Structure Data

What is Microdata?

- **Microdata** is a WHATWG HTML specification used to nest metadata within existing content on web pages.

- WHATWG - The **Web Hypertext Application Technology Working Group** (**WHATWG**) is a community of people interested in evolving HTML and related technologies.
- **Metadata** is - data that provides information about other data

<h1>Avatar</h1>

- What does this refer to?

<h1>Avatar</h1>

- What does this refer to?
  - A movie
  - A profile picture

So, on a web page….

- How would google or any other search engine know how to catagorize this?
- Short answer – It wouldn’t
- So, let’s put some info (microdata) with this to help identify it.

How to mark up your content using microdata

- **Why use microdata?**

  Your web pages have an underlying meaning that people understand when they read the web pages. But search engines have a limited understanding of what is being discussed on those pages. By adding additional tags to the HTML of your web pages—tags that say, "Hey search engine, this information describes this specific movie, or place, or person, or video"—you can help search engines and other applications better understand your content and display it in a useful, relevant way.

  Microdata is a set of tags, introduced with HTML5, that allows you to do this.

- **itemscope and itemtype**

- **itemprop**

So back to the AVATAR example

Let's start with a concrete example. Imagine you have a page about the movie Avatar—a page with a link to a movie trailer, information about the director, and so on. Your HTML code might look something like this:

```
<div>
    <h1>
        Avatar
    </h1>
    <span>
        Director: James Cameron (born August 16, 1954)
    </span>
    <span>
        Science fiction
    </span>
    <a href="../movies/avatar-theatrical-trailer.html">
       Trailer
   </a>
</div>
```

itemscope

itemscope is a boolean global attribute that defines the scope of associated metadata. Specifying the **itemscope** attribute for an element creates a new item, which results in a number of name-value pairs that are associated with the element.

```
<div>
   <h1>
      Avatar
   </h1>
   <span>
      Director: James Cameron (born August 16, 1954)
   </span>
   <span>
      Science fiction
   </span>
   <a href="../movies/avatar-theatrical-trailer.html">
     Trailer
  </a>
</div>
```

This is all one item

itemscope

To begin, identify the section of the page that is "about" the movie Avatar. To do this, add the itemscope element to the HTML tag that encloses information about the item, like this:

```
  <div itemscope>
     <h1>
       Avatar
    </h1>
    <span>
       Director: James Cameron (born August 16, 1954)
    </span>
    <span>
      Science fiction
    </span>
    <a href="../movies/avatar-theatrical-trailer.html">
      Trailer
    </a>
</div>
```

By adding itemscope, you are specifying that the HTML contained in the <div>...</div> block is about a particular item.

itemtype

But it's not all that helpful to specify that there is an item being discussed without specifying what kind of an item it is. You can specify the type of item using the **itemtype** attribute immediately after the itemscope.

```
  <div itemscope itemtype="http://schema.org/Movie” >
     <h1>
        Avatar
    </h1>
    <span>
       Director: James Cameron (born August 16, 1954)
    </span>
    <span>
      Science fiction
    </span>
    <a href="../movies/avatar-theatrical-trailer.html">
      Trailer
    </a>
</div>
```

Now, Properties

Each of these have a standard set of properties.

Let’s look at

https://schema.org/docs/schemas.html

- Browse the full hierarchy in HTML:
  - One page per type
  - Full list of types, shown on one page

itemprop

What additional information can we give search engines about the movie Avatar? Movies have interesting properties such as actors, director, ratings.

To label properties of an item, use the **itemprop** attribute. For example, to identify the director of a movie, add **itemprop="director"** to the element enclosing the director's name. (There's a full list of all the properties you can associate with a movie at http://schema.org/Movie.)

```
<div itemscope itemtype ="http://schema.org/Movie">
    <h1 itemprop="name">
        Avatar
    </h1>
    <span>
        Director:
        <span itemprop="director">
            James Cameron
        </span>
        (born August 16, 1954)
    </span>
    <span itemprop="genre">
        Science fiction
    </span>
    <a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">
        Trailer
    </a>
</div>
```

Note that we added additional <span>...</span> tags to attach the itemprop attributes to the appropriate text on the page. <span> tags don't change the way pages are rendered by a web browser, so they are a convenient HTML element to use with itemprop.

Search engines can now understand not just that http://www.avatarmovie.com is a URL, but also that it's the URL for the trailer for the science-fiction movie Avatar, which was directed by James Cameron.

So that’s it, right?

But no, it doesn’t stop there.

Sometimes the value of an item property can itself be another item with its own set of properties. For example, we can specify that the director of the movie is an item of type Person and the Person has the properties name and birthDate. To specify that the value of a property is another item, you begin a new itemscope immediately after the corresponding itemprop.

So what does that look like?

Embedded items

```
<div itemscope itemtype ="http://schema.org/Movie">
    <h1 itemprop="name">
        Avatar
    </h1>
    <div itemprop="director" itemscope itemtype="http://schema.org/Person">
        Director:
        <span itemprop="name" >
        James Cameron
        </span>
        (born
        <span itemprop="birthDate">
            August 16, 1954
        </span>
        )
    </div>
    <span itemprop="genre">
        Science fiction
    </span>
    <a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">
        Trailer
    </a>
</div>
```

What’s next for the Web Developer

- Spend some time studying Schema.org
- Begin to add structured data to you clients sites
  - It is not an all or none kinda thing. You can modify pages or portions of pages as time permits. Every little bit will help
- Go to Google
  - https:// www.google.com › webmasters › markup-helper
- Go to Mozilla
  - https://moz.com/learn/seo/schema-structured-data
- Then put on some coffee…lots of it.