: Solution

What tag would you use to emphasize some text?

A. ** or **

B. <emphasize>

C. <bold>

D. None of these

Explanation: Option B and Option C are incorrect as there are no such semantic tags available.

Slide Note

Quick Check

Which element denotes that the text is a short fragment of a computer code?

A. <data>

B. <code>

C. <cite>

D. <dfn>

Slide Note

Quick Check: Solution

Which element denotes that the text is a short fragment of a computer code?

A. <data>

B. **<code>**

C. <cite>

D. <dfn>

Explanation: Option A is correct: the <code> element displays its contents as a short fragment of computer code. Title of work is defined by the <cite> tag.
The <data> element links content with a machine-readable translation.
The defining instance of a term in HTML is represented by the <dfn> element.