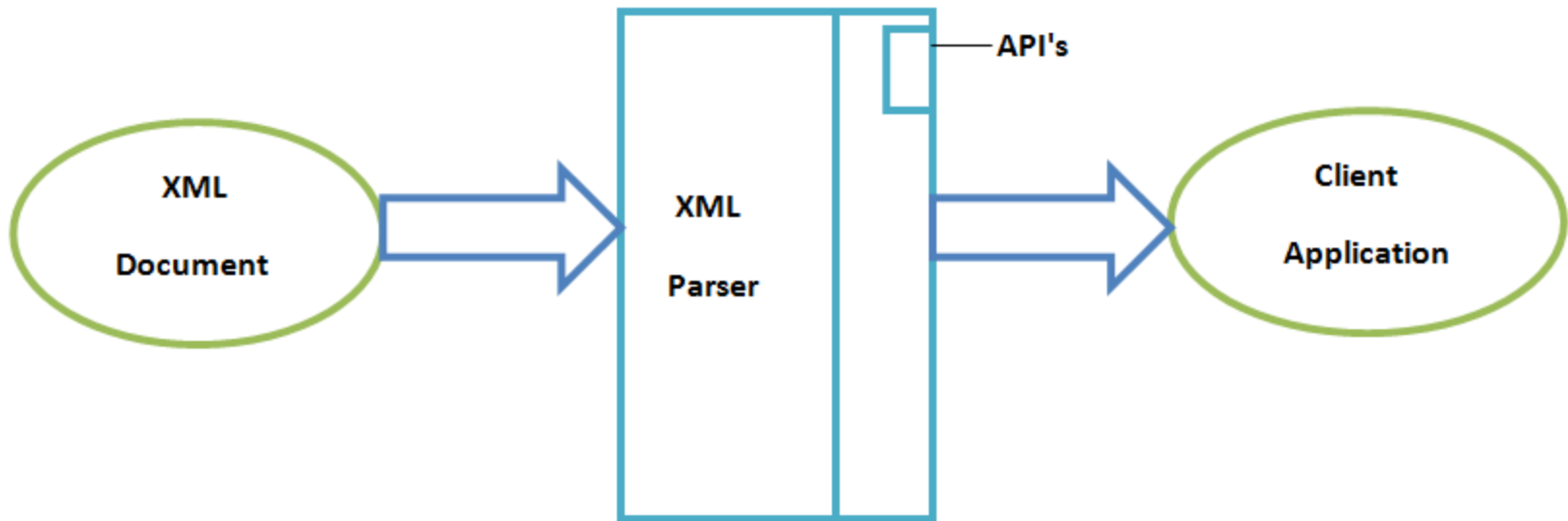


XML Parsers

(DOM & SAX)

XML Parsers

- An XML parser is a software library or package that provides interfaces for client applications to work with an XML document.
- The XML Parser is designed to read the XML document and create a way(interface or API) for programs to use XML.



XML Parsers

Two Types of parsers

➤ **DOM Parser**

➤ **SAX Parser**

DOM (Document Object Model)

- DOM is a platform that allows programs and scripts to dynamically access and update the content and structure of a XML documents.
- The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and provides interface(API) for access documents.
- The Document Object Model can be used with any programming language.
- DOM exposes the whole document to applications.

DOM (Document Object Model)

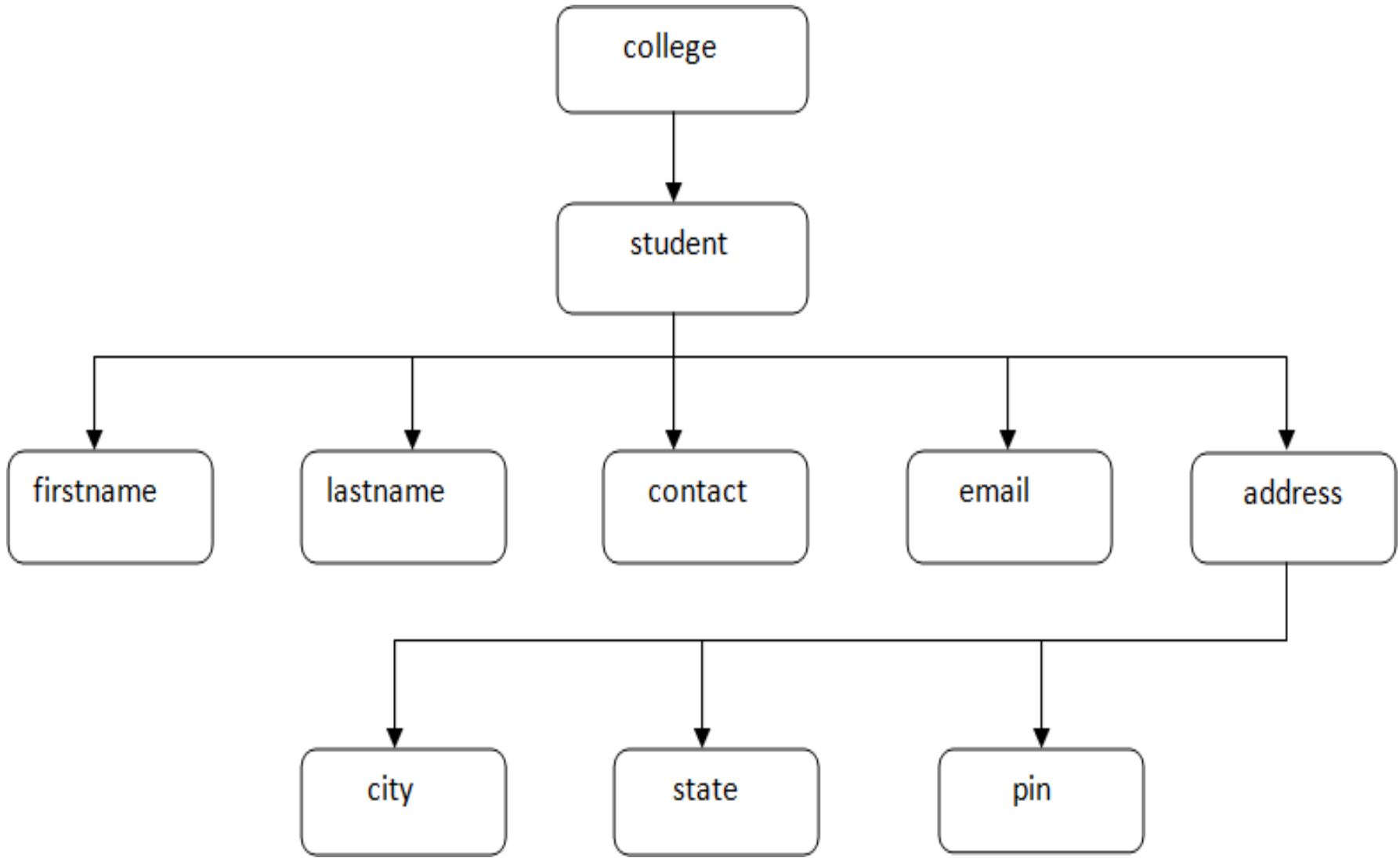
- The XML DOM defines a standard way for accessing and manipulating XML documents. It presents an XML document as a tree-structure.
- The tree structure makes easy to describe an XML document. A tree structure contains root element (as parent), child element and so on.
- The XML DOM makes a tree-structure view for an XML document.
- We can access all elements through the DOM tree. We can modify or delete their content and also create new elements.

DOM (Document Object Model)

```
<?xml version="1.0"?>  
<college>  
  <student>  
    <firstname>Durga</firstname>  
    <lastname>Madhu</lastname>  
    <contact>999123456</contact>  
    <email>dm@abc.com</email>  
    <address>  
      <city>Hyderabad</city>  
      <state>TS</state>  
      <pin>500088</pin>  
    </address>  
  </student>  
</college>
```

DOM (Document Object Model)

Let's see the tree-structure representation of the above example.



DOM (Document Object Model)

- We need a parser to read XML document into memory and converts into XML DOM Object that can be accesses with any programming language (here we can use PHP).
- The DOM parser functions are part of the PHP core. There is no installation needed to use these functions.
- To load XML document in PHP

```
$xmlDoc = new DOMDocument();
```

this statement creates an object.

```
$xmlDoc->load("note.xml");
```

this statement loads a xml file by using object.

DOM (Document Object Model)

These are some typical DOM properties in php:

- `X -> nodeName` - the name of X
- `X -> nodeValue` - the value of X
- `X->parentNode` - the parent node of X
- `X->childNodes` - the child nodes of X
- `X->attributes` - the attributes nodes of X

Where X is Node object.

“note.xml”

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<student>
```

```
<num>521</num>
```

```
<name>xyz</name>
```

```
<age>30</age>
```

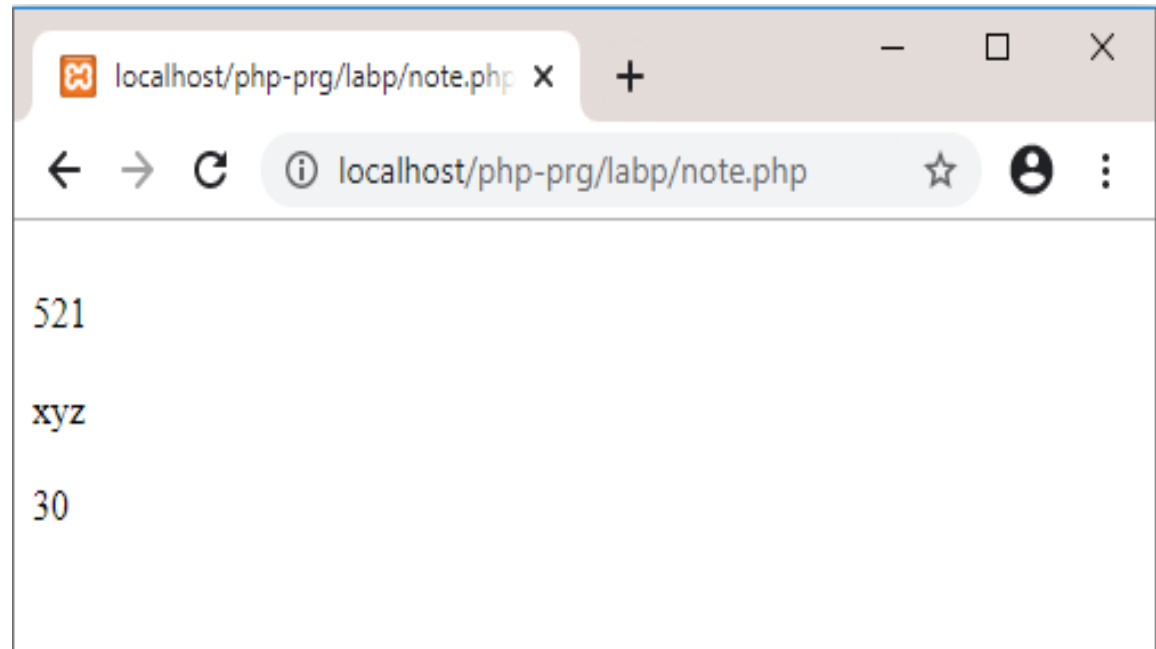
```
</student>
```

DOM (Document Object Model)

“Note.php”

```
<?php
$xmlDoc = new DOMDocument();
$xmlDoc->load("note.xml");
$xml = $xmlDoc->documentElement;
foreach ($xml->childNodes AS $item) {
    print $item->nodeValue . "<br>";
}
?>
```

Output:



SAX

Simple **A**PI for **X**ML

XML Parsers

What is an XML parser?

- An XML parser is a software library or package that provides interfaces for client applications to work with an XML document.
- The XML Parser is designed to read the XML and create a way for programs to use XML.

XML Parsers

Two types of parser

- SAX (Simple API for XML)
 - Event driven API
 - Sends events to the application as the document is read
- DOM (Document Object Model)
 - Reads the entire document into memory in a tree structure

Simple API for XML

SAX Parser

When should I use it?

- Large documents
- Memory constrained devices
- If you need not to modify the document

SAX Parser

Which languages are supported?

- Java
- Perl
- C++
- Python

SAX Implementation in Java

- Create a class which extends the SAX event handler

```
Import org.xml.sax.*;
import org.xml.sax.helpers.ParserFactory;
Public class SaxApplication extends HandlerBase {
    public static void main(String args[]) {
        }
}
```

SAX Implementation in Java

- Create a SAX Parser

```
public static void main(args[]) {
String parserName = "org.apache.xerces.parsers.SAXParser";
try {
    SaxApplication app = new SaxApplication();
    Parser parser = ParserFactory.makeParser(parserName);
    parser.setDocumentHandler(app);
    parser.setErrorHandler(app);
    parser.parse(new InputSource(args[0]));
} catch (Throwable t) {
    // Handle exceptions
}
}
```

SAX Implementation in Java

- Most important methods to parse
 - `void startDocument()`
 - Called once when document parsing begins
 - `void endDocument()`
 - Called once when parsing ends
 - `void startElement(...)`
 - Called each time an element begin tag is encountered
 - `void endElement(...)`
 - Called each time an element end tag is encountered
 - `void error(...)`
 - Called once when parsing error occurred.

DOM	SAX
✓ Tree model parser (Object based) (Tree of nodes).	✓ Event based parser (Sequence of events).
✓ DOM loads the file into the memory and then parse- the file.	✓ SAX parses the file as it reads it, i.e. parses node by node.
✓ Has memory constraints since it loads the whole XML file before parsing.	✓ No memory constraints as it does not store the XML content in the memory.
✓ DOM is read and write (can insert or delete nodes).	✓ SAX is read only i.e. can't insert or delete the node.
✓ If the XML content is small, then prefer DOM parser.	✓ Use SAX parser when XML content is large.
✓ Backward and forward search is possible for searching the tags and evaluation of the information inside the tags.	✓ SAX reads the XML file from top to bottom and backward navigation is not possible.
✓ Slower at run time.	✓ Faster at run time.