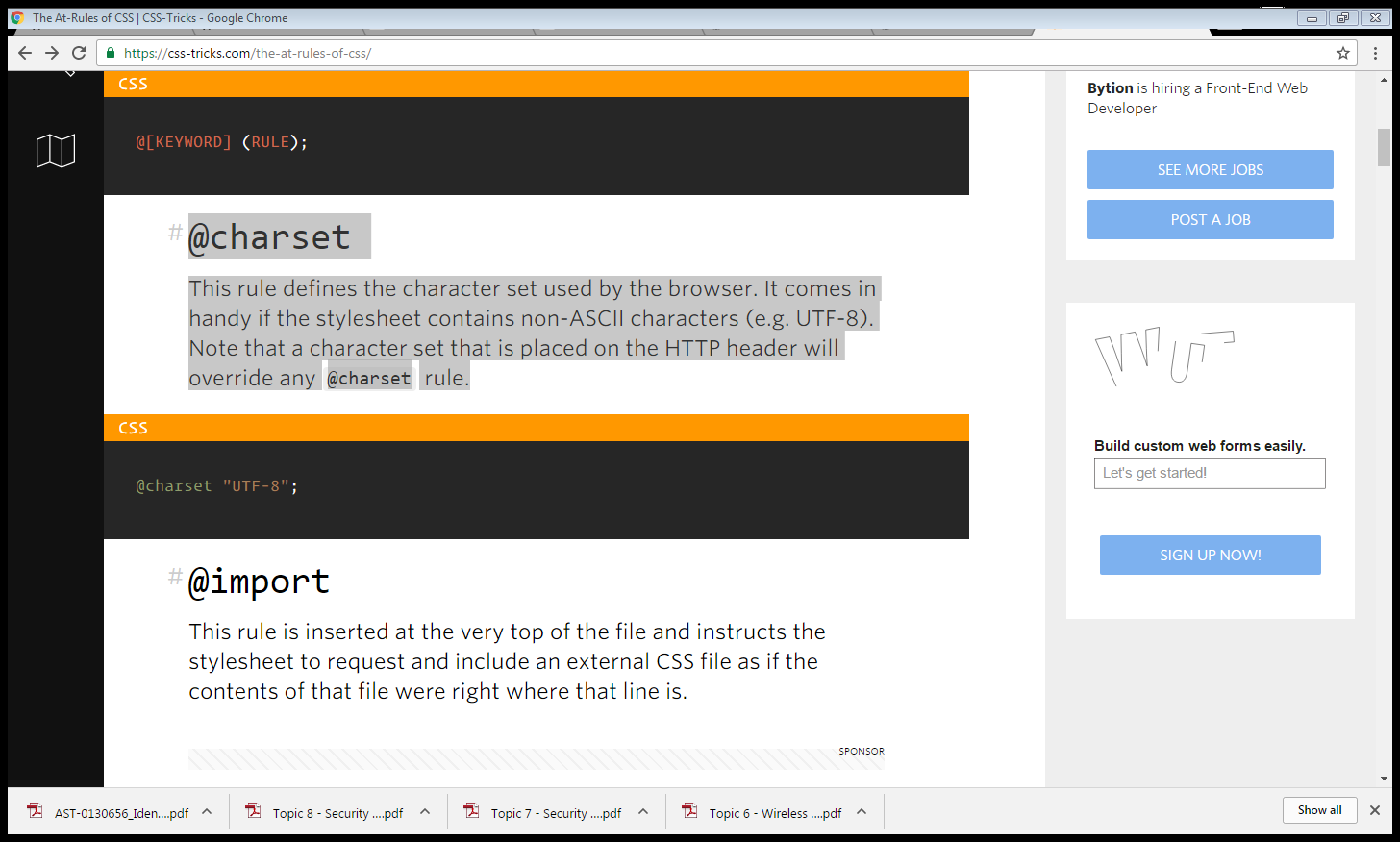
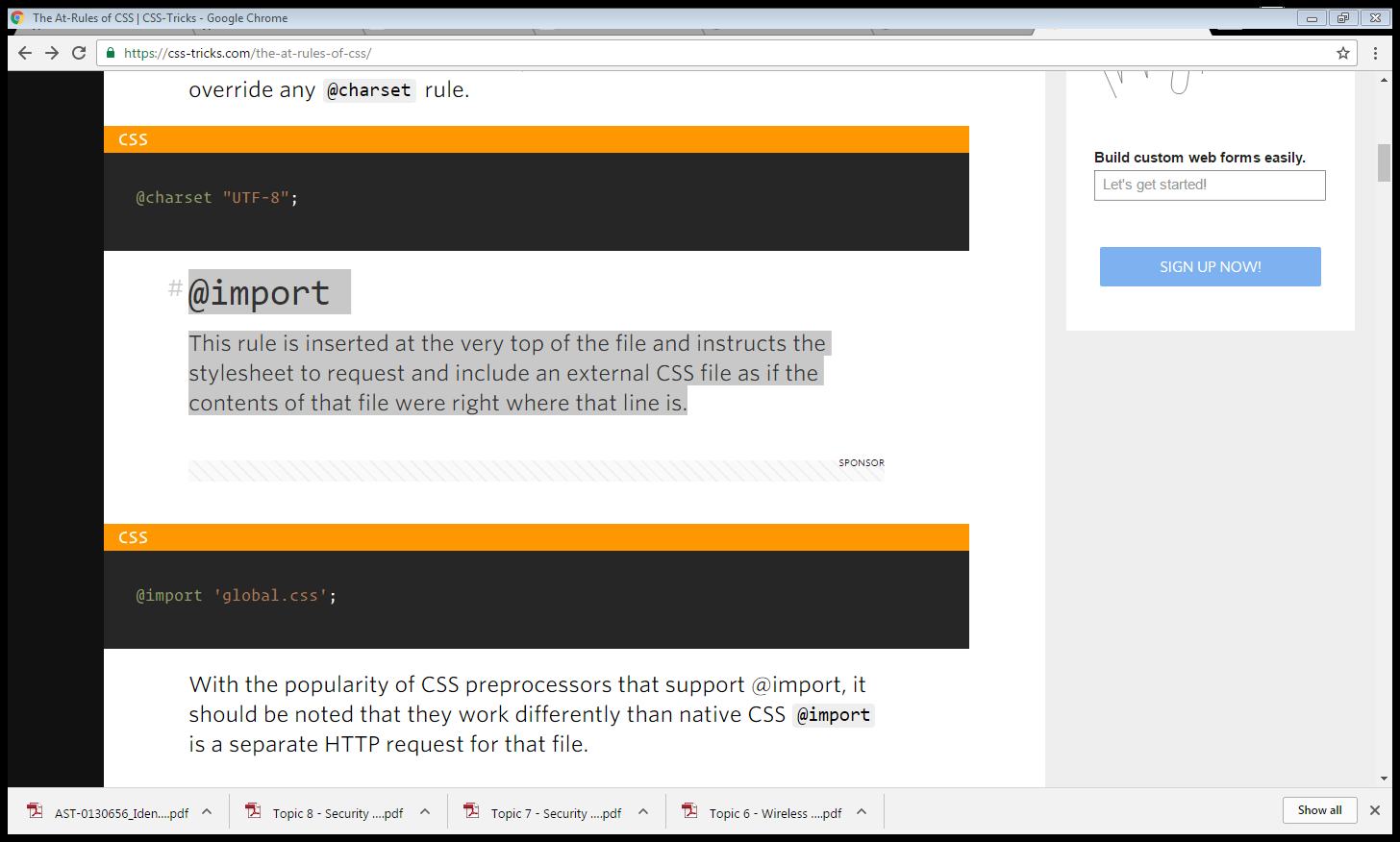
HTML style sheets

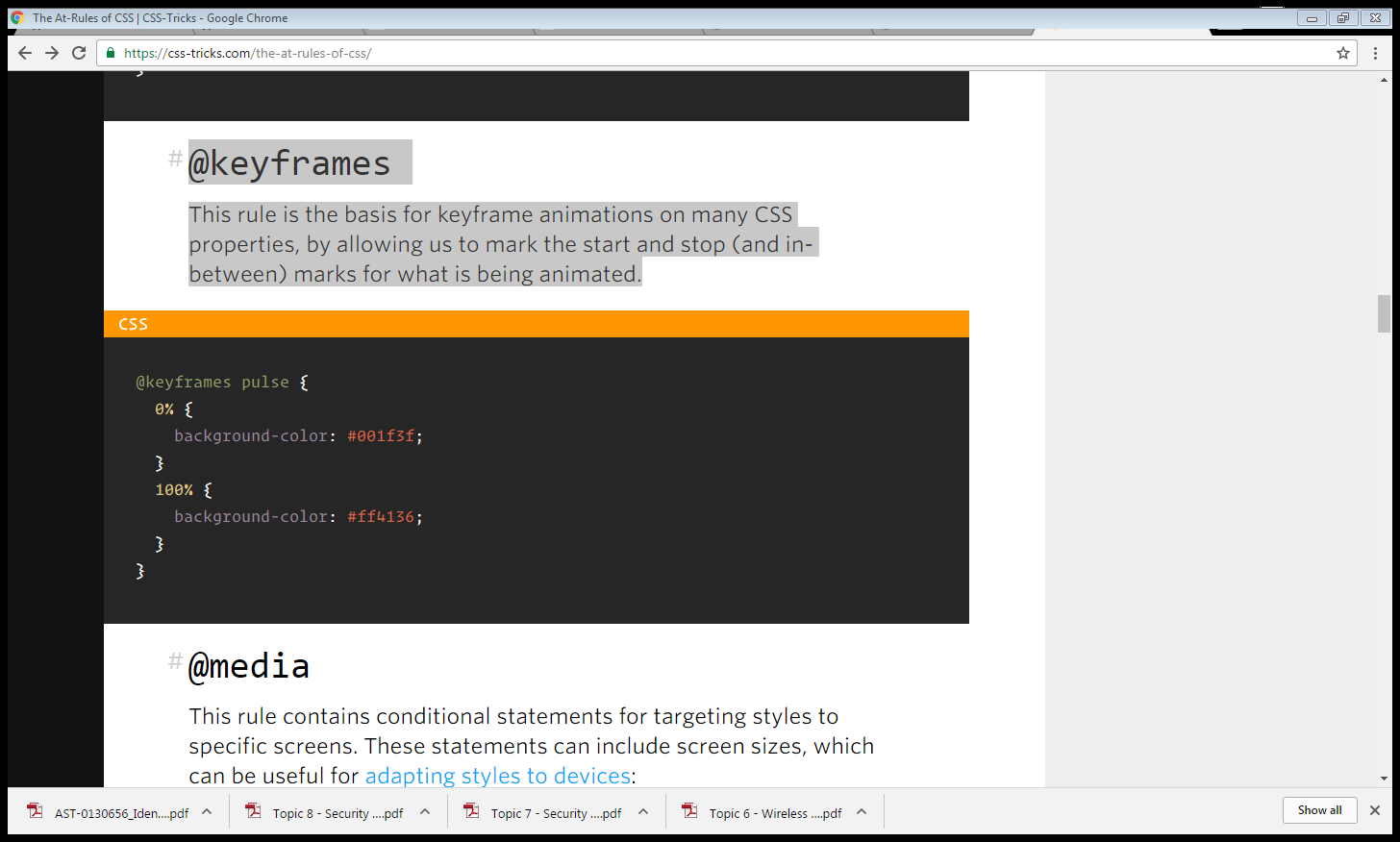
* Author Styles - The author specifies style sheets for a source document according to the conventions of the document language. For instance, in HTML, style sheets may be included in the document or linked externally.
  + **Inline** - by using the style attribute in HTML elements
  + **Embedded** - by using a <style> element in the <head> section
  + **External** - by using an external CSS file
* User Styles - The user may be able to specify style information for a particular document. For example, the user may specify a file that contains a style sheet or the user agent may provide an interface that generates a user style sheet (or behaves as if it did).
* User Agent Style -  [Conforming user agents](https://www.w3.org/TR/CSS2/conform.html" \l "conformance) must apply a default style sheet (or behave as if they did). A user agent's default style sheet should present the elements of the document language in ways that satisfy general presentation expectations for the document language (e.g., for visual browsers, the EM element in HTML is presented using an italic font).

CSS Syntax

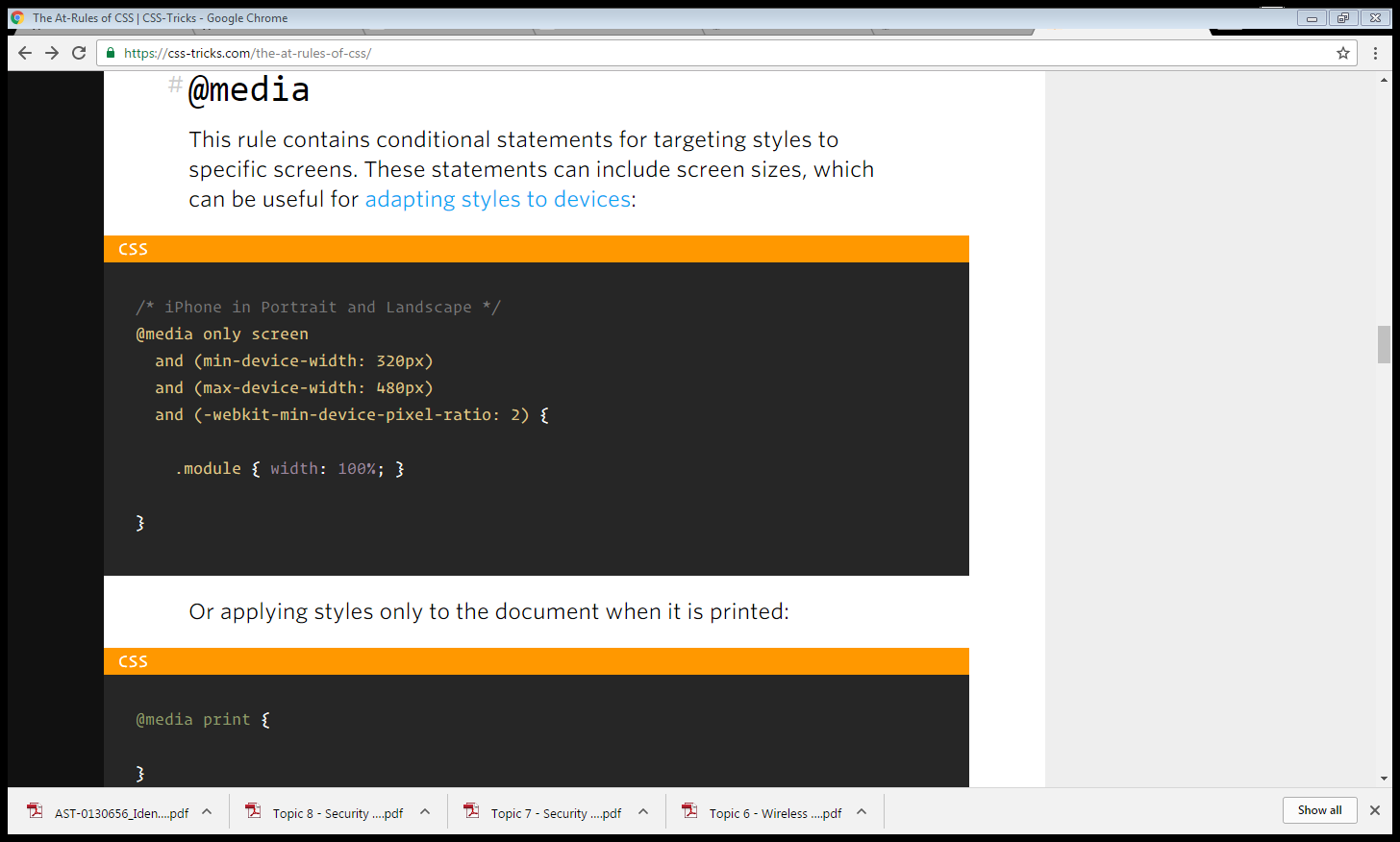
* At-rules
  + At-rules start with an at-keyword, an '@' character followed immediately by an identifier (for example, '@import', '@page').
    - @charset - This rule defines the character set used by the browser. It comes in handy if the stylesheet contains non-ASCII characters (e.g. UTF-8). Note that a character set that is placed on the HTTP header will override any @charset rule. 
    - @import - This rule is inserted at the very top of the file and instructs the stylesheet to request and include an external CSS file as if the contents of that file were right where that line is.



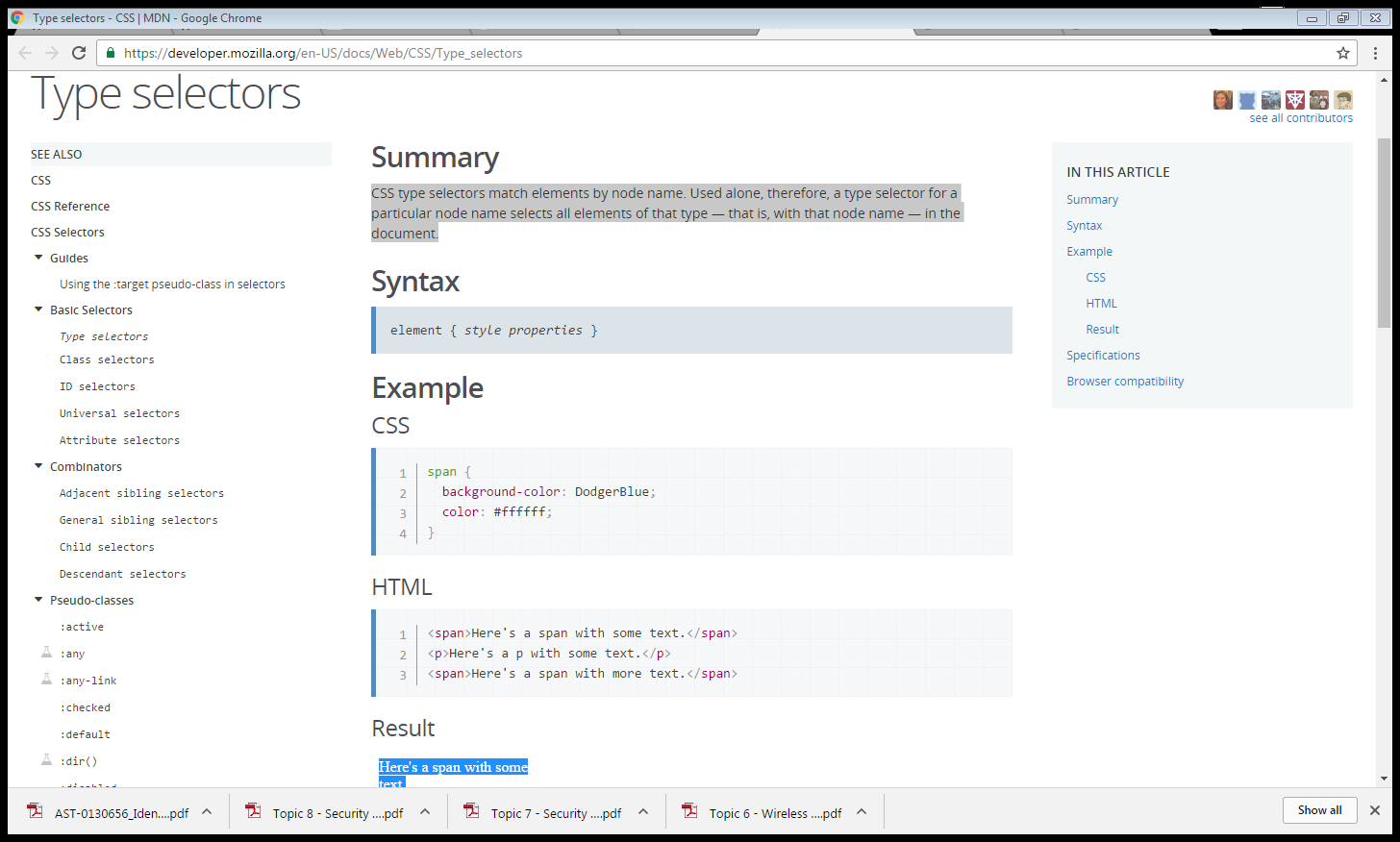
* + - @keyframes - This rule is the basis for keyframe animations on many CSS properties, by allowing us to mark the start and stop (and in-between) marks for what is being animated.



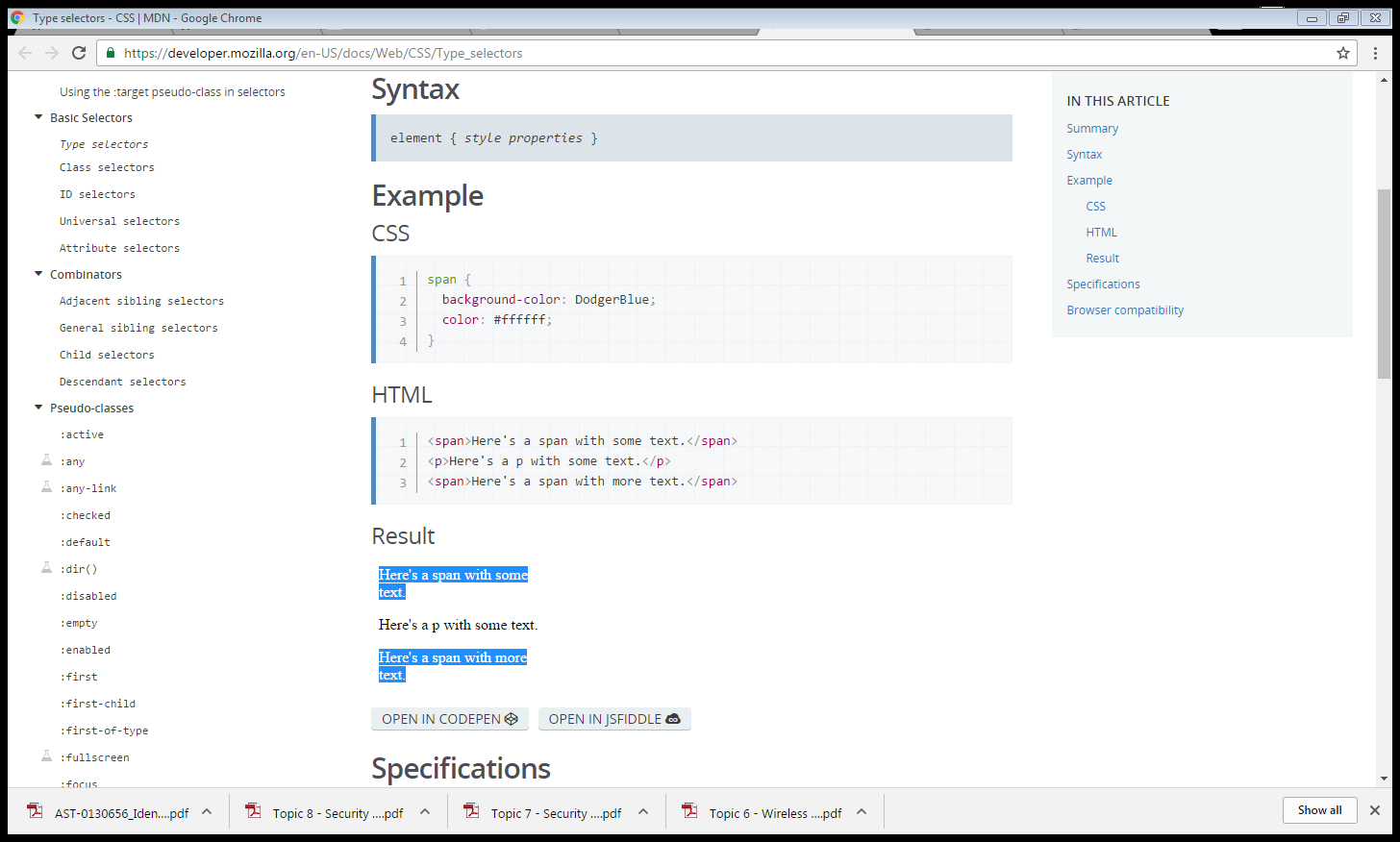
* + - @media - This rule contains conditional statements for targeting styles to specific screens. These statements can include screen sizes, which can be useful for adapting styles to devices:



* CSS Selector
  + Simple Selectors
    - **CSS type selectors**- CSS type selectors match elements by node name. Used alone, therefore, a type selector for a particular node name selects all elements of that type — that is, with that node name — in the document.
      * Syntax



* + - * Example



* + - **Universal selectors -** An asterisk (\*) is the universal selector for CSS. It matches a single element of any type. Omitting the asterisk with simple selectors has the same effect. For instance, \*.warning and .warning are considered equal.

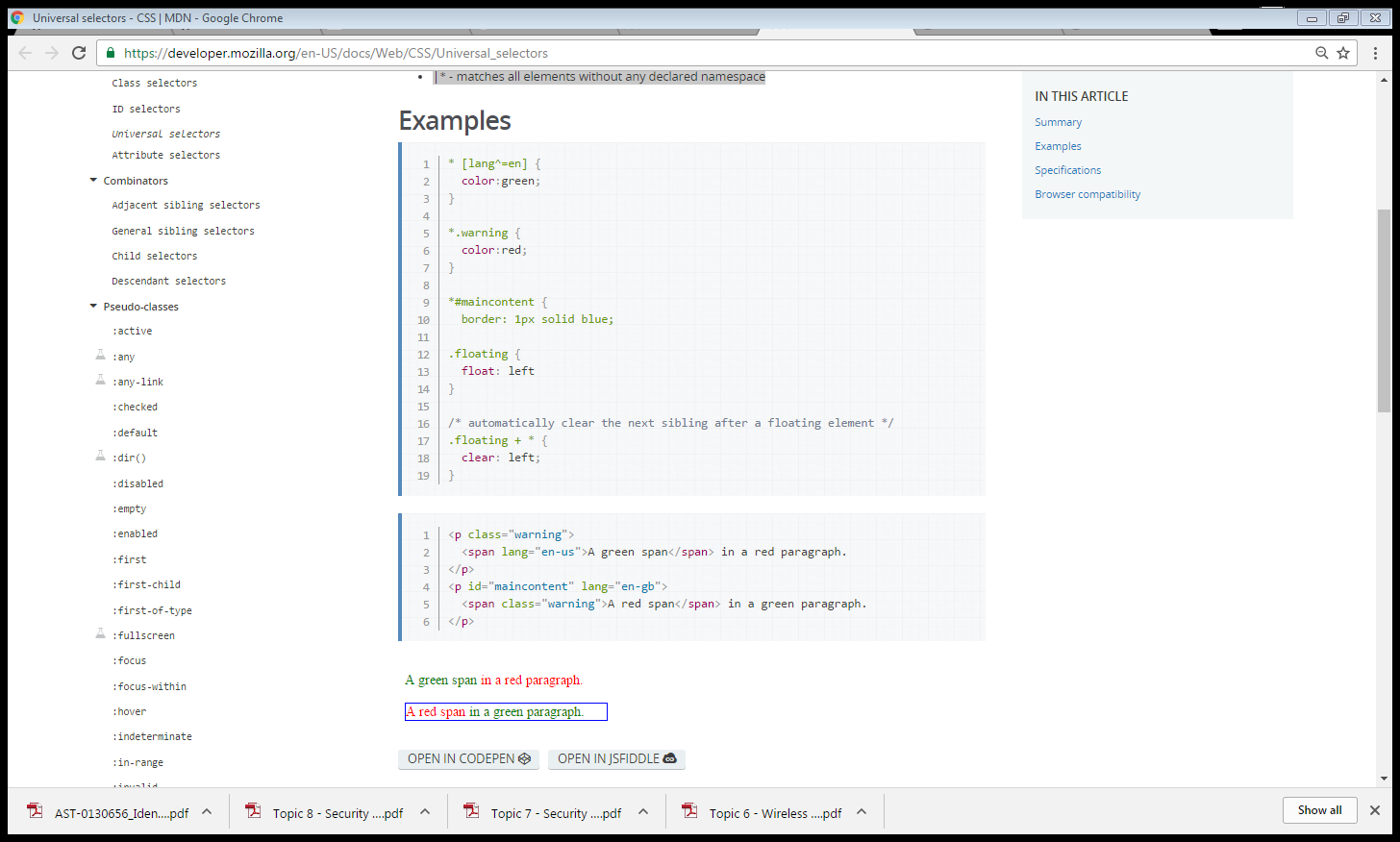
In CSS 3, the asterisk may be used in combination with namespaces:

ns|\* - matches all elements in namespace ns

\*|\* - matches all elements

|\* - matches all elements without any declared namespace

* + - * Example



* + - **Attribute selectors** - Attribute selectors select an element using the presence of a given attribute or attribute value.

[attr] Represents an element with an attribute name of attr.

[attr=value] Represents an element with an attribute name of attr and whose value is exactly "value".

[attr~=value] Represents an element with an attribute name of attr whose value is a whitespace-separated list of words, one of which is exactly "value".

[attr|=value] Represents an element with an attribute name of attr. Its value can be exactly “value” or can begin with “value” immediately followed by “-” (U+002D). It can be used for language subcode matches.

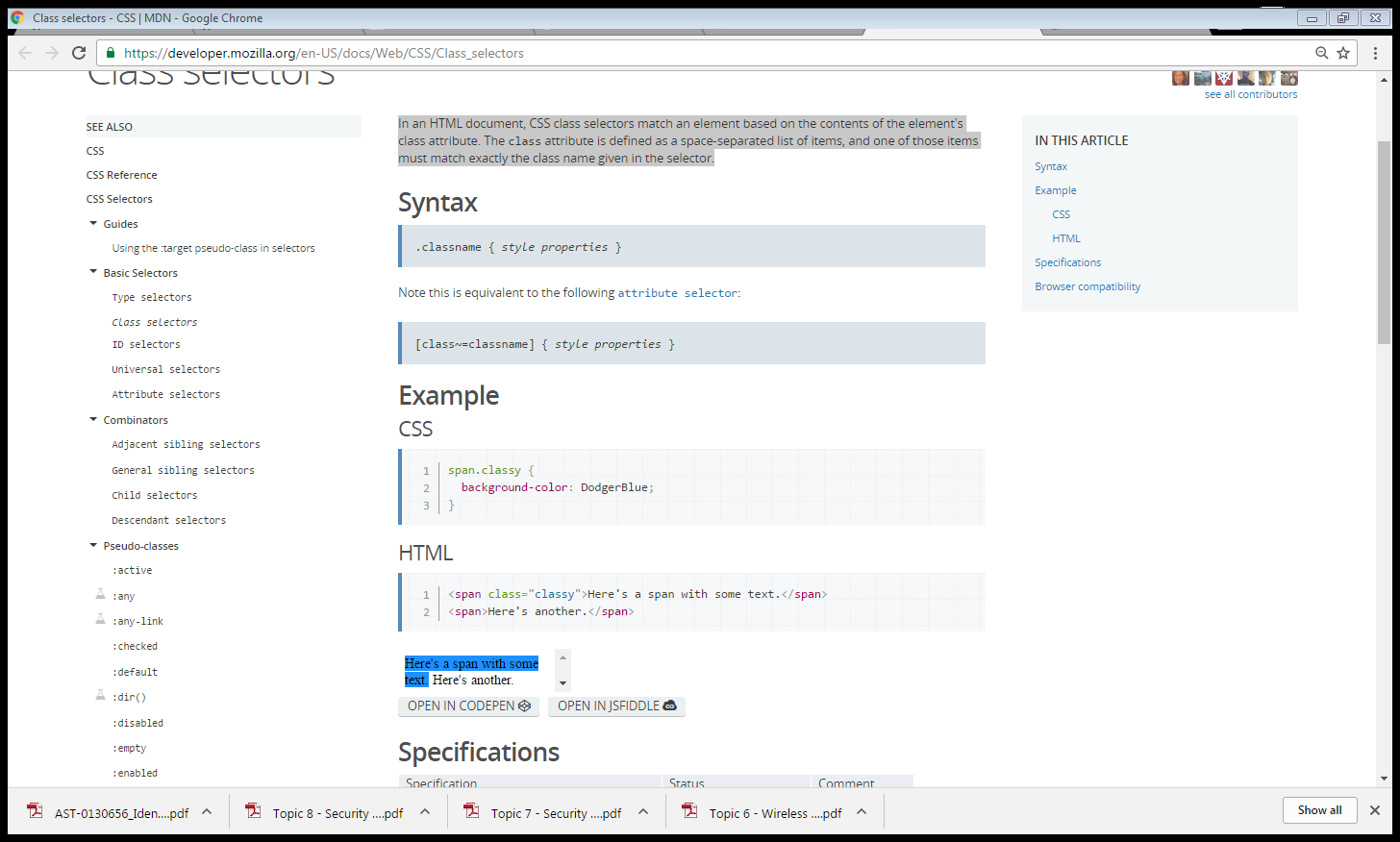
[attr^=value] Represents an element with an attribute name of attr and whose first value is prefixed by "value".

[attr$=value] Represents an element with an attribute name of attr and whose last value is suffixed by "value".

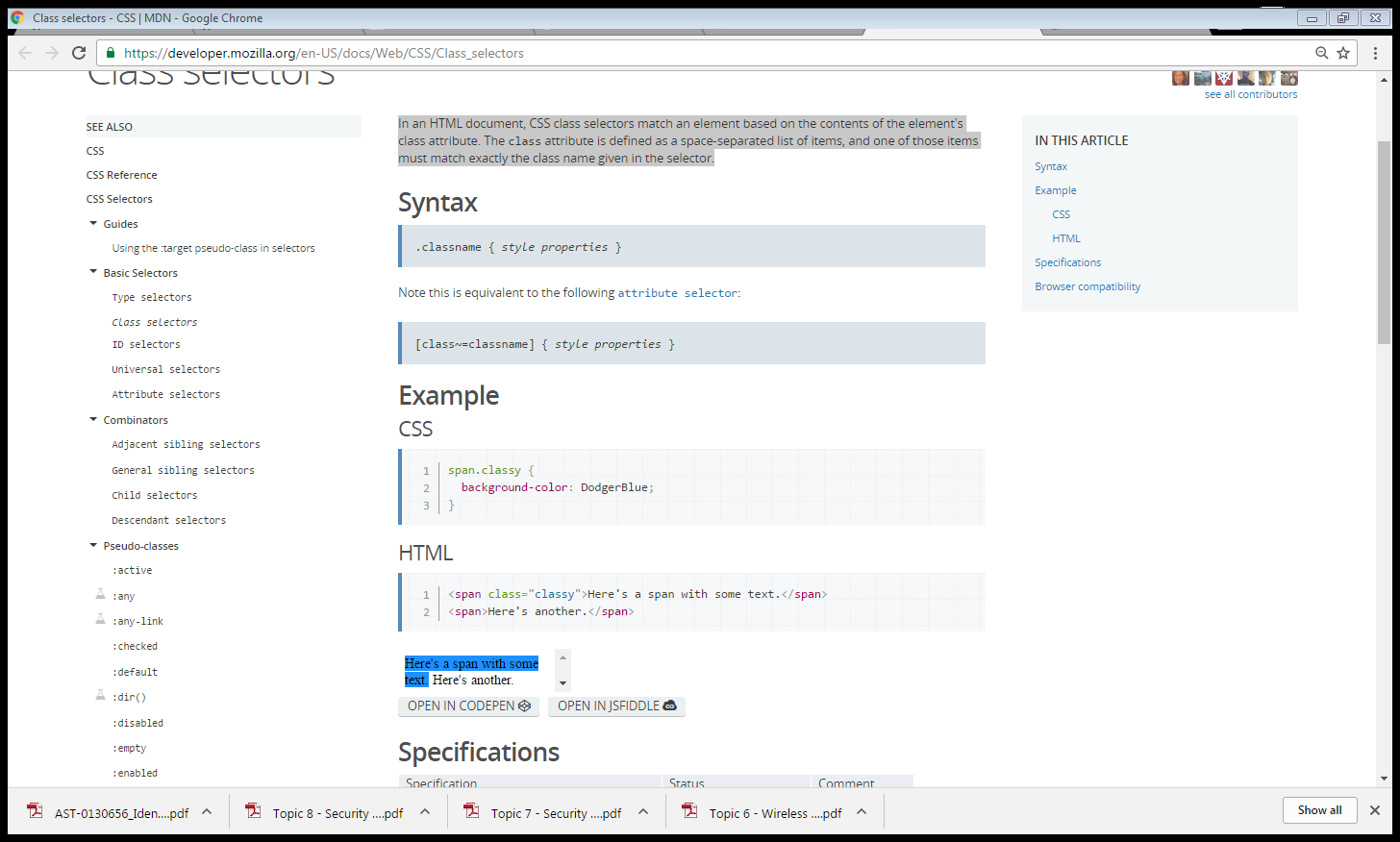
[attr\*=value] Represents an element with an attribute name of attr and whose value contains at least one occurrence of string "value" as substring.

[attr operator value i] Adding an i (or I) before the closing bracket causes the value to be compared case-insensitively (for characters within the ASCII range).

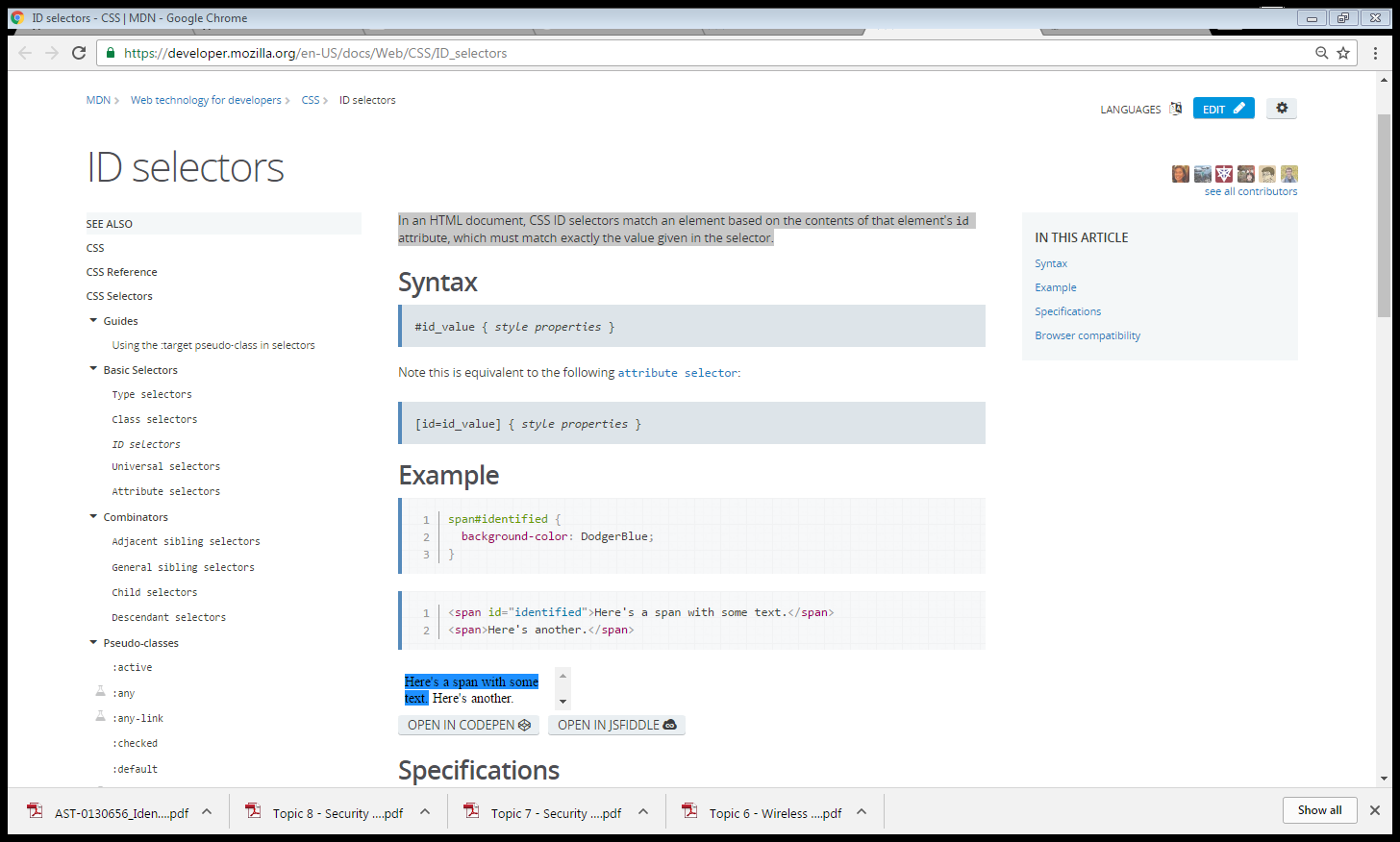
* + - **Class selectors** - In an HTML document, CSS class selectors match an element based on the contents of the element's class attribute. The class attribute is defined as a space-separated list of items, and one of those items must match exactly the class name given in the selector.
      * Syntax



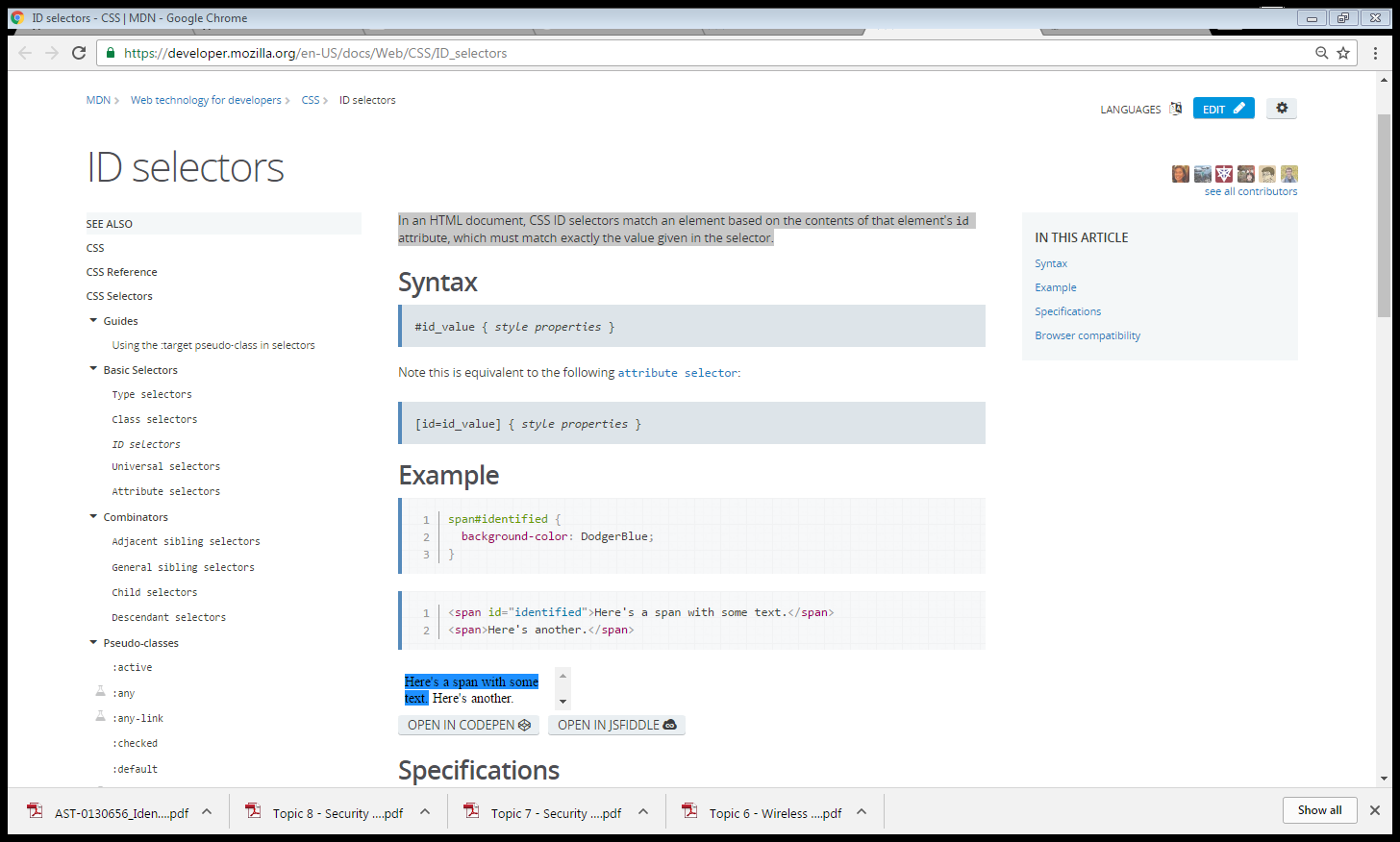
* + - * Example



* + - **ID selectors -** In an HTML document, CSS ID selectors match an element based on the contents of that element's id attribute, which must match exactly the value given in the selector.
      * Syntax



* + - * Example



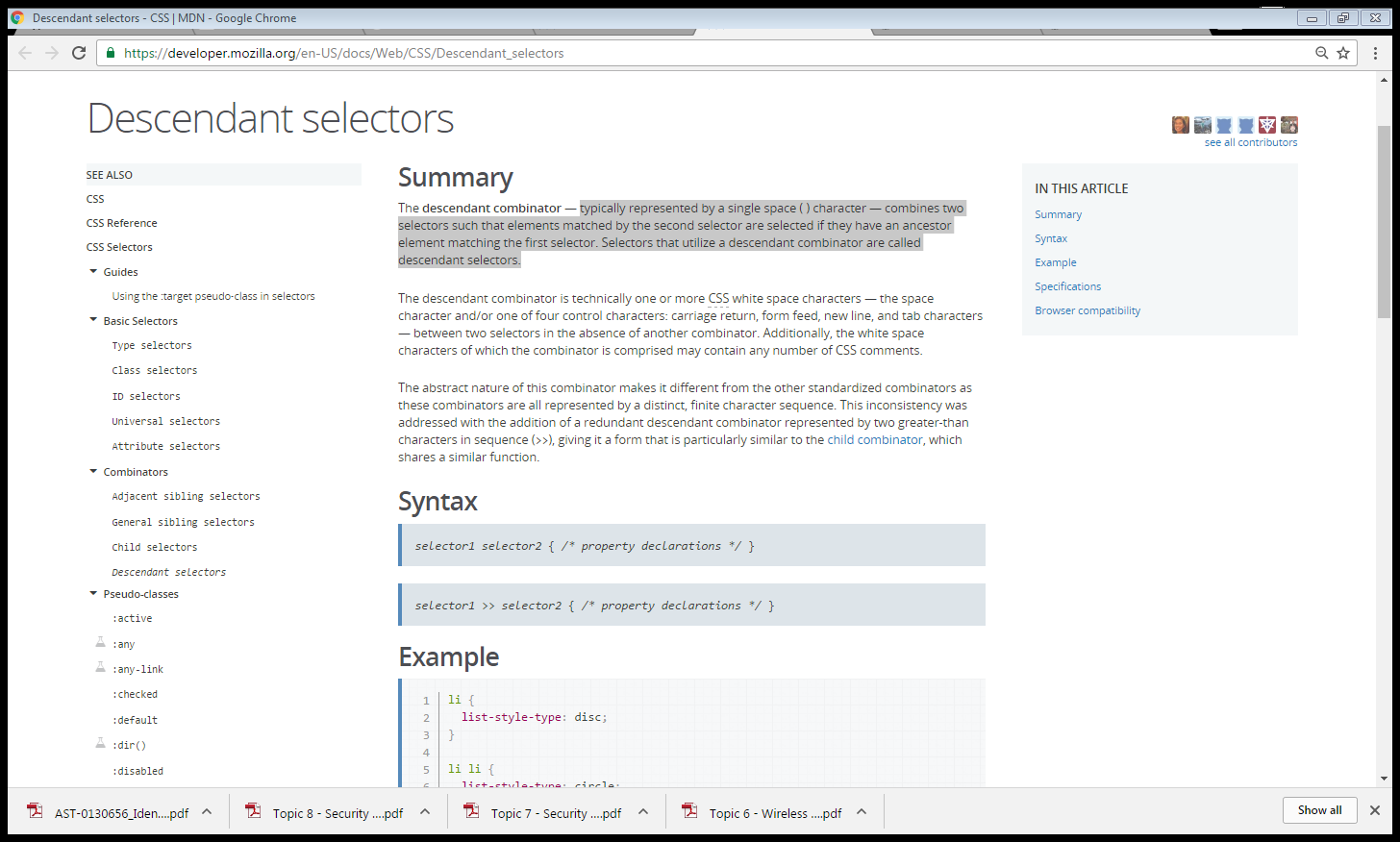
* + Pseudo-class
    - Dynamic
      * Link
        + :link - The :link CSS pseudo-class lets you select links inside elements. This will select any link which has not yet been visited, even those already styled using selector with other link-related pseudo-classes like :hover, :active or :visited.
        + :visited - The :visited CSS pseudo-class lets you select only links that have been visited. This style may be overridden by any other link-related pseudo-classes, that is :link, :hover, and :active, appearing in subsequent rules.
      * User action pseudo-classes
        + :hover - The :hover CSS pseudo-class matches when the user designates an element with a pointing device, but does not necessarily activate it.
        + :active - The :active CSS pseudo-class matches when an element is being activated by the user.
        + :focus - The :focus CSS pseudo-class is applied when an element has received focus, either from the user selecting it with the use of a keyboard or by activating with the mouse (e.g. a form input).
      * Target pseudo-class
        + :target - The :target pseudo-class represents the unique element, if any, with an id matching the fragment identifier of the URI of the document.
      * Language pseudo-class
        + :lang() - The :lang CSS pseudo-class matches elements based on the language the element is determined to be in.
      * UI element state (interaction)
        + :enabled - The :enabled CSS pseudo-class represents any enabled element.
        + :disabled - The :disabled CSS pseudo-class represents any disabled element.
        + :checked - The :checked CSS pseudo-class selector represents any radio (<input type="radio">), checkbox (<input type="checkbox">) or option (<option> in a <select>) element that is checked or toggled to an on state.
        + :indeterminate - The :indeterminate CSS pseudo-class represents:

any <input type="checkbox"> element whose indeterminate DOM property is set to true by JavaScript

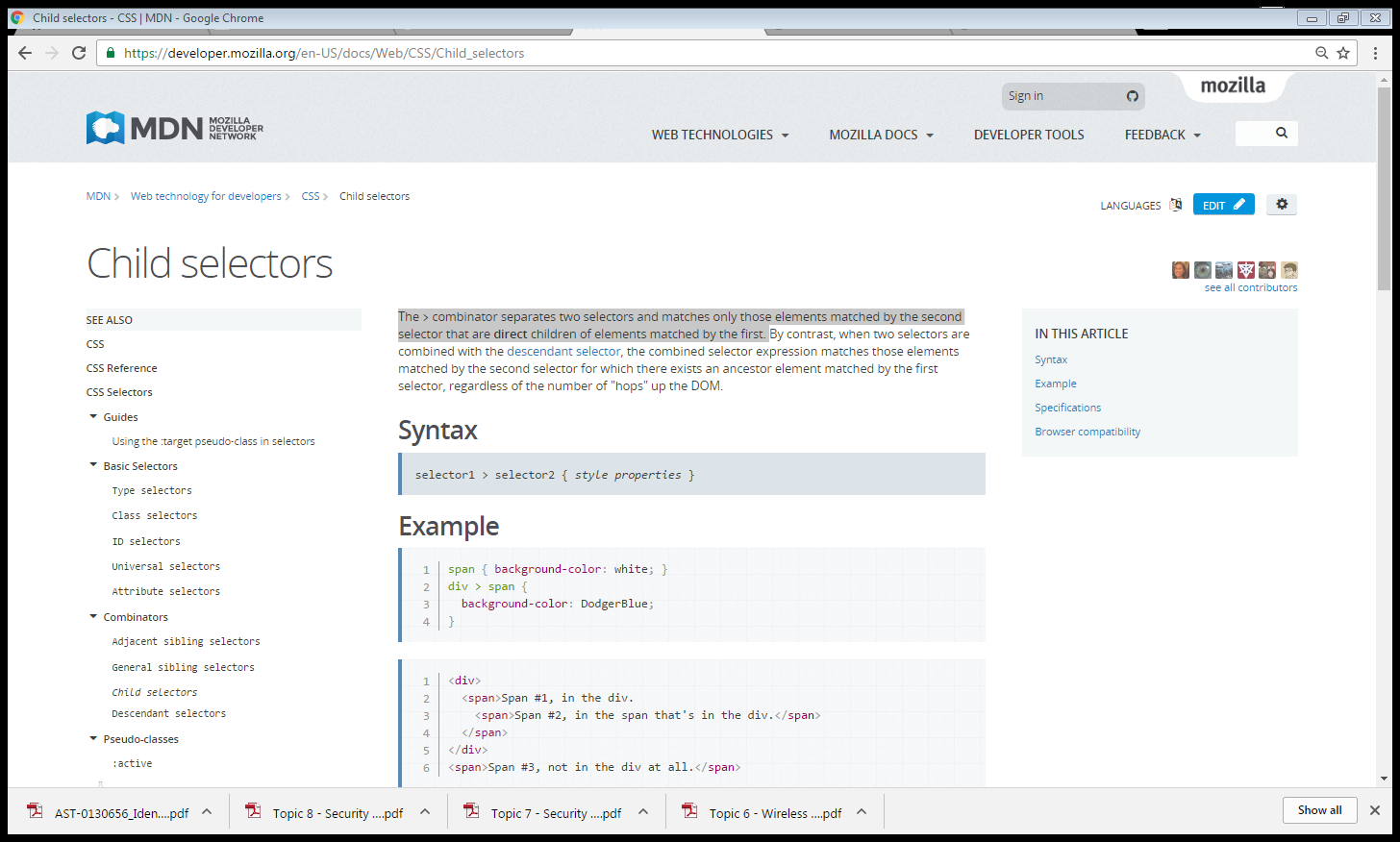
<input type="radio"> elements whose radio button group's radio buttons are all unchecked

<progress> elements in an indeterminate state

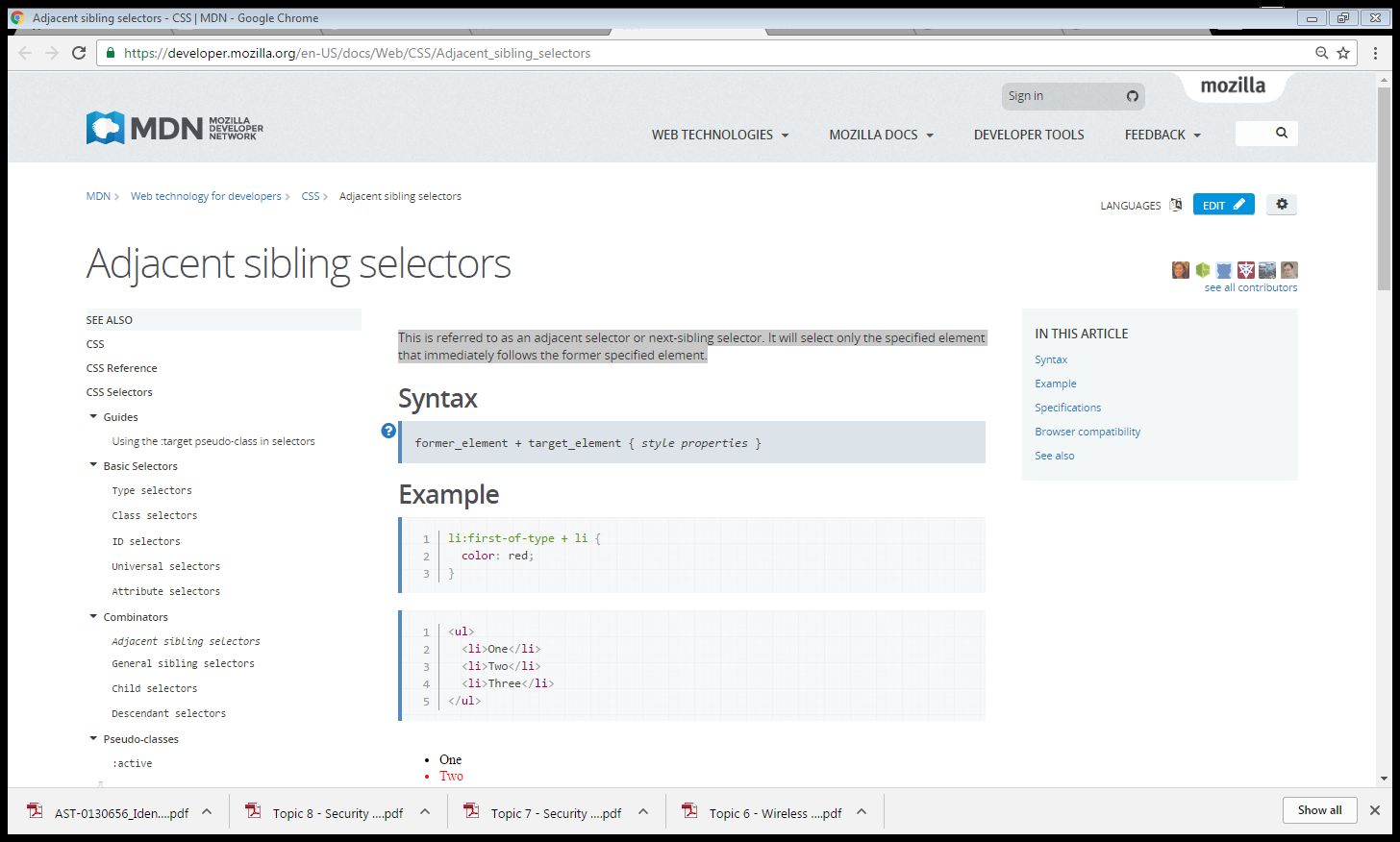
* + - * Psuedo-classes Structural
        + :root - The :root CSS pseudo-class matches the root element of a tree representing the document.
        + :first-child - The :first-child CSS pseudo-class represents any element that is the first among a group of sibling elements.
        + :last-child - The :last-child CSS pseudo-class represents any element that is the last among a group of sibling elements.
        + :only-child - The :only-child CSS pseudo-class represents any element which is the only element in a context where a group of siblings would be expected.
        + :nth-child() - The :nth-child(an+b) CSS pseudo-class matches an element that has an+b-1 siblings before it in the document tree, for a given positive or zero value for n.
        + :nth-last-child() - The :nth-last-child(an+b) CSS pseudo-class matches an element that has an+b-1 siblings after it in the document tree, for a given positive or zero value for n.
  + Combinators
    - **Descendant selectors** - typically represented by a single space ( ) character — combines two selectors such that elements matched by the second selector are selected if they have an ancestor element matching the first selector. Selectors that utilize a descendant combinator are called descendant selectors.
      * Syntax



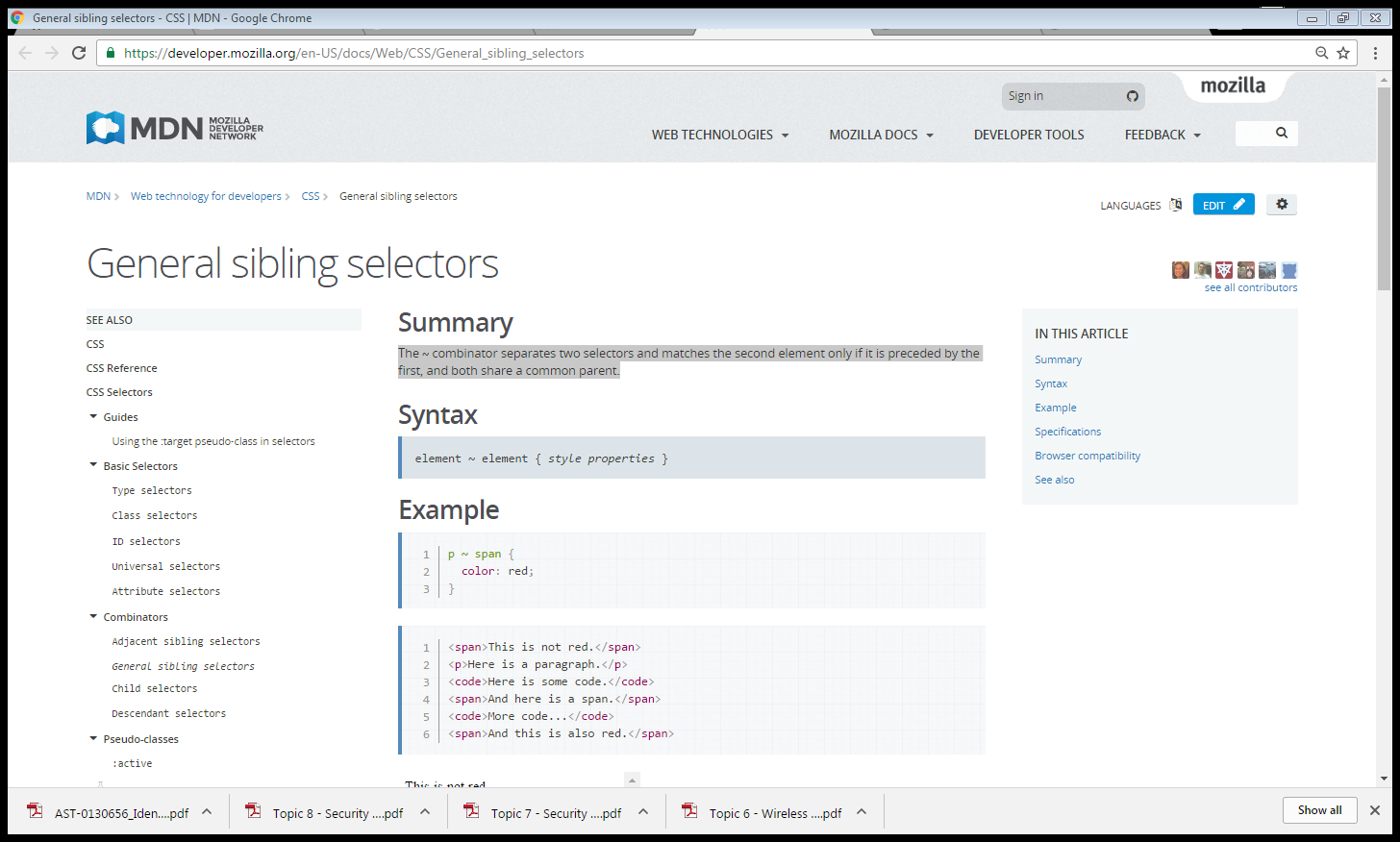
* + - **Child selectors** - The > combinator separates two selectors and matches only those elements matched by the second selector that are direct children of elements matched by the first.
      * Syntax



* + - **Sibling selectors**
      * Adjacent sibling selectors - This is referred to as an adjacent selector or next-sibling selector. It will select only the specified element that immediately follows the former specified element.
        + Syntax



* + - * General sibling selectors - The ~ combinator separates two selectors and matches the second element only if it is preceded by the first, and both share a common parent.
        + Syntax



* + **Pseudo-elements** - Pseudo-elements are abstractions of the tree representing entities beyond what HTML does. For example, HTML doesn't have an element describing the first letter or line of a paragraph, or the marker of a list. Pseudo-elements represent these entities and allow CSS rules to be associated with them. that way, these entitities can be styled independently.