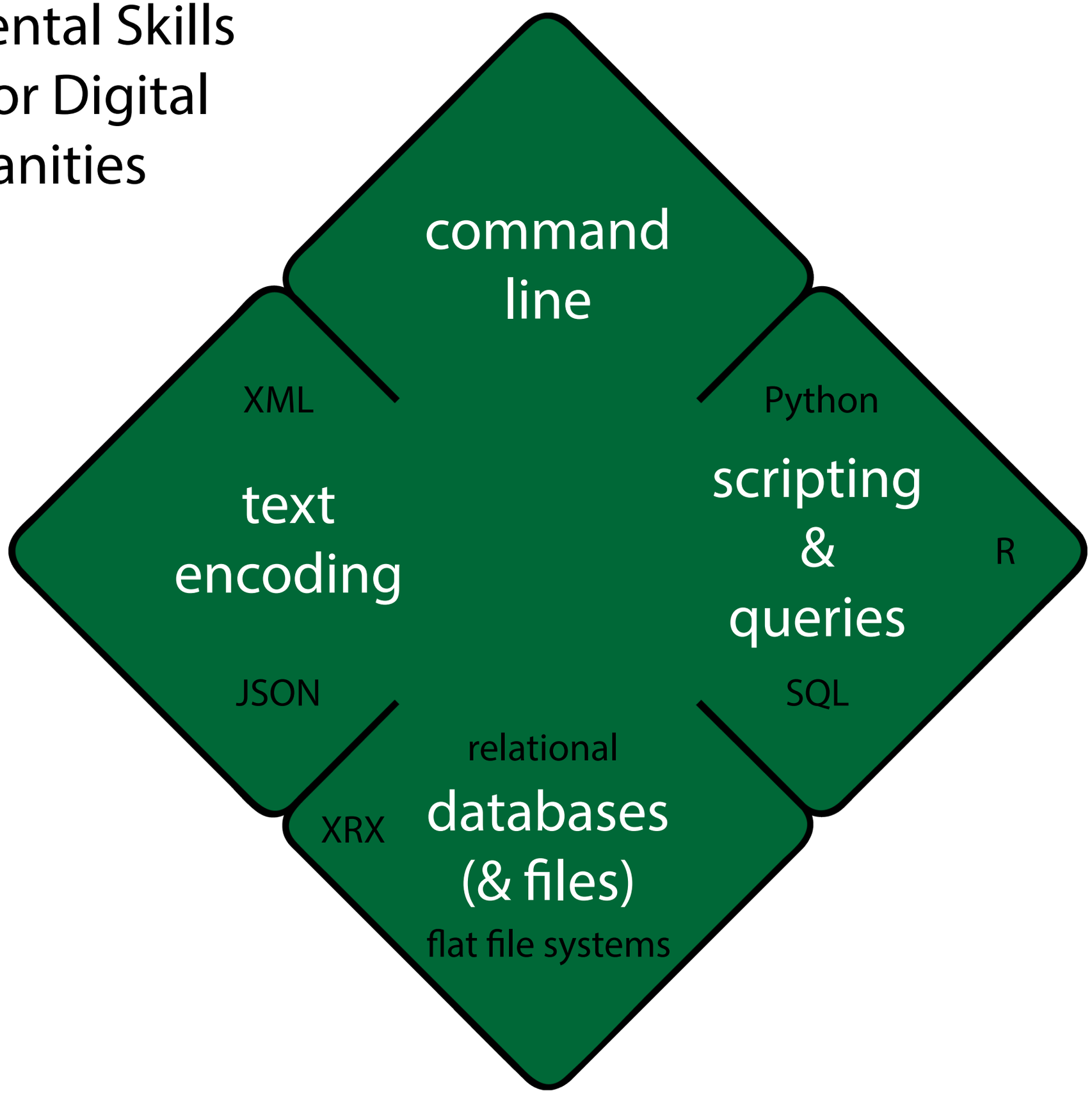
Introduction to XML
for Literary and Historical Research,
Part 1: Basics of XML and HTML

2017 April 13 and 20
Thursdays
12pm to 3pm
D-Lab, 356 Barrows

Scott Paul McGinnis
updated 20170420

Fundamental Skills Useful for Digital Humanities



What is XML?

- eXtensible Markup Language
- a simple set of rules for encoding text or data that can be used in many ways
- a group of closely related languages or “applications”

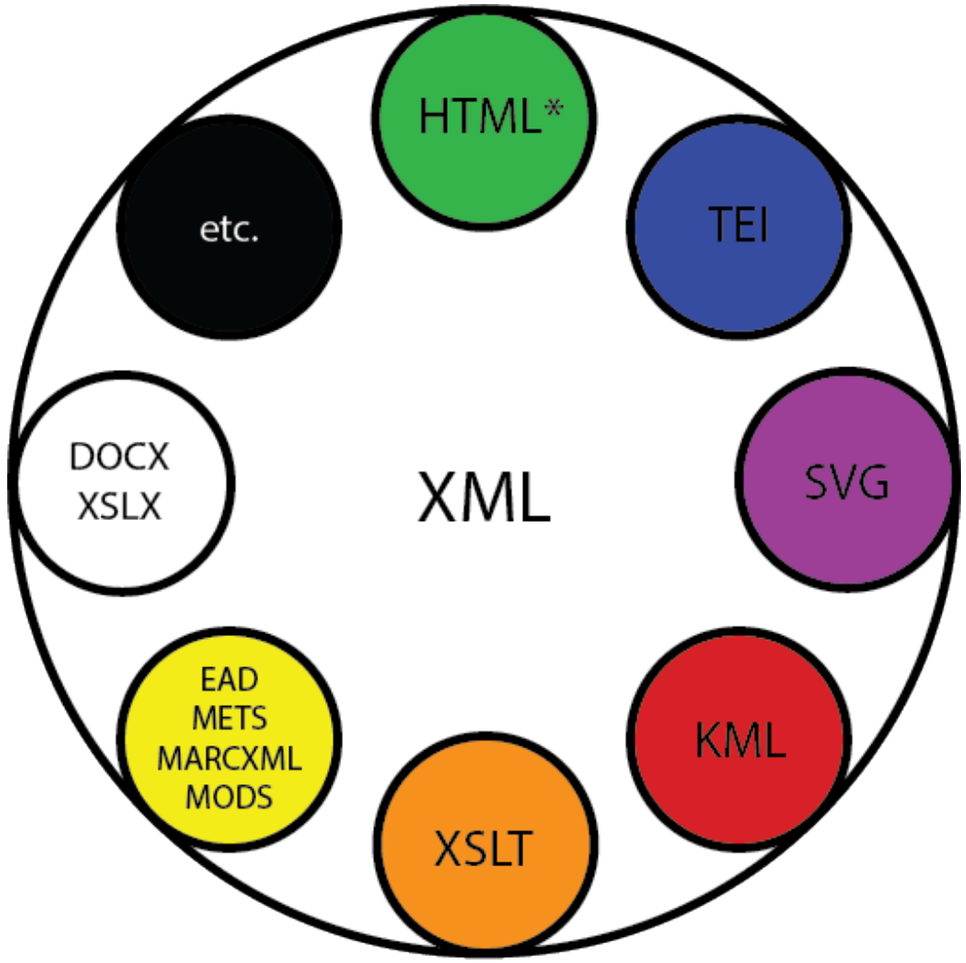
Why Use It?

- ubiquitous
- flexible
- relative ease
- human readable*
- made for text
- working with text or data the web
- maintained by a large and active community
- Unicode compliant

but ... XML alone doesn’t do very much

- Websites: HTML + CSS + Javascript/JQuery
- Wordpress: HTML + CSS + PHP
- XRX Databases: XML + Restful APIs + xQuery
- Web-scraping: HTML + scripting (e.g. Python)
- APIs: XML + URL queries + scripting

Some XML Applications



XML in the Wild

- HTML Websites (dev. tools)
- DOCX Archive
- TEI Digital Edition

XML Basic Concepts 1: Elements and Content

The basic unit of XML is called an element. We might think of it like a container for our data that describes and categorizes it for the computer. When viewing a web page, for example, we don't see the HTML elements. But we can access and manipulate them in all sorts of ways, through CSS, JQuery, XQuery, and other programming languages.

Here is an example using some custom XML:

This is an XML Element. It has two components, the Start-Tag and the End-Tag.

Notice the use and order of the special characters:

< > /

```
<event>
    <name>XML Workshop</name>
    <location abbr="UCB">UC Berkeley</location>
    <year>2016</year>
    <desc>A workshop on XML and its applications.</desc>
</event>
```

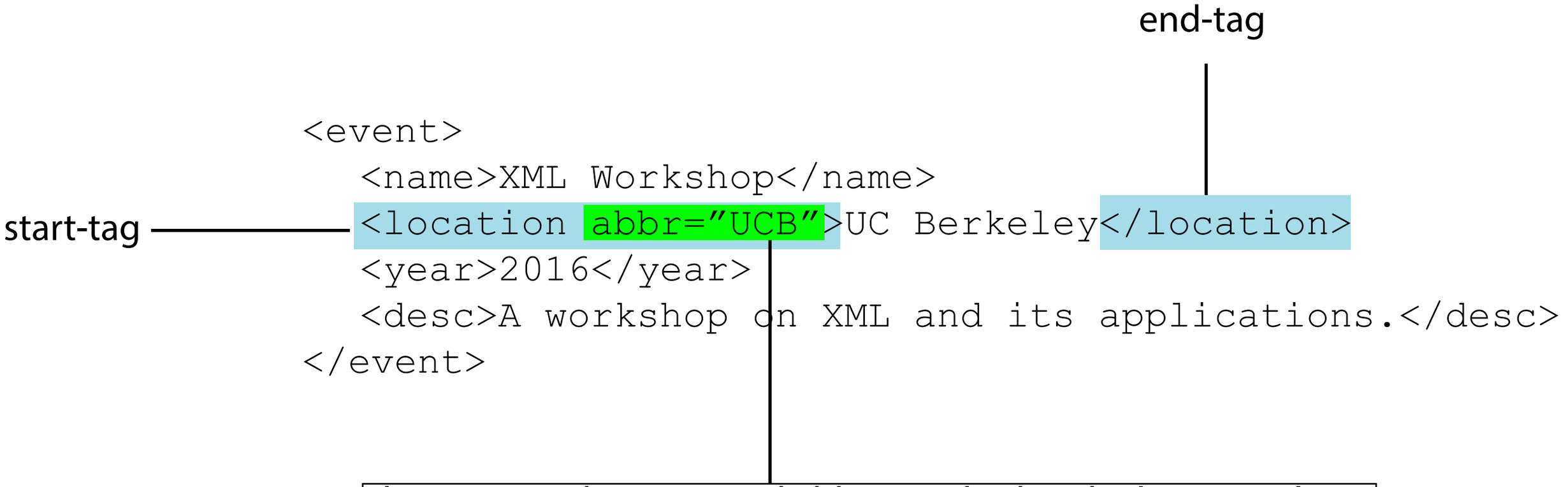
All of this stuff is nested (contained) within the <event> element.

Elements may contain other elements, and they may contain Content, such as the prose description of the workshop seen here.

XML Basic Concepts 2: Attributes and Values

Attributes and their values classify specific instances of an element.

Consider our custom XML:



XML Basic Concepts 3: Well-formedness and Validity

Well-formedness

A well-formed XML document follows these rules:

- It has at least one XML element.
- It has a single root element, which contains all elements and content and which is not contained by any other element.
- All elements are properly closed, with a start tag and an end tag, or as a self-closing element.
- All elements are properly nested. If it opens inside another element, it must close inside it too.
- All elements and attributes obey the proper syntax (e.g. no missing quotation marks). No element my have the same attribute more than once.
- Case sensitive
- (Other, more technical rules, which we don't have to worry about.)

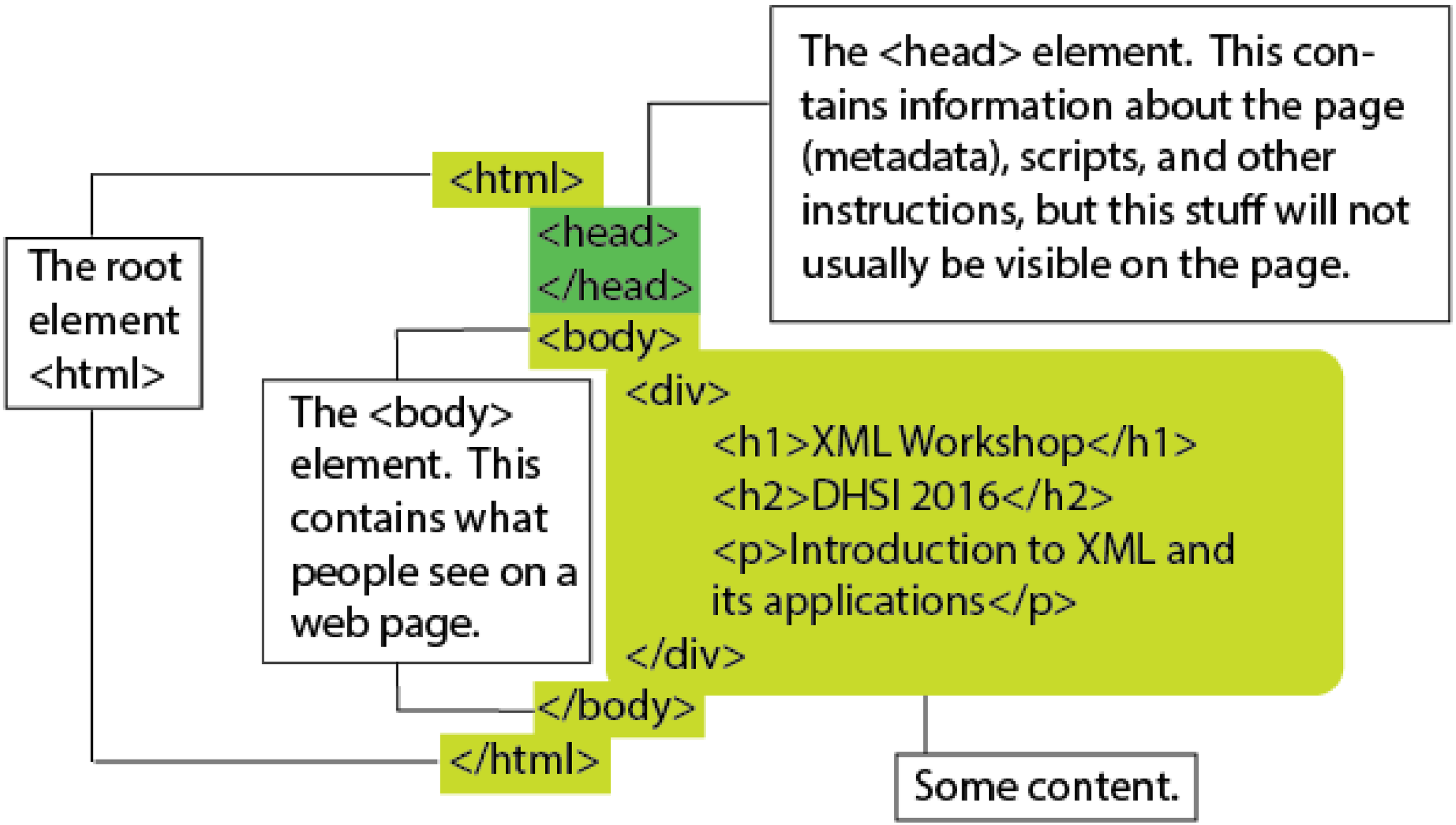
In class: Well-formed Exercise

Validity

Validation rules are additional requirements of a specific XML application.

- Controlled vocabulary for element and attribute names and for content
- Data-type Restrictions
- Controlled hierarchies
- Document Type Declaration (DTD) or Schema

Ex: Valid HTML

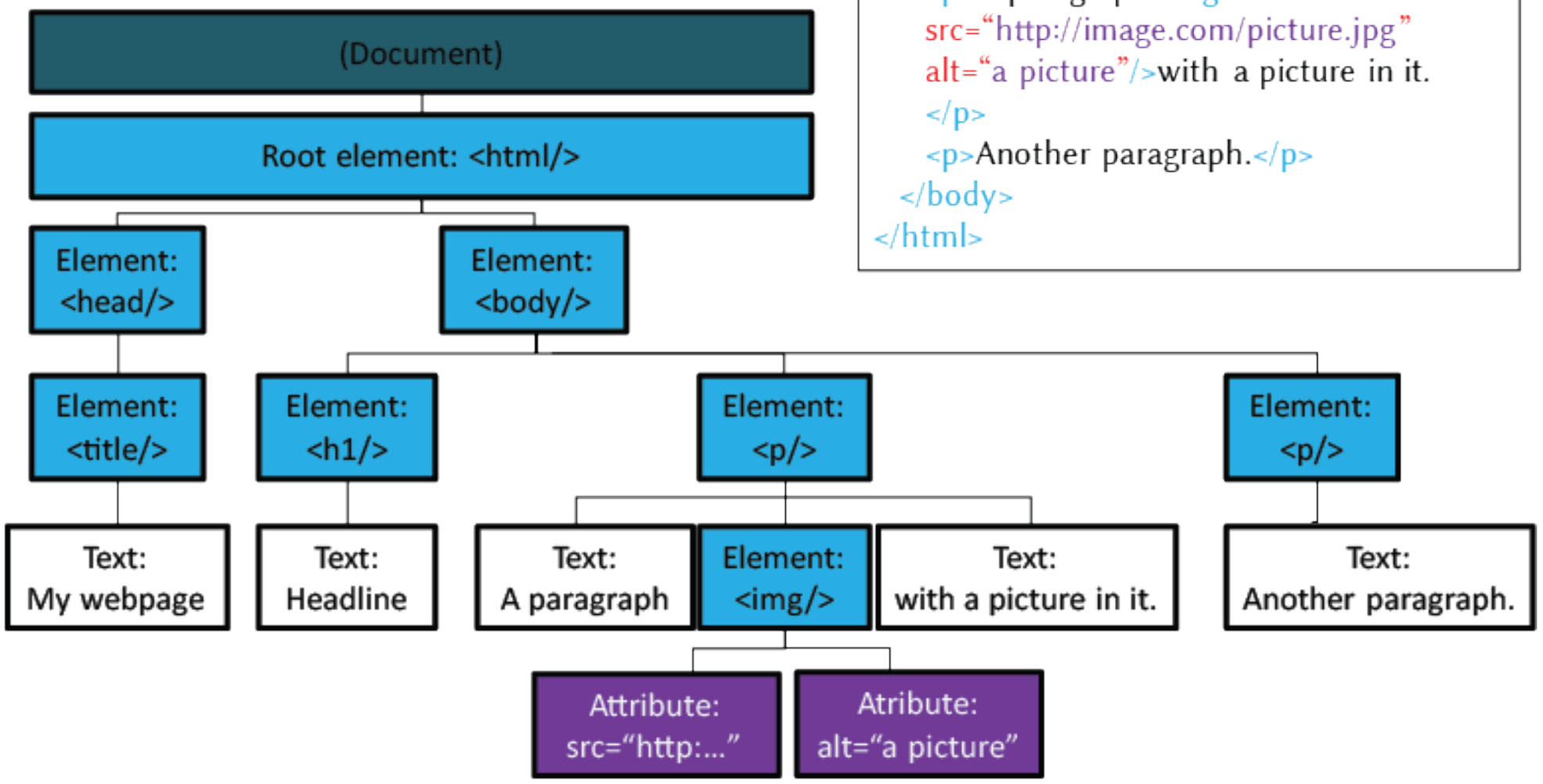


The XML Tree

- “Document Object Model” (DOM)

A mixed metaphor:
- Nodes and leaves
- Parents, Children, Siblings, etc.

```
<html>
  <head>
    <title>My Webpage</title>
  </head>
  <body>
    <h1>Headline</h1>
    <p>A paragraph<img
    src=“http://image.com/picture.jpg”
    alt=“a picture”/>with a picture in it.
    </p>
    <p>Another paragraph.</p>
  </body>
</html>
```



Pirates Exercise: Getting Started

1. Create a directory (folder) in a convenient place on your computer, and call it XML-workshop.

2. Copy the following files from the provided thumb drive and save them to your XML-workshop folder:

- pirates-instructions.pdf
- template.html
- style.css
- info.txt
- The CaseFiles directory and all of its contents

3) Download and install the Atom text editor. For Windows, go to atom.io. For others, installation instructions can be found here: https://atom.io/docs/v0.194.0/getting-started-installing-atom.

Some cautionary notes:

- Please do not store your project in a cloud-based folder like Google Drive, it won't work properly.
- Please do not rename any of the files. (Case-sensitive)
- Please do not copy-paste from the instructions pdf. The formatting will cause problems.

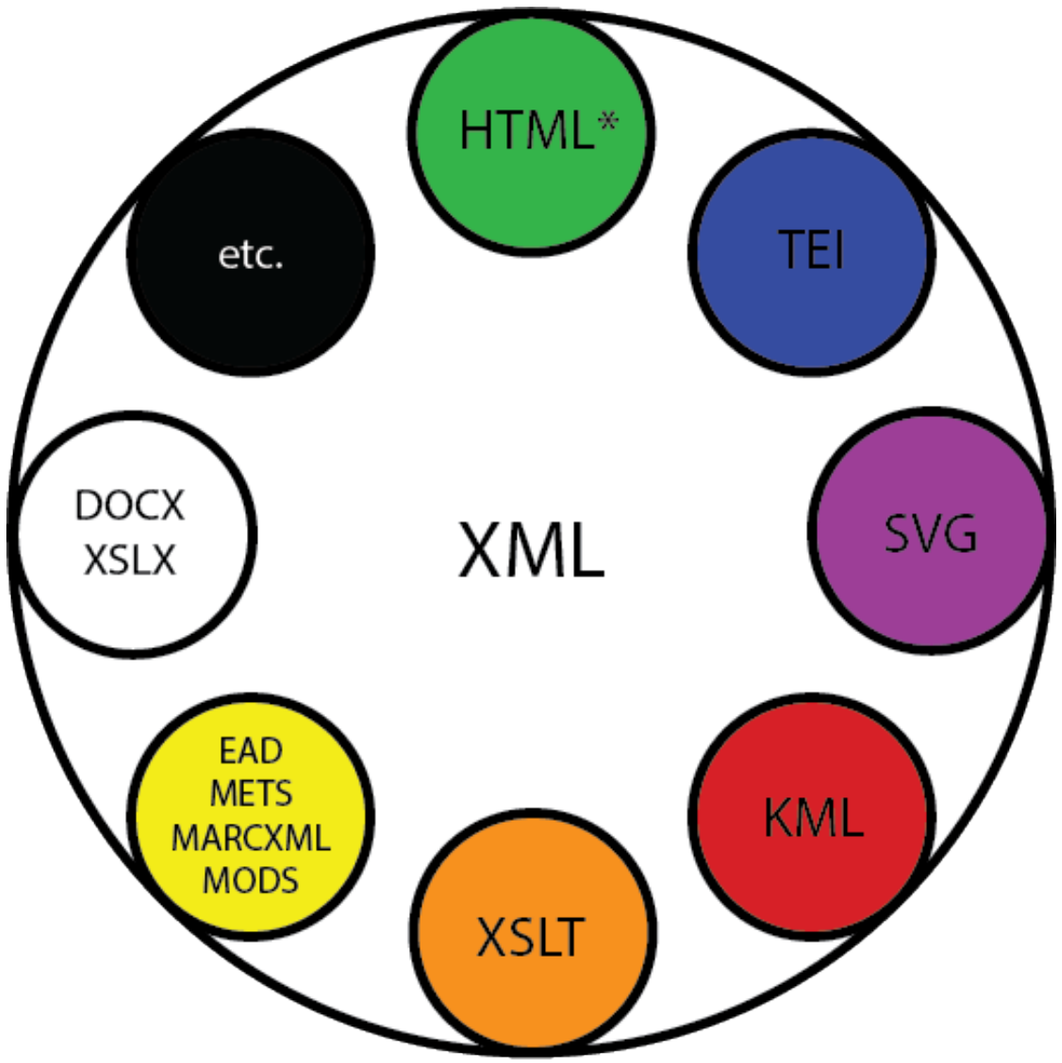
Introduction to XML
for Literary and Historical Research
Part 2: TEI and KML

2017 April 13 and 20,
Thursdays
12pm to 3pm
D-Lab, 356 Barrows

Scott Paul McGinnis

Review

- Elements and their Content
- Attributes and their Values
- Well-formedness & Validity
- The XML Tree Structure
- XML Applications



More XML...

1.XML/HTML comments `<!-- -->`

```
<!--  A comment goes here.  Anything
in this tag is ignored by the XML
processor or web browser. -->
<!-- Comments exist in all
programming languages, but are often
written differently.  -->
```

In CSS, a comment `/*looks like this*/`
In Python `# this is a comment`

2.Character Data `<![CDATA[]]>`

Another way to tell the processor to ignore this text, but in this case, the text is passed through. These are useful for displaying XML without having that XML processed.

```
<![CDATA[ Just about anything can
go in here, even <xml/> and the
processor will treat it as plain
text. ]]>
```

Text Encoding Initiative (TEI)
tei-c.org

About the TEI

- text encoding
- multiple versions (schemas) + customization
- highly interpretive

In class: how to navigate the guidelines

The TEI Header

A section for bibliography and metadata, with five parts:

```
<fileDesc/>
<encodingDesc/>
<profileDesc/>
<xenoData/>
<revisionDesc/>
```

- see the guidelines for explanation

Example TEI

Here are the main parts of a typical TEI document:

```
<TEI>
   <teiHeader>
      <fileDesc></fileDesc>
      <encodingDesc></encodingDesc>
      <profileDesc></profileDesc>
      <xenoData></xenoData>
      <revisionDesc></revisionDesc>
   </teiHeader>
   <text>
      <front></front>
      <body></body>
      <back></back>
   </text>
</TEI>
```

In class: review the tree structure

Exercise 1: Reading the Header

In pairs or small groups,

1. Open one of the Perseus files in your text editor (Atom, etc.).

2. Read through the `<teiHeader/>` section of the file. Using what you know about XML and the information given there, try to determine what you can about the text, its provenance, editorial practices, and so on.

3. Share with the class what you found out about your text. Discuss any issues or questions that came up.

TEI Exercise 2: Encoding a Document

(less technical version)

1. Get a printout of one of the Old Bailey Pirates cases.

2. Identify what you think are the salient features of the text, and annotate the printout accordingly. Think about structural features, such as paragraphs and page breaks, as well as important features of the content.

3. Compare with the Old Bailey TEI (found in the CaseFiles folder). What did the Old Bailey editors identify as relevant? What interpretive issues are at stake? Consult the TEI Guidelines (tei-c.org) for clarity about their choices.

(more technical version)

1. Get a printout of one of the Old Bailey Pirates cases.

2. Identify what you think are the salient features of the text, and annotate the printout accordingly. Think about structural features, such as paragraphs and page breaks, as well as important features of the content.

3. Chose five or six of the most important features, and use the TEI guidelines to decide what elements to use.

4. Compare with the Old Bailey TEI (found in the CaseFiles folder). What did the Old Bailey editors identify as relevant? What interpretive issues are at stake? Consult the TEI Guidelines (tei-c.org) for clarity about their choices.

Overview of Geospatial Tools and Skills

ArcGIS and QGIS
- powerful
- good with large data sets
- many resources available, especially robust datasets
- but... not as good the farther you go back in time
- somewhat steep learning curve

Google Earth
- dynamic views
- 3D modeling
- Website (requires a browser plugin)
- Requires web design and javascript / jquery

Google Maps and Leaflet leafletjs.com
- dynamic views example
- websites (no plugin required)
- requires some knowledge of web design and javascript / jquery

Python Libraries / R Packages

Gephi gephi.org/
- Network visualization tool with map views example
- data as nodes and edges

Carto DB
- Integrated CMS for Mapping
- Javascript Library + SQL database + Special CSS Rules

Cartography Skills
- geo-referencing / Geo-rectification
- map projections
- coordinate systems

Computer Languages and the Web
- Spreadsheets and SQL (a language for querying databases)
- XML, KML, and HTML (markup and text-encoding languages)
- javascript and/or jquery (dynamic web)
- application programming interfaces (API)

Keyhole Markup Language (KML)

- XML for Geographical Data
- Google Maps, Google Earth, Leaflet.js, etc.
- Data Visualized as Rasters and Vectors
- 3 Types of Vectors: Points, Lines, and Polygons
- Examples of Geographical Features and their Visiualizations

Feature	Representation
Rivers	Lines
Roads (centerlines)	Lines
Vegetation	Polygons
Cities	Points
Urban Areas	Polygons
Administrative Boundaries	Polygons
Satillite Imagery	Rasters
Census Tracts	Polygons

Basic KML

Common Elements

```
<Point></Point>

<LineString></LineString>

<Polygon></Polygon>

<Placemark></Placemark>

<coordinates></coordinates>
```

note: (Lat., Long) vs. (x, y, z)

```
<![CDATA[…]]>
```

(Case Sensitive)

Example KML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>US States and State Capitals</name>
    <description>
        <![CDATA[Some prose or code]]>
    </description>
    <Placemark>
       <name>Washington</name>
       <description>
           <![CDATA[Olympia]]>
       </description>
       <Point>
         <coordinates>
-122.905014,47.035805,0.000000
         </coordinates>
       </Point>
     </Placemark>
   </Document>
</kml>
```

KML Exercise: Getting Started

1. Create a new directory (folder) in a convenient place on your computer (but not in a cloud service like Google Drive). Call it XML-workshop-day2.

2. Download these files (from the location on the board) and save them to the same folder:

- pirates-maps.html
- pm-style.css
- holland.kml
- placemark-simple-template.kml
- locations.txt
- map-scripts.txt

3. Copy-Paste the CaseFiles folder and all of its contents into this folder. (Get the CaseFiles folder from me if you don't have it from last time.)