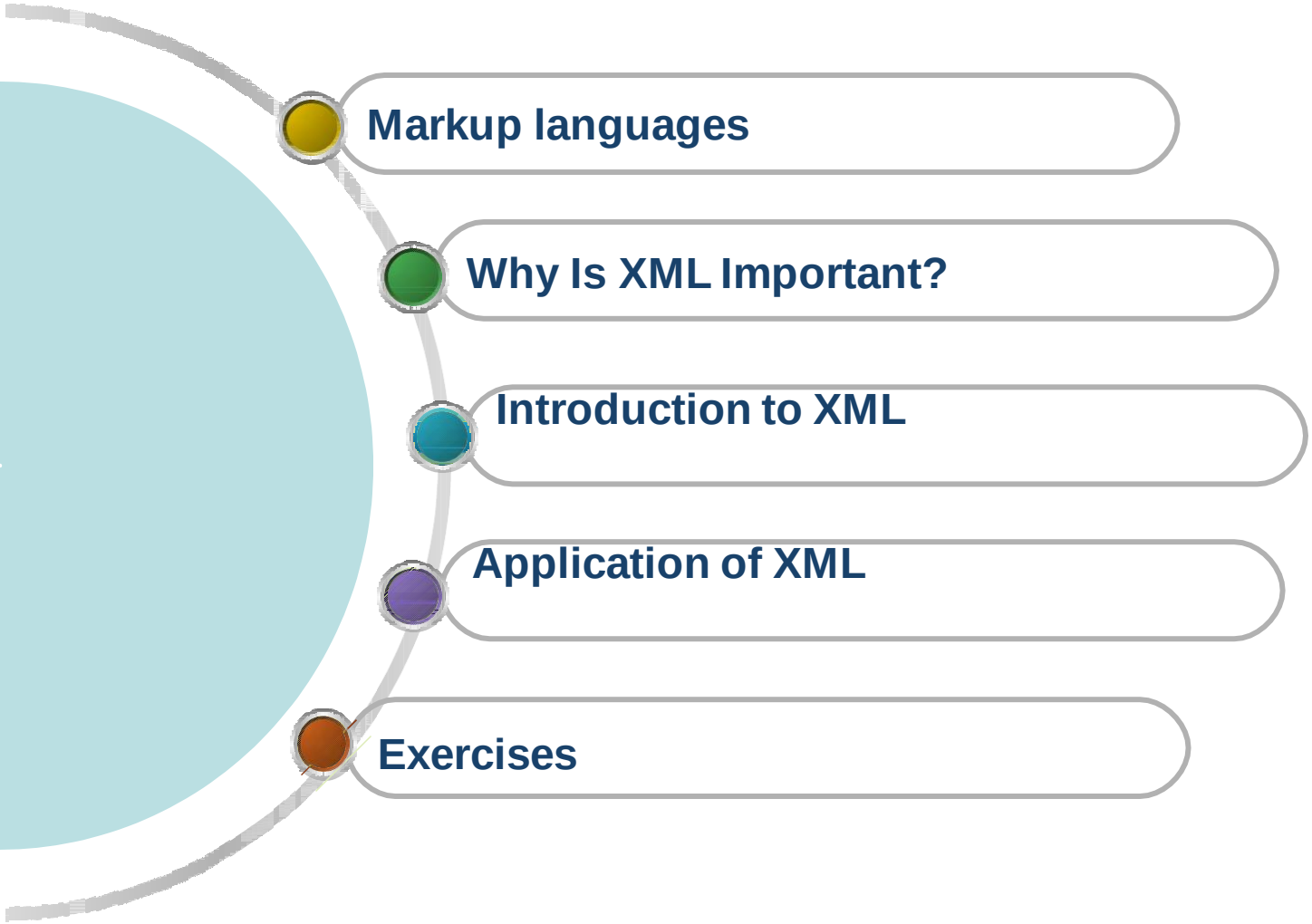
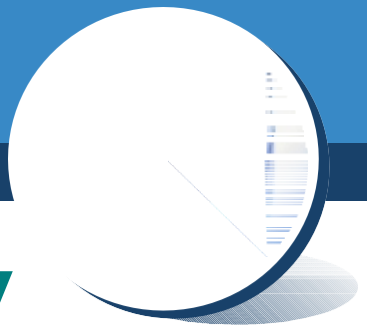
XML Introduction

XML

Contents



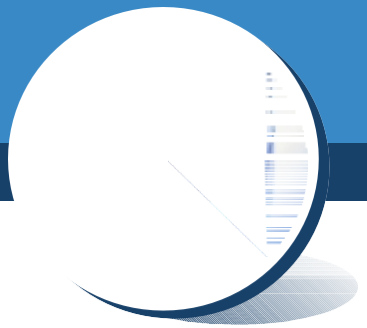
Markup languages



- **A markup language must specify**
  - What markup is allowed
  - What markup is required
  - How markup is to be distinguished from text
  - What the markup means

- **XML only specify the first three, the fourth is specified by DTD**

What Is XML?



- eXtensible Markup Language
- It is a text-based markup language
- XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data.

Difference between XML and HTML

XML was designed to carry data, not displaying data

- XML is not a replacement for HTML.

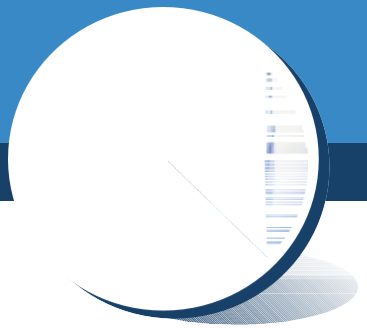
- Different goals:

  XML was designed to describe data and to focus on what data is.

  HTML was designed to display data and to focus on how data looks.

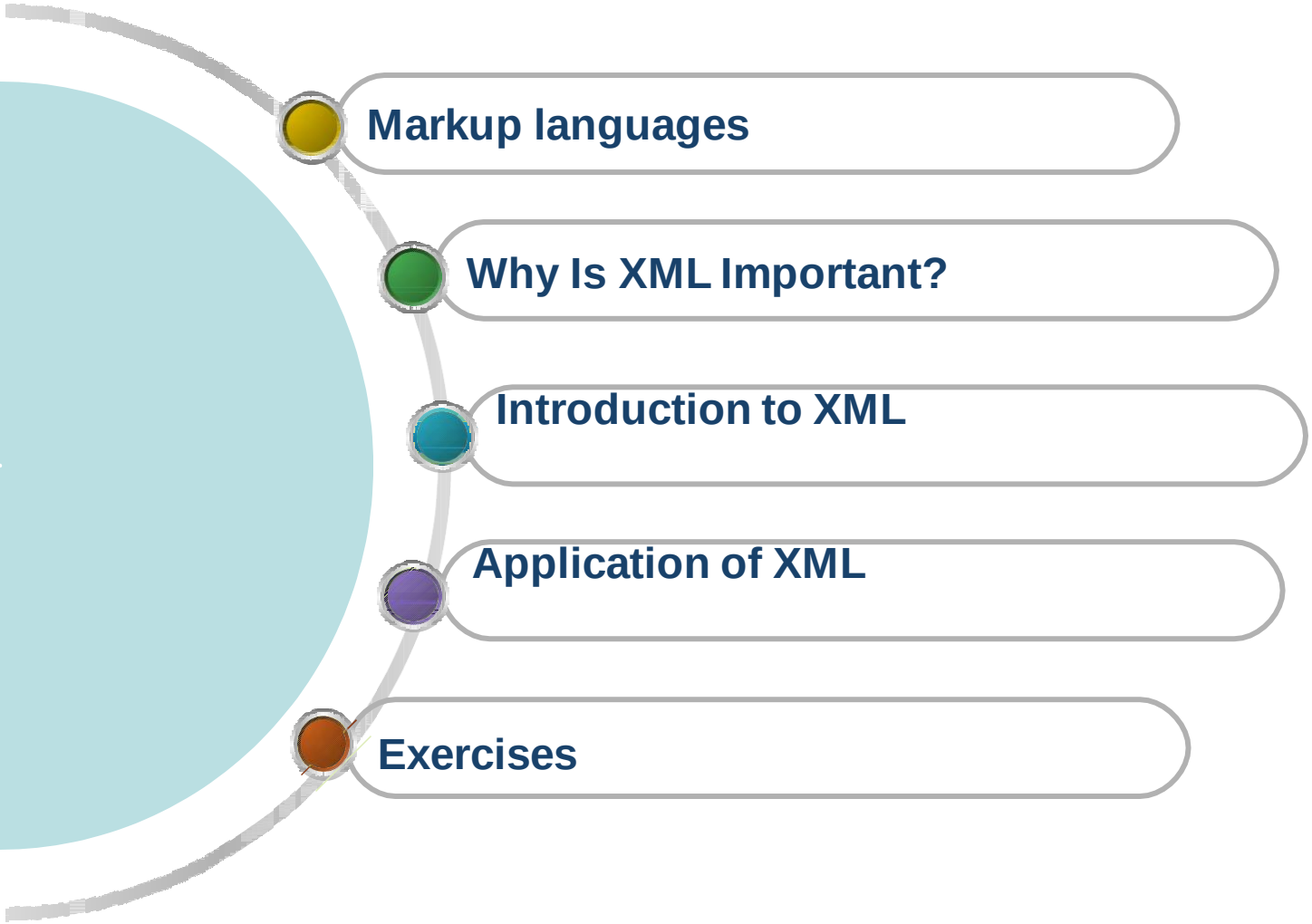
- HTML is about displaying information, XML is about describing information.

An example of XML

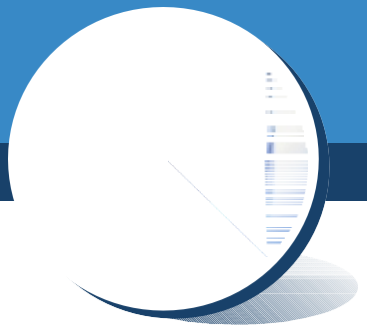


```
<?xml version="1.0"?>
<products>
   <product id="PRO001">
       <name>Coca Cola</name>
       <price>5000</price>
   </product>
   <product id="PRO002">
       <name>Pepsi</name>
       <price>4500</price>
   </product>
   <product id="PRO003">
       <name>Number One</name>
       <price>5500</price>
   </product>
</products>
```

Contents

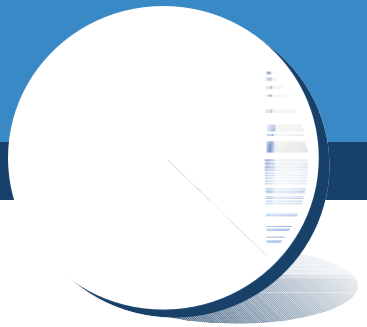


Why Is XML Important?



- **Plain Text**
  - Easy to edit
  - Platform independent
- **Data Identification**
  - Tell you what kind of data you have
  - Can be used in different ways by different applications
- **Easily Processed**

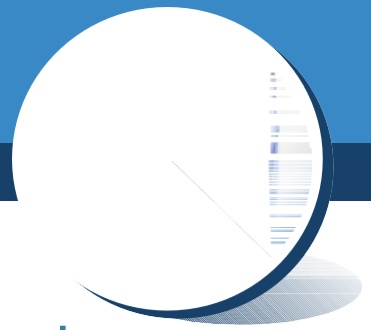
Why Is XML Important?



- **Stylability**
  - Inherently style-free
  - XSL---Extensible Stylesheet Language
  - Different XSL formats can then be used to display the same data in different ways

- **Hierarchical**
  - Faster to access
  - Easier to rearrange

XML Features



- **Extensible**

  XML allows you to create your own self-descriptive tags, or language, that suits your application.

- **Portable**

  XML enables separation of content and presentation, the content, or data, is **portable** across heterogeneous systems.

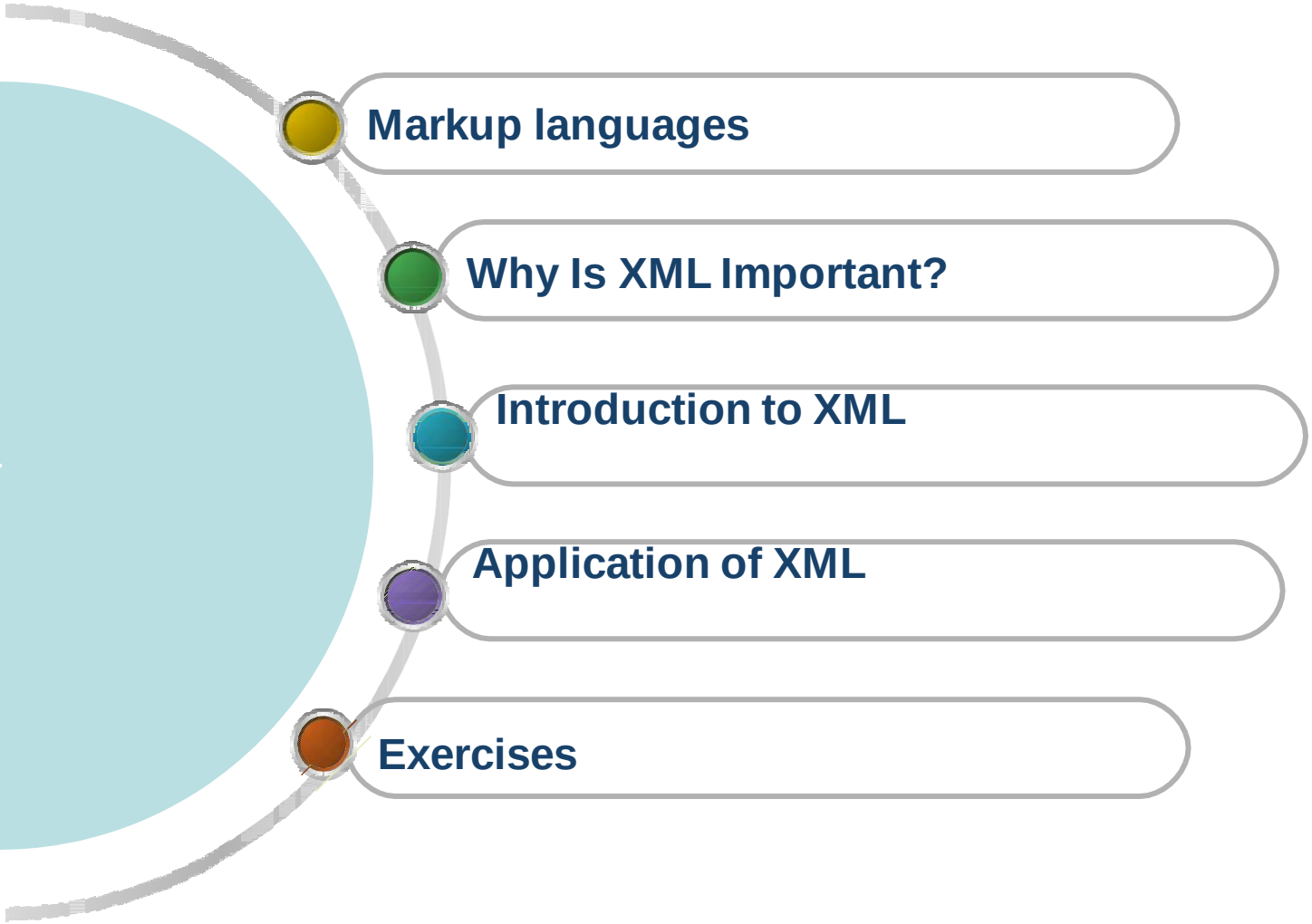
  XML enables document **portability**, that is, parties who use **XML** must agree to certain conditions. For example, in addition to agreeing to use **XML** for communicating, two applications must agree on the set of elements they will use and what those elements mean.

XML Features

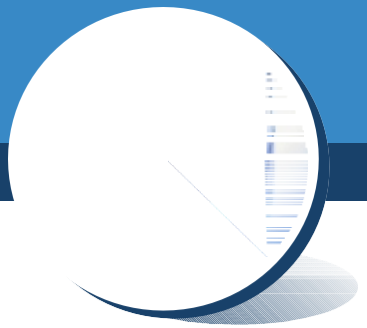
- **Descriptive**
  Since XML tags describes the meaning of data it contains therefor it is known as descriptive

- **Structured**
  XML document are represented with particular structure.

Contents



XML Building blocks



- PI (Processing Instruction)
- Tags
- Elements
- Content
- Attributes
- Entities
- Comments

XML Building blocks--Prolog

- The part of an XML document that precedes the XML data
- Includes
  - A declaration: version
  - An optional DTD (Document Type Definition)
- Example

```
<?xml version="1.0"?>
```

- The prolog of an XML document comprises everything from the start of the file to the document root tag.
- It may contain the XML declaration, processing instructions, comments, and a document type definition.

Tags

- *Tags* are used to specify a name for a given piece of information.
- A tag consists of opening and closing angular brackets (<>) that enclose the name of the tag.
- Example

```
<EMP_NAME>Nick Shaw</EMP_NAME>
```

Elements

- Elements are represented using tags.
- An XML document **must always** have a *root element*.
- General format:

```
<element> … </element>
```

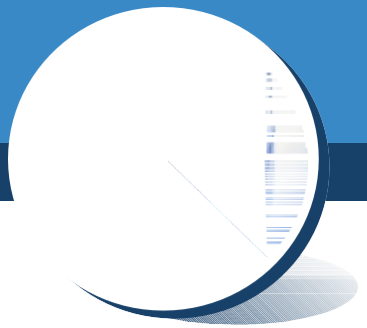
- Empty element:

```
<empty-Element />
```

- Example

```
<Authorname>John Smith</Authorname>
```

Content

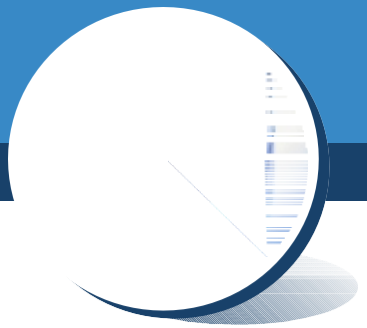


- *Content* refers to the information represented by the elements of an XML document.
  - Character or data content
  - Element content
  - Combination or mixed content

- Example

```
<BOOKNAME>The Painted
  House</BOOKNAME>
```

Attributes



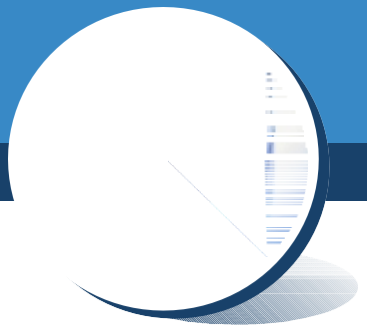
- Located in the start tag of elements
- Provide additional information about elements
- Often provide information that is not a part of data
- Must be enclosed in quotes
- Should I use an element or an attribute?

  metadata (data about data) should be stored as attributes, and that data itself should be stored as elements

Entities

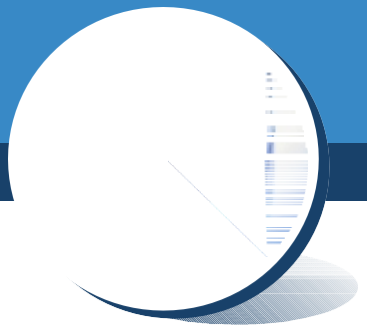
- An *entity* is a name that is associated with a block of data..
  - Entities: < , >…
  - Entities include & , name of entity and semicolon
  - It is a shortcut
  - General Entities
    - General entities are declared in *Document Type Definitions (DTD)*
    - *Example*
      - <! ENTITY nyt "The Times of India.">
      - For more information, please visit &nyt; Thank you!

Comments



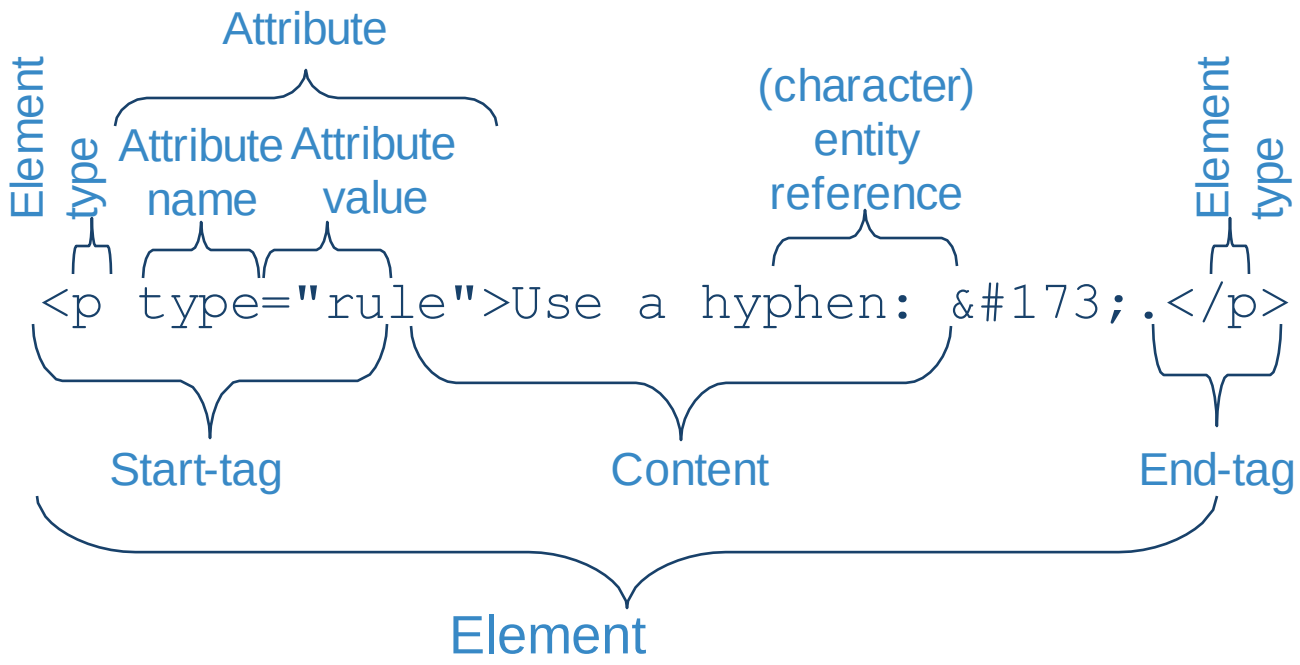
- *Comments* are statements used to explain the XML code.
- Example
  - **<!--PRODUCTDATA is the root element-->**

XML Syntax

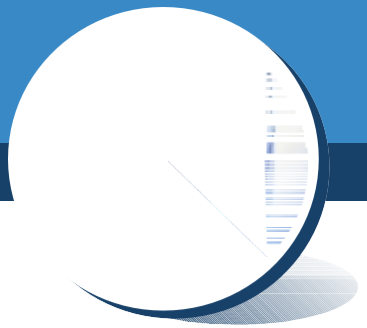


- All XML elements must have a closing tag
- XML tags are case sensitive
- All XML elements must be properly nested
- All XML documents must have a root tag
- Attribute values must always be quoted

Anatomy of an element



XML Validation



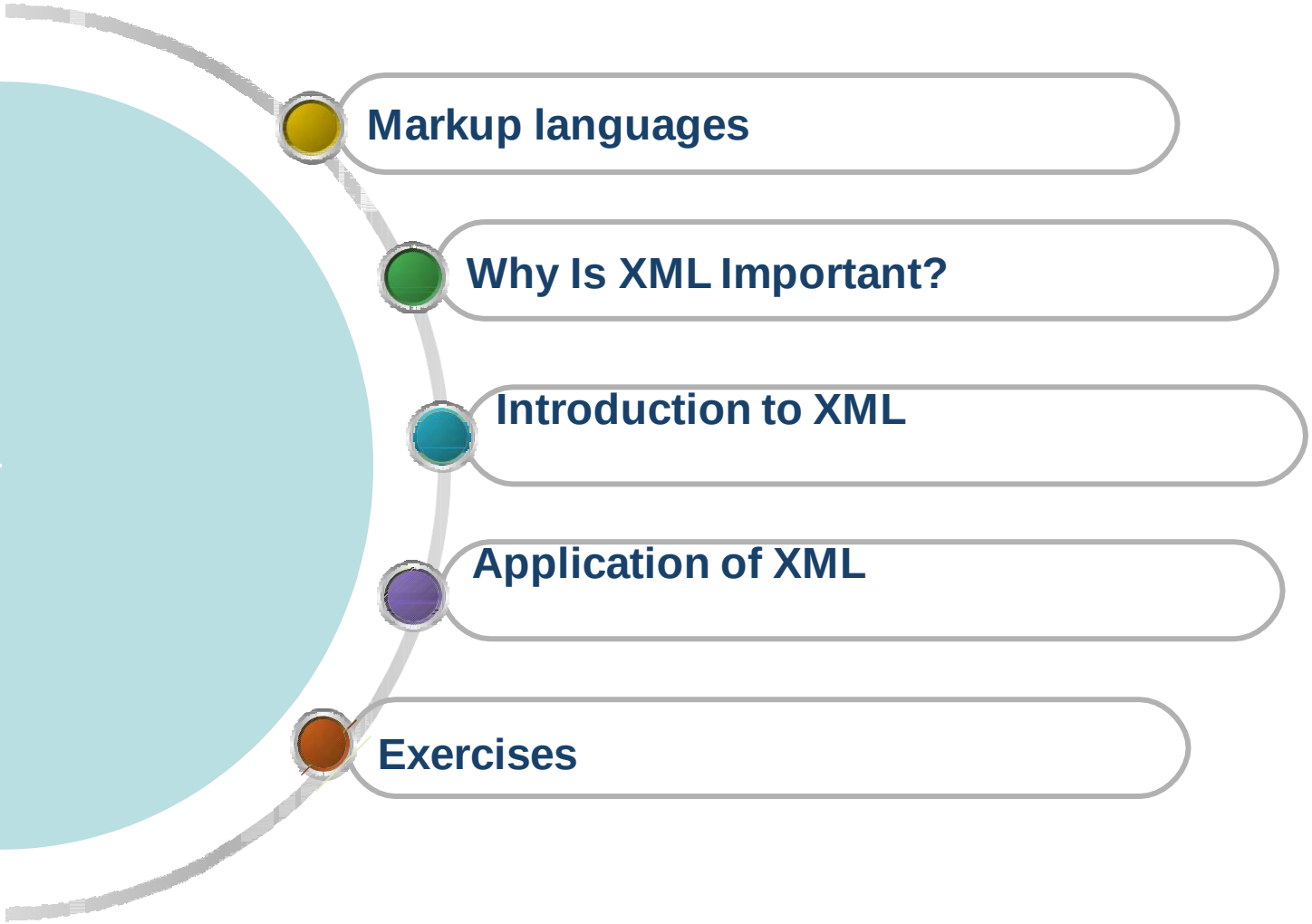
- **"Well Formed" XML document**
  --correct XML syntax
- **"Valid" XML document**
  - “well formed”
  - Conforms to the rules of a DTD (Document Type Definition)
- **XML DTD**
  - defines the legal building blocks of an XML document
  - Can be inline in XML or as an external reference
- **XML Schema**
  - an XML based alternative to DTD, more powerful
  - Support namespace and data types

Displaying XML

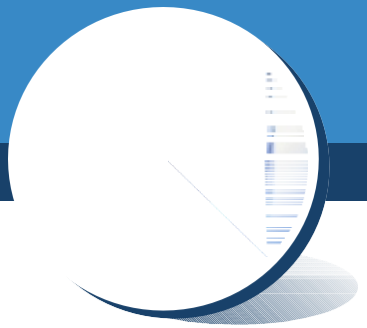
- **XML documents do not carry information about how to display the data**

- **We can add display information to XML with**
  - CSS (Cascading Style Sheets)
  - XSL (eXtensible Stylesheet Language) --- preferred

Contents


Markup languages
Why Is XML Important?
Introduction to XML
Application of XML
Exercises

XML Application1—Separate data



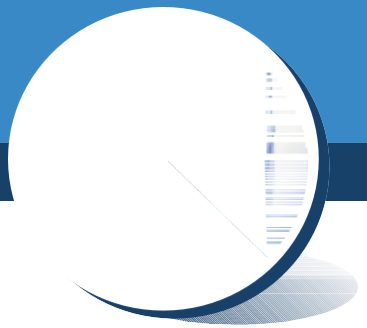
XML can Separate Data from HTML

- Store data in separate XML files
- Using HTML for layout and display
- Using Data Islands
- Data Islands can be bound to HTML elements

Benefits:

Changes in the underlying data will not require any changes to your HTML

XML Application2—Exchange data



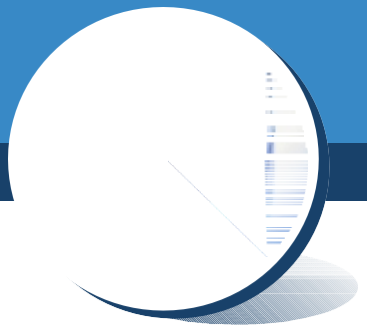
XML is used to Exchange Data

- Text format
- Software-independent, hardware-independent
- Exchange data between incompatible systems, given that they agree on the same tag definition.
- Can be read by many different types of applications

Benefits:

- Reduce the complexity of interpreting data
- Easier to expand and upgrade a system

XML Application3—Store Data



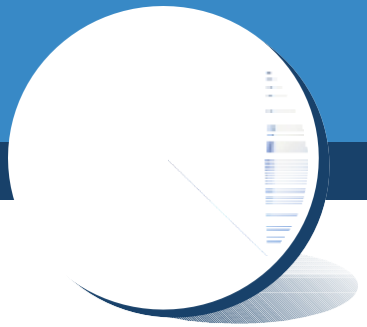
XML can be used to Store Data

- Plain text file
- Store data in files or databases
- Application can be written to store and retrieve information from the store
- Other clients and applications can access your XML files as data sources

Benefits:

Accessible to more applications

Conclusion



- XML is a self-descriptive language
- XML is a powerful language to describe structure data for web application
- XML is currently applied in many fields
- Many vendors already supports or will support XML

Thank You !