

SEMANTIC ELEMENTS

(And some other pieces of Accessibility)

WHAT IS ACCESSIBILITY?

The principle that everyone should be able to use the web equally, regardless of any disability forms the basis of accessibility.

WHAT IS ACCESSIBILITY?

People who have limited use of their arms, people who cannot hear or see well and people who process information differently, all need to access web sites and web applications.

WHAT IS ACCESSIBILITY?

Web technologies have lots of built-in features for accessibility. We must know them so that we do not exclude any of our users.

WHAT IS ACCESSIBILITY?

One of the most significant is
HTML5 semantic elements.

WHAT DOES SEMANTIC MEAN?

So, what the heck are semantic elements?

WHAT THE HECK ARE SEMANTIC ELEMENTS

One place to start is the definition of the word semantic.
What does that mean?

WHAT DOES SEMANTIC MEAN?

Semantic: of or relating to the meanings of words and phrases.

– Merriam-Webster Dictionary

WHAT DOES SEMANTIC MEAN?

In code, the word has basically the same meaning.

WHAT DOES SEMANTIC MEAN?

It refers to the meaning of a piece of code.

WHAT DOES SEMANTIC MEAN?

For example, “what purpose or role does that HTML element have” (rather than “what does it look like?”.)

SEMANTIC ELEMENTS

Semantic elements add meaning to our code. They were created – in HTML5 – primarily to replace the `<div></div>` tag, which is a generic separator tag.

SEMANTIC ELEMENTS

Take a look at a piece of code
like this...

```
<div>
  <a href="" title="Site title">
    <h1>Site title</h1>
  </a>
  <div>
    <ul>
      <li> <a href="#" title="Home">Home</a> </li>
      <li> <a href="#" title="About">About</a> </li>
      <li> <a href="#" title="Sign Up">Sign Up</a> </li>
      <li> <a href="#" title="Contact">Contact</a> </li>
      <li> <a href="#" title="Careers">Careers</a> </li>
    </ul>
  </div>
</div>
```

SEMANTIC ELEMENTS

It looks like this:

Site title

- [Home](#)
- [About](#)
- [Sign Up](#)
- [Contact](#)
- [Careers](#)

SEMANTIC ELEMENTS

But when you just look at the code it's hard to tell what it does.

SEMANTIC ELEMENTS

However, if we use semantic tags, the meaning becomes clear.

SEMANTIC ELEMENTS

```
body>  
  <a href="" title="Site title">  
    Site Title  
  </a>  
  <nav>  
    <ul>  
      <li><a href="" title="Home"></a>Home</li>  
      <li><a href="" title="About"></a>About</li>  
      <li><a href="" title="Sign up"></a>Sign up</li>  
    </ul>  
  </nav>  
</body>  
</html>
```

SEMANTIC ELEMENTS

It doesn't change how the page looks, but we can tell it is a navbar at the top of the page.

SEMANTIC ELEMENTS

Site title

- [Home](#)
- [About](#)
- [Sign Up](#)
- [Contact](#)
- [Careers](#)

SEMANTIC ELEMENTS

That's the point of semantic elements. They're especially important when it comes to coding web pages that are accessible to all users.

SEMANTIC ELEMENTS EXAMPLES

<article> – Is used for an independently distributable piece of content on a page, like a newspaper article.

SEMANTIC ELEMENTS EXAMPLES

<aside> – content that is not part of the flow of the text in which it appears, however still related in some way, like a sidebar.

SEMANTIC ELEMENTS EXAMPLES

<figcaption> – used for photo captions.

<figure> – used for photos.

<footer> – content at the bottom of the page. Often copyright.

SEMANTIC ELEMENTS EXAMPLES

<header> – The opposite of footer. It usually contains the main header element on the page and navigation.

<section> – Similar to article but less independent.

SEMANTIC ELEMENTS EXAMPLES

<label> – It wraps the text for a specific form control item. It's usually the name or label for a choice. This ties meaning to the item and makes the form more readable.

SEMANTIC ELEMENTS EXAMPLES

<fieldset> – This element surrounds an entire grouping of radio buttons to show that these choices are part of a set.

SEMANTIC ELEMENTS EXAMPLES

<time> – This element is used to wrap a date or time on a page.

POP QUIZ

How many heading elements are available to us in HTML?

POP QUIZ

There
are
six.

```
<body>
```

```
  <h1>I'm an h1</h1>
```

```
  <h2>I'm an h2</h2>
```

```
  <h3>I'm an h3</h3>
```

```
  <h4>I'm an h4</h4>
```

```
  <h5>I'm an h5</h5>
```

```
  <h6>I'm an h6</h6>
```

```
</body>
```

POP QUIZ

There are
six. Headings with
equal (or higher)
rank start new
implied sections...

I'm an h1

I'm an h2

I'm an h3

I'm an h4

I'm an h5

I'm an h6

POP QUIZ

...Headings with lower rank start subsections of the previous one.

I'm an h1

I'm an h2

I'm an h3

I'm an h4

I'm an h5

I'm an h6

A FEW ODDS AND ENDS

We remember our friend the `input` and its friend the `type` attribute. We can use a date value for the `type` attribute to create a date-picker.


A FEW ODDS AND ENDS

The code will look like this...

```
<form>
  <label for="party">Party:</label>
  <input type="date" id="party" name="party">
</form>
```

A FEW ODDS AND ENDS

We'll see something like this:

Party: 

The user can click on this to select a date for the party.

A FEW ODDS AND ENDS

It's also worth noting when we look at this code the use of the **for** attribute.

```
<form>
  <label for="party">Party:</label>
  <input type="date" id="party" name="party">
</form>
```

A FEW ODDS AND ENDS

The **for** attribute on the label shares a value with the id attribute on the input.

```
<form>
  <label for="party">Party:</label>
  <input type="date" id="party" name="party">
</form>
```

A FEW ODDS AND ENDS

That's a best practice.



```
<form>  
  <label for="party">Party:</label>  
  <input type="date" id="party" name="party">  
</form>
```

A FEW ODDS AND ENDS

We can also use an access key attribute to assign a quick key for the user activate an element.

```
<form>  
  <label for="party">Party:</label>  
  <input type="text" id="party" name="party" accesskey="i">  
</form>
```

A FEW ODDS AND ENDS

Access keys can be tricky. They may conflict with other key standards in the browser.

```
<form>
  <label for="party">Party:</label>
  <input type="text" id="party" name="party" accesskey="i">
</form>
```


A FEW ODDS AND ENDS

To avoid this, most browsers will use access keys only if pressed together with the Alt key.

```
<form>  
  <label for="party">Party:</label>  
  <input type="text" id="party" name="party" accesskey="i">  
</form>
```

A FEW ODDS AND ENDS

You'll also find that some folks say not to use access keys because they may cause problems for international users whose keyboards aren't the same as ours.

A FEW ODDS AND ENDS

Still, they can be a way to make the page more accessible to users who can't use a mouse.

A FEW ODDS AND ENDS

Another way to accomplish a similar thing is the `tabindex` attribute. It specifies the tab order of an element (when the “tab” button is used for navigating).

A FEW ODDS AND ENDS

It looks like this...

```
<form action="">
  <label for="party">Party:</label>
  <input type="text" id="party" name="party" tabindex="1">
  <label for="gift">Gift:</label>
  <input type="text" id="gift" name="gift" tabindex="2">
</form>
```

A FEW ODDS AND ENDS

Now, if the user hits the “tab” button, it will allow her to access the first input in our form. If she does it again, she can access the second.

A FEW ODDS AND ENDS

It will look like this...

Party: Gift:

A FEW ODDS AND ENDS

The `tabindex` attribute should be used with positive numbers as values, and the elements with lower values will be focused first.

A FEW ODDS AND ENDS

If multiple elements share the same tabindex, they will be navigated in the order they appear in the HTML source.

A FEW ODDS AND ENDS

The text we put in our links also can play a part in accessibility. The text should be concise but descriptive. This is important for users with screen readers.

A FEW ODDS AND ENDS

For example...

```
<a href="https://www.w3schools.com">Go to w3schools</a>
```

It's short and tells the user exactly what it does.

[Go to w3schools](https://www.w3schools.com)