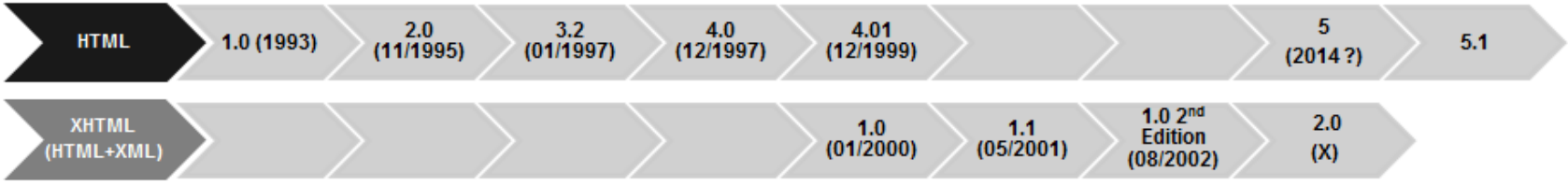
HyperText Markup Language

pdf

HTML : A bit of history



- Initial version created by Tim Berners-Lee in 1989
  - as an open language, royalty-free
  - Then developed by the World Wide Web Consortium (W3C)
  - Now developed by W3C and WHATWG
- Several versions
  - HTML 4 Strict, Transitional, Frameset
  - HTML vs. XHTML
  - HTML 5 (HTML.next, HTML 5.1)

HTML 5 : The language

- 1 language, 2 syntaxes
  - **HTML**, identified by documents of type `text/html`
  - **XHTML** (XML), identified by `application/xhtml+xml`
  - Similar syntaxes but different processing (e.g. +/- strict)
- Text-based
  - mix of **tags** (markup) and text
  - no compilation step
  - can easily view the source code
- Presentation agnostic
  - Might be rendered by different renderers (screen, printer, text-only, speech, . . .)
  - Rendering can be configured via **CSS**
- Basic Interactivity (navigation, forms)
  - Advanced Interactivity to be provided by **JS**
- Associated with a tree representation and JS APIs : **DOM**

HTML 5 : by example

Go into https://jsfiddle.net/ :

- experiment with tags html, head, body, h1, p, hr, br, a, div, span, table, img. . .
- experiment with attributes src, href, width, style. . .
- experiment with CSS

HTML 5 : Tags

- start (opening) tag :

```
<mytag>
```

- end (closing) tag :

```
</mytag>
```

- Tags should be closed
  - in XML-compatible syntax : always
    - in particular with self-closing tags :

    `<mytag/>`
  - in non-XML syntax : most of the time
    - except for some tags (historical reasons) : `img`, `br`, `input`, `...`
  - must be closed in the right order

```
<a><b></a></b> // wrong
<a><b></b></a> // correct
```

- Tags structure the content of an HTML document into a tree : the DOM Tree

HTML 5 : Attribute

An attribute indicates a property of a DOM element

- specified on the corresponding start tag or self-closing tag

```
<mytag property-name='property-value'></mytag>
<mytag property-name='property-value'/>
```

- Using quotes " or single-quotes ’

```
<mytag name="value"></mytag>
<mytag name='value'/>
```

- Possibly with nested quotes

```
<mytag name="value with 'inside'">
<mytag name='value with "inside"'>
```

- Aternate HTML 5 syntaxes (not XML-compatible)

HTML 5 : Attributes

Multiple attributes can be specified :

- space separated

```
<mytag attr1="value1" attr2="value2">
```

- order is not important

```
<mytag attr2="value2" attr1="value1">
```

- cannot duplicate the same attribute twice

```
<mytag attr1="value2" attr1="value1">
```

HTML 5 : A large standard

Defines many tags

- Paragraphs, Tables, Forms
- Multimedia : images, videos, audios
- Graphical Primitives
- . . .

Defines JavaScript APIs

- Basic document manipulations
- Element-specific APIs (e.g. video)
- Advanced APIs (Offline Storage, Database, communications)
- . . .

Defines how to integrate with other Web technologies

- Mix of SVG and MathML within the HTML page

HTML 5 : Demos

- Demos
- Mozilla's

HTML 5 : Tags and Content Model

Tags are organized in different categories

- Root & Metadata : `html`, `head`, `title`, `meta`, `link`, `base`, `style`
- Scripting : `noscript`, `script`, `template`
- Sections : `body`, `section`, `nav`, `article`, `h1...h6`, `header`, `footer`, `address`, `main`
- Grouping : `div`, `p`, `hr`, `pre`, `blockquote`, `ol`, `ul`, `li`, `dl`, `dt`, `dd`,`figure`, `figcaption`
- Text : `a`, `em`, `strong`, `small`, `s`, `cite`, `q`, `dfn`, `abbr`, `data`, `time`, `code`, `var`, `samp`, `kbd`, `sub`, `sup`, `i`, `b`, `u`, `mark`, `ruby`, `rt`, `rp`, `bdi`, `bdo`, `br`, `wbr`
- Media elements : `img`, `video`, `audio`, `source`, `track`, `canvas`, `svg`
- Embedded content : `iframe`, `embed`, `object`, `param`, `map`, `area`, `math`
- Forms : `form`, `fieldset`, `label`, `legend`, `input`, `button`,

HTML 5 Hello World !

As simple as that !

```
Hello World!
```

Browser's parsing algorithm are very robust (tag soup)

- Will create the page structure for you !
- Will try to close tags for you !
- ...

HTML 5 Basic page structure

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>This is the title</title>
</head>
<!-- this is a comment -->
<body>
<!-- visible content goes here -->
</body>
</html>
```

HTML 5 header

- The header of a document is delimited by the `head` tags.

```
<head> ... </head>
```

- The header contains meta-informations about the document, such as its title, encoding, associated files, etc.
- Some common items are :
  - metadata
    - The character set of the page, usually at the very beginning of the header (not reliable)

```
<meta http-equiv="Content-Type" content="text/html; charset
<meta charset="UTF-8">
```

  - The title of the page, displayed in the title bar of Web browsers.

```
<title>My great website</title>
```

  - Javascript & CSS links

HTML 5 body

- The content of the document is delimited by the `body` tags.

```
<body> ... </body>
```

- The body may be structured into sections, paragraphs, lists, etc.

HTML 5 body content

- Typically uses tags describe sections, by decreasing order of importance :

```
<h1>Title of the page</h1>
<h2>Title of a main section</h2>
<h3>Title of a subsection</h3>
<h4>Title of a subsubsection</h4>
```

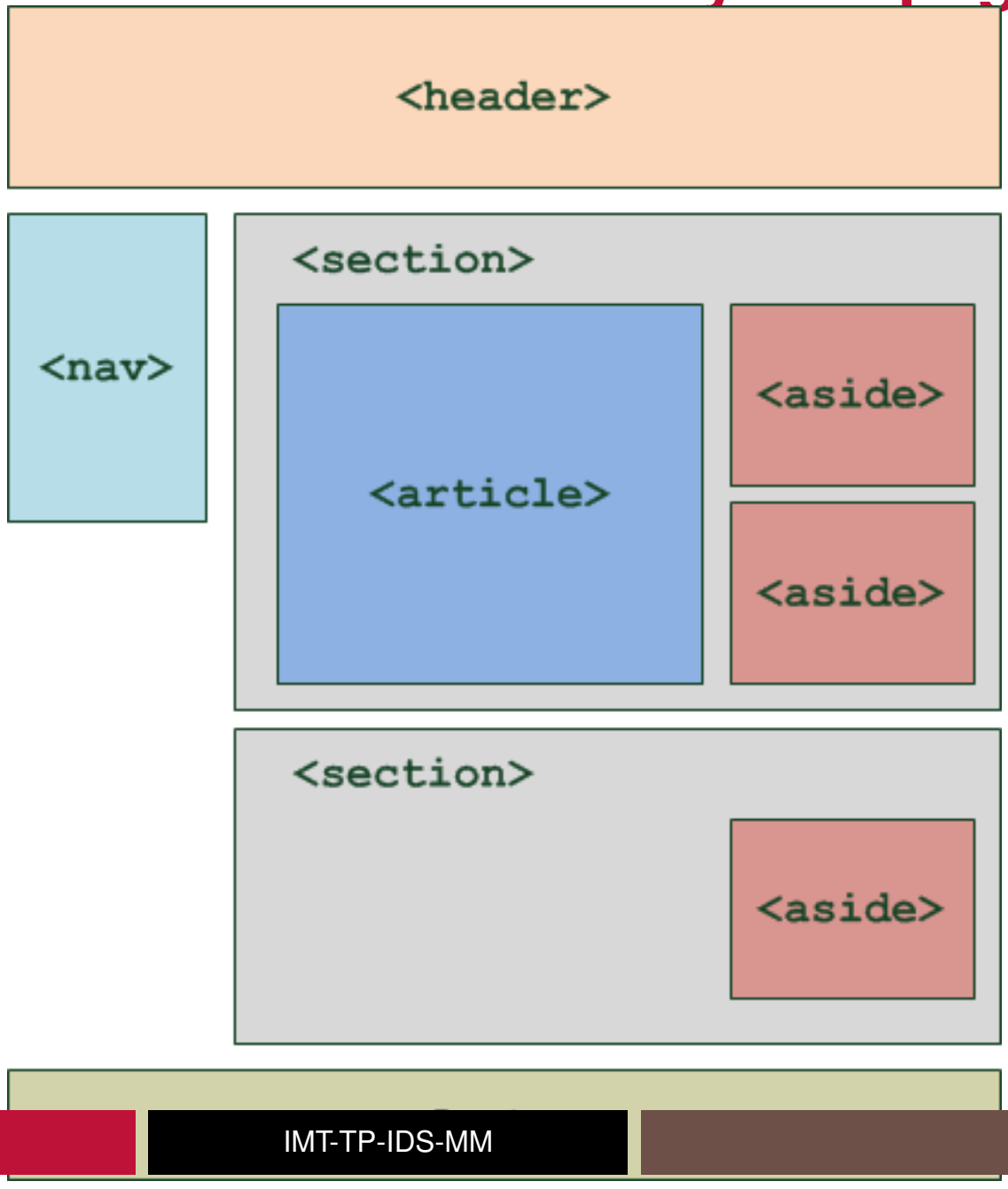
- Or paragraphs of text :

```
<p> ... </p>
```

- Or simple grouping elements without semantics :

```
<div> ... </div>
```

Structured body of a page



Links

- What differentiates Web pages (hypertext pages) from normal documents : links !
- Introduced with `<a> ... </a>`
- Navigating a link can bring to :
  - a resource on another server or another file of the same server

```
<a href="http://www.cnrs.fr/">
  <img src="images/cnrs.gif" alt="CNRS">
</a>
<a href="bio/indexbioinfo.html">Bioinformatics</a>
```

- another part of the same document with anchors

Relative URLs

- with respect to a context (e.g., the URL of the parent document, the **base URL**) :
- If context is : `https://www.example.com/toto/toto2/toto3`

relative URL	Absolute URL
/titi	https ://www.example.com/titi
tata	https ://www.example.com/toto/toto2/tata
#tutu	https ://www.example.com/toto/toto2/toto3#tutu

Anchors

- Anchors serve to reach a precise point in the document.
- They are defined, either on an existing tag by using the `id` attribute, or with an

```
<a name="tutorials">
```

- Then, one can link to this anchor :

```
<a href="#tutorials">tutorials</a>
<a href="http://www.w3.org/#tutorials">tutorials</a>
```

Lists

- Unordered lists
  First bullet point
  Second bullet point
- Ordered lists
  First ordered point
  Second ordered point

Tables

row 1 - column 1

row 1 - column 2

row 2 - column 1

row 2 - column 2

Other options : `th`, `caption`, `thead`, `tbody`, `tfoot`, `col`, `colgroup`

Forms

Information First name : Last name :

Password : Male Female I have a bike I have a car Date : Nationality French English

Other options : colors, time, . . .

Nested documents

- Render the content of another page in the current page
- Using `<iframe>` tags

```
<iframe width="400" height="215" frameborder="0
            scrolling="yes" marginheight="0" marginwidth="0"
            src="http://www.telecom-paris.fr/...">
</iframe>
```

Document Object Model

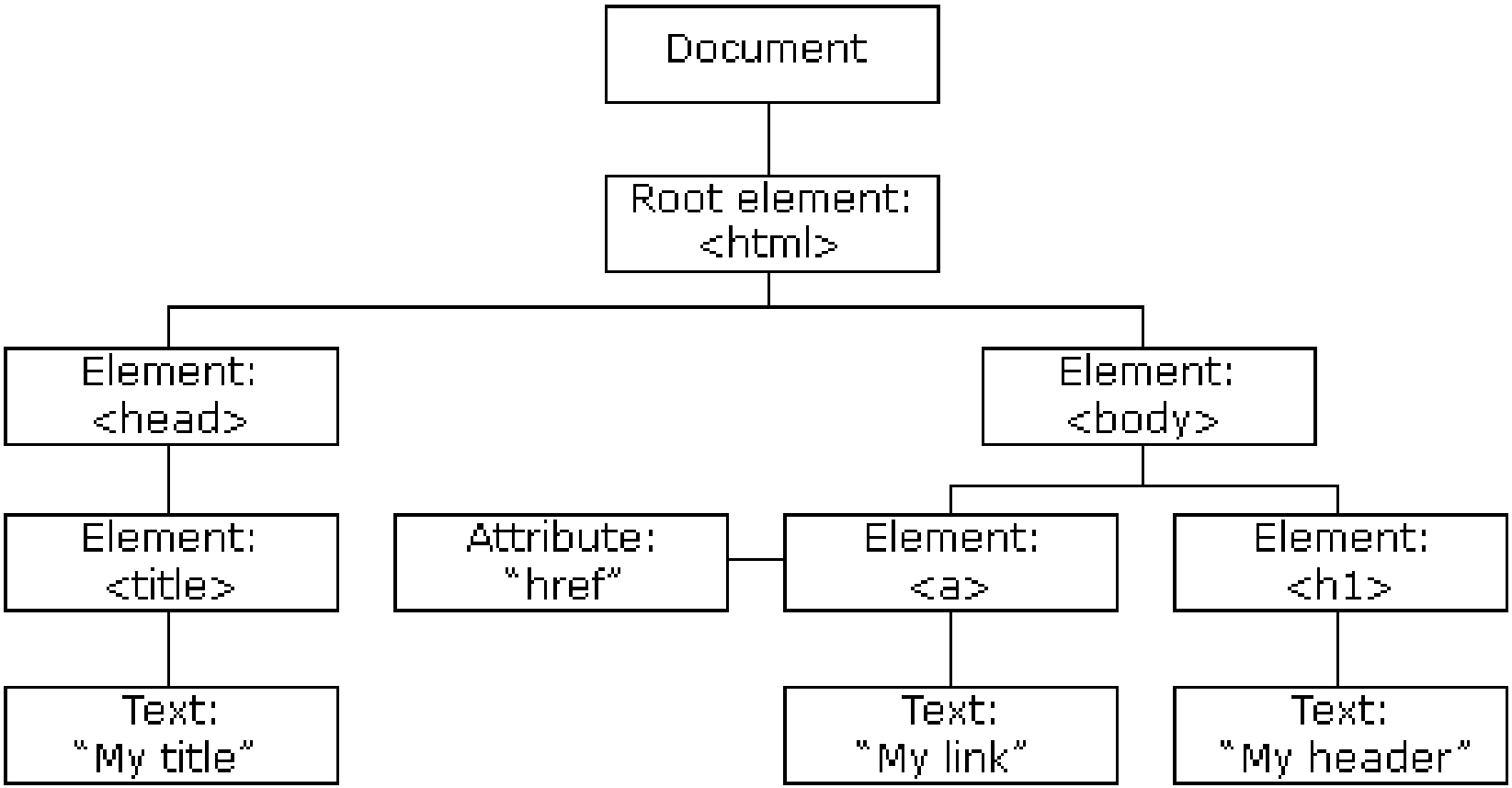
Tree-based representation of an HTML document

DOM Node=

- DOM Text node
- DOM Comment node
- DOM Element
- DOM Attribute

DOM Nodes, DOM Elements . . . can be manipulated by script via specific interfaces

DOM Tree example




```
<html>
 <head>
  <title>My title</title>
 </head>
```

Summary of this lesson

- HTML history, HTML5, tags, attributes, JS APIs
- demo, basic structure, basic elements
- DOM = JS API to HTML