

T: Web service
Applications
RSS and Atom

encoding is a optional tag

* Rules for XML

- Only one parent element.
- Declared empty or non-empty
- Must have closing tags.
- Case-sensitive
- They are name-value pairs and values must be quoted.

12/2/22

* PHP & XML

XML Parser - 1. Tree based

IMTH 8 IMX 2. Event

Parser

↓
reads
and
converts
into
machine
readable
language.

DOM

↓
set of, instance tree object
object that
can read
XML data

XML There are 2 types

Simple XML and
DOM

The 1st we have to
load file then
2nd create object
and 3rd used to
traverse with
the help of
\$obj

Simple XML, DOM

\$obj = new DomDocument();

For event based XML you need to initialize all.
For creating event based

1. XML parser_create();

2. event declare or initialize
XML_set_element_handler();

3. XML_parse() XML_set_character_data_handler();

4. XML_parse() → to call function

5. XML_parser_free();

*1. PHP and XML

PHP has functions and classes to make it easier to work with XML document. PHP - you can read an XML document, you can also add more elements to the XML document, and also modify and remove element from the XML document.

eg.

```
<?php  
print "<?XML Version='10'>";  
?>
```

Answer - <?XML Version='10'>

*2. XML Parser

XML Parser is used to read XML document. XML Parsing is the process of reading an XML document and providing an interface to user application for accessing the document.

Two major types of XML parsers:-

I. Tree based parser

II. Event based parser

I Tree based parser

It transforms the XML document into tree. Then you can access tree element individually. eg of tree parser

i. Simple XML

ii. Dom

No use
foreach
loop.

II Event based parser

In it view an XML document as a series of event. When a specific event occurs it calls a function to handle it.

Attributes

- 4.) DOMAttr a() - represents an element attributes.
- 5.) DOMText a() - a plain text node.
- 6.) DOMCharacterData() - represents a character data node or c data CDATA node.

To start working with DOM you can create first dom object

```
$obj = new DomDocument();
```

We can use this object to read or write or write XML document. To create a node you call various methods of dom document class.

- 1.) createElement (Name, Value);
- 2.) createTextNode (Content); → ^{will create} text node. ↓ create an element
- 3.) CreateCDataSection (data); → ^{will create} character data node that contains character node
- 4.) SetAttribute(); → sets elements of attributes call name
- 5.) saveXML(); → it saves converts DomDocument into string
- 6.) createAttribute();

26/4/22

XML Continue

Adding elements to an existing doc.

```
<html>
```

```
<body>
```

```
<?php
```

```
$doc = new DomDocument();
```

```
$doc → load ("stock-list.xml");
```

```
$stocklistElements = $doc → getElementByTagName ("StockList");
```

```
$StockList = $StocklistElements → item (0);
```

```
$item = $doc → createElement ("Item");
```

```
$item → setAttribute ("type", "vegetable");
```

```
$StockList → appendChild ($item);
```

```
$name = $doc → createElement ("name", "carrot");
```

```
$item → appendChild ($name);
```

Continued in between Javascript chapter.

21/4/22

if statement
 if else
 if else if
 switch statement

Control Statement. JS Control Statement.

Loops → for
 → while
 → do while
 → for / m

<html>

<head>

<body>

<script type = "text/Javascript">

var x = 7;

var y = 7;

if (x == y)

{

}

document.write ("Both are equal");

</script>

</body>

</html>

if.....else

var x = 10;

var y = 2;

if (x % y == 0)

{

document.write ("X is divisible by y");

{

else

{
} document.write ("x is divisible by y");

if..... else if

<html>

<head>

<body>

<script = "text / Javascript" > new version of javascript
var your_age = 14;
var friend_age = 16;
if (your_age >= 18)

does not heed this
(if we write also A
to oh)

{ document.write ("Get a driver license");

}

else if (friend_age >= 18)

{

document.write ("Let our friend drive the car");

}

else

{

document.write ("Kids, strict to drive the car");

}

</script>

</body>

</head>

</html>

switch

<html>

<body>

<head>

<script>

var x;

for loop → initialization ; condition ; increment

| | |
|----------|--|
| Page No. | |
| Date | |

```
var d = new Date().getDay();  
switch(d){
```

case 0:

x = "Today is Sunday";
break;

}

case 1:

x = "Today is Monday";
break;

case 2:

x = "Today is Tuesday";
break;

}

upto

Saturday

default :
x = "Looking forward to weekend";

document.write(x)

```
</script>  
</body>  
</html>
```

for loop

<html>

<head>

<body>

<script>

```
for (var x = 6; x <= 10; x++)
```

{

document.write("Number is " + x);
-----||----- (" " +
");

3

```
</script>
</head>
</body>
</html>
```

for m loop.

```
<html>
```

```
<head>
```

```
<body>
```

```
<script>
```

```
var x; // welcome
```

```
var arr = [" "]; // array
```

```
var person = { name: "John", lname: "Doe", age: 25 };
```

```
for (x in person)
```

{

```
document.write(person[x] + " ");
```

when
array
not
(in)

```
</script>
```

```
</body>
```

```
</html>
```

John Doe 25 welcome

do while

while

```
<html>
```

loop:

```
<head>
```

```
<body>
```

```
var number = 3;
```

```
while (number <= 10)
```

{

```
document.write("number is : " + number);
```

```
number++;
```

}

8. Server load. → same as ①

* Disadvantages

1. Client side JS does not allow reading or writing of files.
2. JS cannot be used for networking applications because there is no support available.
3. JS does not have any multithreading or multiprocessing capability.
4. Security - No secured language.
∴ we use server validation.

* JS datatypes

One of the most fundamental characteristics of a programming language is the set of datatypes it supports. JS provides different datatypes to hold different types of values.

- | | |
|-------------------------|--------------------------------|
| 1. Primitive data types | 2. Non primitive datatypes |
| - String | - Object |
| - Number | - Array |
| - Boolean | - Regular expression (reg exp) |
| - undefined | |
| - null | |

* Variables

→ are containers for storing information. JS variables are used to hold values or expressions in them. Syntax is `var + variable name = value`

e.g. `var x = 10;`

22/4/22

* Javascript function

1. Built-in → confirm(), alert(), prompt()
2. User defined

Syntax

```
<script type = "text/javascript">
```

```
function addition(x, y)
```

```
{
```

```
    result = x + y;
```

```
    document.write("addition is : " + result)
```

```
}
```

```
add(10, 10)
```

```
add(20, 20)
```

```
</script>
```

* JS - HTML DOM Events

Window Object → total window info

Navigation →

Document →

Form →

History → → total history

Location → current URL info

* Events in Javascript

1. onclick. → if click on the left of the mouse this function will be generated.

```
<script type = "text/javascript">
```

```
function sayHello()
```

```
{
```

```
    alert("Hi");
```

```
}
```

```
</script>
```

```
<input type = "submit" onclick = "sayHello()" value = "ok">
```

```
</body>
```

```
</html>
```

```
<script type = "text/javascript">
```

```
function over()
```

```
{
```

```
    alert ("Mouse over");
```

```
}
```

```
function out()
```

```
{
```

```
    alert ("Mouse out");
```

```
}
```

<p> Bring our mouse inside the div to see result </p>

```
<div onmouseover = "over()" onmouseout = "out()>
```

```
<h2> This is inside div </h2>
```

```
</h2> </div>
```

```
</body>
```

```
</html>
```

* Variable scope:

- Local variable
- Global variable

The scope of variable is the region of our program in which it is defined

Global

local variable

A _____ has global scope which means it is defined everywhere in our javascript code

Local variable

A local variable will be visible only within a function where it is defined.

Function parameters are always located to that function.

Note: Global variables are accessible from anywhere in the program. On other hand local variable are accessible only in the function.

* JS control statements

JS supports conditional statements which are used to perform different actions based on different conditions. JS have foll. control statements

i. if statement

→ "if" is the fundamental statement that allows js to make decision and execute statements conditionally. We can use the if statement to execute some code if a specified condition is true.

ii. if else statement

→ "we can use" to execute some code if a condition is true and another condition is false. The keyword if execute a statement only if condition is true.

iii. if else if statement

→ "we can use" is the one level advance form of control statement that allows js to make correct decision out of several conditions.

iv. Switch Statement

"switch" is used to execute one of statement from many blocks of statement.

Switch Statement is like enhance if else statement. Only less confusing and more easy.

Switch (value / exprn)

{

case value 1:

// code;

break;

case value 2:

// code;

break;

default:

// statements;

}

- easier as only one class is there.
- 5. `children()` → returns an array of all elements children
- 6. `getName()` → returns the name of element as a string
- 7. `Xpath(xpath)` → find child element that match the given xpath, i.e. XML

e.g. `<html>`

`<?php>`

```
$stocklist = simpleXML_load_file ("StockListStockList.XML");
echo htmlspecialchars ($stocklist->asXML ());
?>
</html>
```

* Inbuilt functions of SimpleXML

1. `SimpleXMLElement_import dom(node);`
2. `SimpleXMLElement_load_file (filename);`
3. `SimpleXMLElement_string (String);`

27/4/22 Simple XML Writing

`<?php`

```
$simple_xml = new SimpleXMLElement ('<student></student>');
$simple_xml->addChild ('result');
```

```
$simple_xml->result[0]->addAttribute ('id', 1);
```

```
$simple_xml->result[0]->addChild ('name', 'Opal Kole');
```

```
($sgpa, '8.1');
```

```
($cgpa, '8.4');
```

```
$simple_xml->asXML ('simple_xml_create.xml');
```

?>

Output:

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

```
<student><result id="1">
```

```
<name> Opal Kole </name>
```

```
<sgpa> 8.1 </sgpa>
```

```

<cgpa>8.4</cgpa>
</result>
</student>

```

Simple XML reading

```

<?php
$xml = simplexml_load_file ('simple_xml_create
.xml');
foreach ($xml->result as $result)
{
    echo 'Result Id: ' . $result ['id'] .
        'Student'. $result->name .
        'SGPA' . $result->sgpa .
        'CGPA' . $result->cgpa."";
}

```

* PHP simplexml_import_dom() Function

It accepts an object of the class DOMNode as a parameter, converts it into an object of SimpleXMLElement class and returns it.

Syntax

```

Simplexml_import_dom ($filename, [$classname, $options,
$ns, $is_prefix]);

```

node(Mandatory) → This is an object of the class DOMNode
 class_name(Optional) → This is a string value to representing the name of class (sub class of the simpleXMLElement). If you pass this value, the given XML string is returned as the object of the specified class