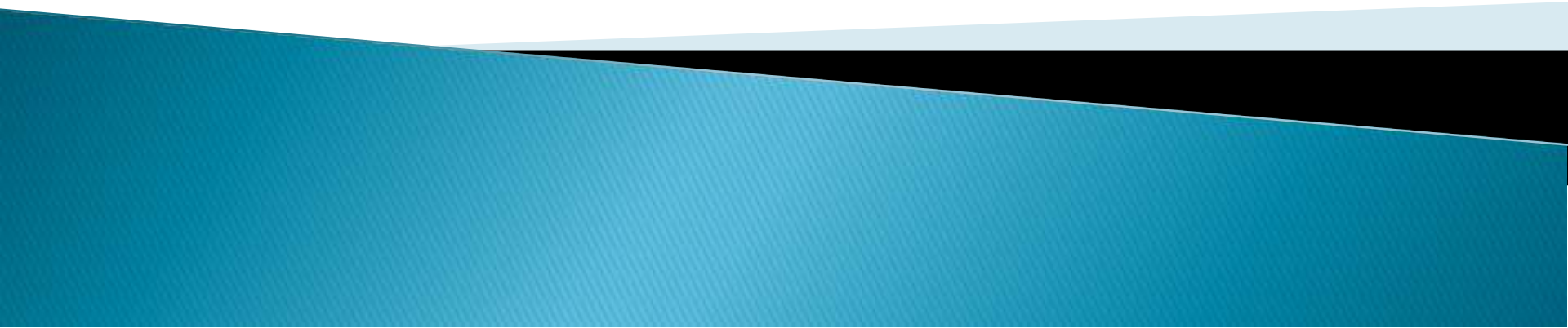
INTRODUCTION TO XML



- XML is a markup language that defines set of rules for encoding documents in a format that is both human-readable and machine-readable.

- XML stands for eXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to store and transport data
- XML was designed to be self-descriptive
- XML is often used to separate data from presentation.

Difference Between XML and HTML

- XML and HTML were designed with different goals:
- XML was designed to carry data – with focus on what data is
- HTML was designed to display data – with focus on how data looks
- XML tags are not predefined like HTML tags

What is Markup?

- Markup is information added to a document that enhances its meaning in certain ways, in that it identifies the parts and how they relate to each other.

```
<message>
<text>Hello, world!</text>
</message>
```

- This snippet includes the markup symbols, or the tags such as <message>...</message> and <text>... </text>. The tags <message> and </message> mark the start and the end of the XML code fragment. The tags <text> and </text> surround the text Hello, world!.

XML Syntax Rules

- XML documents must contain one **root** element that is the **parent** of all other elements

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

The XML Prolog

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

- The XML prolog is optional. If it exists, it must come first in the document.
- UTF-8 is the default character encoding for XML documents.
- UTF-8 is also the default encoding for HTML5, CSS, JavaScript, PHP, and SQL.

Rules

- All XML Elements Must Have a Closing Tag
- Opening and closing tags must be written with the same case
- XML tags are case sensitive. The tag <Letter> is different from the tag <letter>.
- XML Elements Must be Properly Nested
- XML Attribute Values Must Always be Quoted

Entity References

- If you place a character like "<" inside an XML element, it will generate an error because the parser interprets it as the start of a new element.

<message>salary < 1000</message>

<message>salary < 1000</message>

- There are 5 pre-defined entity references in XML

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

Comments in XML

```
<!-- This is a comment -->

<!-- This is an invalid -- comment -->
```

- Two dashes in the middle of a comment are not allowed
- XML does not truncate multiple white-spaces (HTML truncates multiple white-spaces to one single white-space)

XML Element

- An XML element is everything from (including) the element's start tag to (including) the element's end tag.
- An element can contain:
  - text
  - attributes
  - other elements
  - or a mix of the above

```
<element></element>
```

```
<element />
```

XML Naming Rules

- Element names are case-sensitive
- Element names must start with a letter or underscore
- Element names cannot start with the letters xml (or XML, or Xml, etc)
- Element names can contain letters, digits, hyphens, underscores, and periods
- Element names cannot contain spaces
- Any name can be used, no words are reserved (except xml).

Naming Conventions

Style	Example	Description
Lower case	<firstname>	All letters lower case
Upper case	<FIRSTNAME>	All letters upper case
Snake case	<first_name>	Underscore separates words (commonly used in SQL databases)
Pascal case	<FirstName>	Uppercase first letter in each word (commonly used by C programmers)
Camel case	<firstName>	Uppercase first letter in each word except the first (commonly used in JavaScript)

XML Attributes

- XML elements can have attributes, just like HTML.
- Attributes are designed to contain data related to a specific element.
- XML Attributes Must be Quoted

```
<person gender="female">
```

```
<person gender='female'>
```

Attribute Rules

- Some things to consider when using attributes are:
  - attributes cannot contain multiple values (elements can)
  - attributes cannot contain tree structures (elements can)
  - attributes are not easily expandable (for future changes)

```
<note day="10" month="01" year="2008"
to="Tove" from="Jani" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```

XML Namespaces

- In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.
- XML Namespaces provide a method to avoid element name conflicts.
- The namespace declaration has the following syntax.

xmlns:*prefix*="*URI*".

```
<h:table xmlns:h="http://www.google.com/">
<f:table xmlns:f="https://www.w3schools.com/furniture">
```

```
<root xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="https://www.w3schools.com/furniture">
```

- Default Namespaces

xmlns="*namespaceURI*"

```
<table xmlns="http://www.google.com/">
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```