

PHP – HÜ 6

Index.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Hausübung 6 - DOM/SAX Parser</title>
</head>
<body>
<?php
spl_autoload_register('autoload');
function autoload($sClassName){
    require_once("$sClassName.php");
}
//DOMPARSER
$oDomParser = new DomParser("quiz.xml");
if($oDomParser->load()){
    $oDomParser->output();
}
else echo("Document could not be loaded!");

//SAXPARSER
$saxParser = new SaxParser("quiz.xml");
$saxParser->parse();
?>
</body>
</html>
```

Quiz.xml

```
<?xml version="1.0" encoding="utf-8"?>
<quiz>
  <meta>
    <title>WEB KWM Quiz</title>
    <difficulty>Mittel</difficulty>
    <author>
      <firstname>Max</firstname>
      <lastname>Masterstudent</lastname>
    </author>
    <start>
      <date>
        <day>1</day>
        <month>4</month>
        <year>2021</year>
      </date>
      <time>
        <hours>7</hours>
        <minutes>15</minutes>
      </time>
    </start>
    <end>
      <date>
        <day>30</day>
        <month>6</month>
        <year>2021</year>
      </date>
      <time>
        <hours>23</hours>
        <minutes>59</minutes>
      </time>
    </end>
  </meta>
  <questions>
    <question type="open" datatype="integer" correctanswer="2006">
      <text>In welchem Jahr wurden die ersten KWM Studierenden aufgenommen?</text>
    </question>
    <question type="single">
      <text>Wer war der/die erste Studiengangsleiter/in des KWM-Vorgängers CBL?
    </text>
    <answer correct="false">Josef Altmann</answer>
    <answer correct="false">Heidrun Allert</answer>
    <answer correct="true">Wolfgang Schreiner</answer>
  </question>
    <question type="open" datatype="string" correctanswer="Engineering für Computer-
basiertes Lernen">
      <text>Wofür stand der Name CBL?</text>
    </question>
    <question type="multiple">
      <text>Wer lehrt nebenberuflich bei KWM?</text>
      <answer correct="false">Josef Altmann</answer>
      <answer correct="false">Mirjam Augstein</answer>
      <answer correct="true">Wolfgang Hoffelner</answer>
      <answer correct="false">Tanja Jadin</answer>
      <answer correct="false">Carrie Kovacs</answer>
      <answer correct="false">Christina Ortner</answer>
      <answer correct="false">Johannes Schönböck</answer>
      <answer correct="true">Wolfgang Schreiner</answer>
    </question>
    <question type="multiple">
      <text>Welche Kurse werden bei KWM gelehrt?</text>
      <answer correct="true">Serverseitige Web-Programmierung</answer>
      <answer correct="true">Web und Hypermedia Grundlagen</answer>
      <answer correct="false">Deep Learning</answer>
    </question>
  </questions>
</quiz>
```

```
<answer correct="true">Lernpsychologie</answer>
<answer correct="true">Datenbanken</answer>
<answer correct="false">Compilerbau</answer>
</question>
</questions>
</quiz>
```

DomParser.php

```
<?php
```

```
class DomParser{
    private string $sFile;
    private DomDocument $oDoc;

    public function __construct(string $sFile){
        $this->sFile = $sFile;
        $this->oDoc = new DOMDocument();
    }

    public function load():bool{
        return ($this->oDoc->load($this->sFile));
    }

    public function output():void
    {
        $sOutput = "";
        $oElementsTitle = $this->oDoc->getElementsByTagName("title");

        //TITLE
        foreach ($oElementsTitle as $oTitle) {
            $sOutput .= "<h1>" . $oTitle->nodeValue . "</h1>";
        }

        //AUTHOR
        $oElementsAuthor = $this->oDoc->getElementsByTagName("author");
        foreach ($oElementsAuthor as $oAuthor) {
            $oInfoAuthor = $oAuthor->childNodes;
            $sTempInfo = "";
            foreach ($oInfoAuthor as $oInfoItem) {
                $sCurrentNodeName = $oInfoItem->nodeName;
                if ($sCurrentNodeName == "firstname") {
                    $sTempInfo .= $oInfoItem->nodeValue . "<br>";
                } else if ($sCurrentNodeName == "lastname") {
                    $sOutput .= "<b>Ersteller*in: </b>" . $oInfoItem->nodeValue . ", ";
                }
            }
            $sOutput .= $sTempInfo;
        }

        //DIFFICULTY
        $oElementsDiff = $this->oDoc->getElementsByTagName("difficulty");
        foreach ($oElementsDiff as $oDiff) {
            $sOutput .= "<b>Schwierigkeitsgrad: </b>" . $oDiff->nodeValue . "<br>";
        }
    }
}
```

```

//ONLINE
//START
$oElementsStart = $this->oDoc->getElementsByTagName("start");
foreach ($oElementsStart as $oStart) {
    $sOutput .= "<b>Online:</b> von ";
    $oInfoStart = $oStart->childNodes;
    foreach ($oInfoStart as $oInfoItem) {
        $sCurrentNodeName = $oInfoItem->nodeName;
        if ($sCurrentNodeName == "date") {
            $oDate = $oInfoItem->childNodes;
            foreach ($oDate as $oDateItem) {
                $sCurrentNodeNameDate = $oDateItem->nodeName;
                if ($sCurrentNodeNameDate == "day") {
                    $sOutput .= $oDateItem->nodeValue . ".";
                } else if ($sCurrentNodeNameDate == "month") {
                    $sOutput .= $oDateItem->nodeValue . ".";
                } else if ($sCurrentNodeNameDate == "year") {
                    $sOutput .= $oDateItem->nodeValue . " ";
                }
            }
        } else if ($sCurrentNodeName == "time") {
            $oTime = $oInfoItem->childNodes;
            foreach ($oTime as $oTimeItem) {
                $sCurrentNodeNameTime = $oTimeItem->nodeName;
                if ($sCurrentNodeNameTime == "hours") {
                    $sOutput .= "(" . $oTimeItem->nodeValue . ":";
                } else if ($sCurrentNodeNameTime == "minutes") {
                    $sOutput .= $oTimeItem->nodeValue . ") ";
                }
            }
        }
    }
}

//END
$oElementsEnd = $this->oDoc->getElementsByTagName("end");
foreach ($oElementsEnd as $oEnd) {
    $oInfoEnd = $oEnd->childNodes;
    foreach ($oInfoEnd as $oInfoItem) {
        $sCurrentNodeName = $oInfoItem->nodeName;
        if ($sCurrentNodeName == "date") {
            $oDate = $oInfoItem->childNodes;
            foreach ($oDate as $oDateItem) {
                $sCurrentNodeNameDate = $oDateItem->nodeName;
                if ($sCurrentNodeNameDate == "day") {
                    $sOutput .= "bis " . $oDateItem->nodeValue . ".";
                } else if ($sCurrentNodeNameDate == "month") {
                    $sOutput .= $oDateItem->nodeValue . ".";
                } else if ($sCurrentNodeNameDate == "year") {
                    $sOutput .= $oDateItem->nodeValue . " ";
                }
            }
        } else if ($sCurrentNodeName == "time") {
            $oTime = $oInfoItem->childNodes;
            foreach ($oTime as $oTimeItem) {
                $sCurrentNodeNameTime = $oTimeItem->nodeName;
                if ($sCurrentNodeNameTime == "hours") {
                    $sOutput .= "(" . $oTimeItem->nodeValue . ":";
                } else if ($sCurrentNodeNameTime == "minutes") {
                    $sOutput .= $oTimeItem->nodeValue . "<br><br>";
                }
            }
        }
    }
}

```

```

//QUESTIONS ELEMENT
$oElementsQuestions = $this->oDoc->getElementsByTagName("questions");
foreach ($oElementsQuestions as $oQuestions) {

    //QUESTION ELEMENT + COUNTER FOR QUESTIONS
    $oElementsQuestion = $this->oDoc->getElementsByTagName("question");
    $iCounter = 0;

    foreach ($oElementsQuestion as $oQuestion) {
        $iCounter++;
        if ($oQuestion->hasAttribute("type")) {
            $sInputType = $oQuestion->getAttribute("type");
        }
        $oQuestionInfos = $oQuestion->childNodes;

        //QUESTION-TEXT AND ANSWERS
        foreach ($oQuestionInfos as $oQuestionInfo) {
            $sCurrentNodeName = $oQuestionInfo->nodeName;

            //QUESTION AND INPUT FIELD FOR OPEN INPUT
            if ($sCurrentNodeName == "text" && $oQuestion->
>hasAttribute("correctanswer")) {
                $sOutput .= "<b>" . $iCounter . ". <label for='" . $oQuestion->
>getAttribute("correctanswer") . "'>" . $oQuestionInfo->nodeValue . "</label></b>";
                if ($oQuestion->hasAttribute("datatype"))
                $sOutput .= " (bitte <b>" . $oQuestion->
>getAttribute("datatype") . "</b> eingeben) ";
                $sOutput .= "<input type='text' name='" . $oQuestion->
>getAttribute("correctanswer") . "'><br>";

                } //QUESTION FOR RADIO/CHECKBOXES
                else if ($sCurrentNodeName == "text") {
                    $sOutput .= "<b>" . $iCounter . ". " . $oQuestionInfo->
>nodeValue . "</b>";
                    if ($oQuestion->hasAttribute("datatype"))
                    $sOutput .= " (bitte <b>" . $oQuestion->
>getAttribute("datatype") . "</b> eingeben) ";
                    $sOutput .= "<br>";
                }
                //RADIO/CHECKBOXES
                if ($sCurrentNodeName == "answer") {
                    if ($sInputType == "single") {
                        $sOutput .= "<input type='radio' value='" . $oQuestionInfo->
>nodeValue . "' name='answer" . $iCounter . "'>";
                        $sOutput .= "<label for='answer" . $iCounter . "'>" .
$oQuestionInfo->nodeValue . "</label><br>";
                    } else if ($sInputType == "multiple") {
                        $sOutput .= "<input type='checkbox' value='" .
$oQuestionInfo->nodeValue . "' name='" . $oQuestionInfo->nodeValue . "'>";
                        $sOutput .= "<label for='" . $oQuestionInfo->nodeValue .
"'">" . $oQuestionInfo->nodeValue . "</label><br>";
                    }
                }
            }
            $sOutput .= "<br>";
        }
    }
    $sOutput .= "<br><input type='submit' value='Abschicken'>";
    echo($sOutput);
}

?>

```

SaxParser.php

<?php

```
class SaxParser
{
    private string $file;
    private $parser;
    private string $sCurrentElement;
    private int $iCounter = 0;
    private string $sFirstName = ""; //Saves String
    private string $sDifficulty = ""; //Saves String
    private string $sType = ""; //Radio, Checkbox, Textbox
    private string $sDataType = ""; //String, Integer, ... for Textbox

    public function __construct(string $file){
        $this->file = $file;
        $this->parser = xml_parser_create();
        xml_set_object($this->parser, $this);
        xml_parser_set_option($this->parser, XML_OPTION_CASE_FOLDING, false); //dann
nicht case-sensitive
        xml_parser_set_option($this->parser, XML_OPTION_SKIP_WHITE, true); //großer
whitespace nicht berücksichtigen

        //Arbeitet Eventbasiert
        //Eventhandler mitteilen (bei Start- und Endevents):
        xml_set_element_handler($this->parser, "startElementHandler",
"endElementHandler");

        //Bei unstrukturierten Inhalt:
        xml_set_character_data_handler($this->parser, "characterDataHandler");
    }

    //Dokument wird nicht nur geladen wie bei DOMParser,
    //sondern der Parser führt selbstständig Dokument ganz durch
    //(in Reihenfolge von XML Dok), darum Funktion parse() statt load()
    public function parse(){
        if(!$handle = fopen($this->file, "r")){ //Modus r -> Öffnen zum Lesen
            return;
        }

        while($data = fread($handle, filesize($this->file))){
            if(!xml_parse($this->parser, $data, feof($handle))){
                $line = xml_get_current_line_number($this->parser);
                $errorCode = xml_get_error_code($this->parser);
                $errorText = xml_error_string($errorCode);
                echo("<b>Error at line ".$line.", code: ".$errorCode.
                    ", message: ".$errorText."</b><br>");
            }
        }

        fclose($handle);
        xml_parser_free($this->parser); //Speicherplatz freigeben
    }

    //Starttags
    private function startElementHandler($XMLParser, string $sElement, array
$aAttributes){
        $this->sCurrentElement = $sElement;
```

```

switch ($sElement){
    case "author":
        echo ("<b>Ersteller*in: </b>");
        break;
    case "start":
        echo("<b>Online: </b> von ");
        break;
    case "end":
        echo("bis ");
        break;
    case "question":
        $this->iCounter++; //Counter for Question-Numbers
        echo("<br><b>".$this->iCounter."</b>");
        //Save question type
        if(isset($aAttributes["type"])) {
            //Textbox, Radiobuttons or Checkboxes
            if ($aAttributes["type"] == "single")
                $this->sType = "single";
            else if ($aAttributes["type"] == "open")
                $this->sType = "open";
            else if ($aAttributes["type"] == "multiple")
                $this->sType = "multiple";
        }
        //Sava data type of question
        if(isset($aAttributes["datatype"])){
            $this->sDataType = $aAttributes["datatype"];
        }
        break;
    case "answer":
        echo("<br>");
        if($this->sType == "single")
            echo("<input type='radio' name='answer' . $this->iCounter . '>");
        else if($this->sType == "multiple")
            echo("<input type='checkbox' name='answer' . $this->iCounter.'>");
        break;
    default: break;
}
}

```

```

//Endtags
private function endElementHandler($XMLParser, string $element){
    switch($element){
        case "lastname":
            echo(", ".$this->sFirstName);
            break;
        case "author":
            echo("<br>");
            echo("<b>Schwierigkeitsgrad: </b>".$this->sDifficulty."<br>");
            break;
        case "day":
            echo(".");
            break;
        case "month":
            echo(".");
            break;
        case "year":
            echo(" (");
            break;
        case "hours":
            echo(":");
            break;
        case "minutes":
            echo(")");
            break;
        case "end":
            echo("<br>");
    }
}

```



```

        break;
    case "question":
        //If datatype available -> must stand next to the question
        if($this->sDataType != "")
            echo("(bitte <b>".$this->sDataType."</b> eingeben)");
        //If textbox -> must stand next to the question
        if($this->sType == "open")
            echo("<input type='text' name='question'.".$this->iCounter.">");
        echo("<br>");
        //Empty data for following question
        $this->sType = "";
        $this->sDataType = "";
        break;
    case "questions":
        echo("<br><br><input type='submit' value='Abschicken'>");
        break;
    default: break;
}
}

```

```

//Textdata
private function characterDataHandler($XMLParser, string $data){
    if($this->sCurrentElement == "title")
        echo("<h1>".$data."</h1>");
    else if($this->sCurrentElement == "firstname")
        $this->sFirstName .= $data;
    else if($this->sCurrentElement == "difficulty")
        $this->sDifficulty .= $data;
    else if($this->sCurrentElement == "text")
        echo("<b>".$data."</b>");
    else if($this->sCurrentElement == "answer"){
        if($this->sType == "single")
            echo("<label for='answer' . $this->iCounter . '>' . $data .
"</label>");
        else if($this->sType == "multiple")
            echo("<label for=' . $data . '>' . $data . "</label>");
        }
    else echo($data);
}
}

```

Testfälle

DomParser:

WEB KWM Quiz

Ersteller*in: Masterstudent, Max

Schwierigkeitsgrad: Mittel

Online: von 1.4.2021 (7:15) bis 30.6.2021 (23:59)

1. In welchem Jahr wurden die ersten KWM Studierenden aufgenommen? (bitte **integer** eingeben)

2. Wer war der/die erste Studiengangsleiter/in des KWM-Vorgängers CBL?

- ☐ Josef Altmann
- ☐ Heidrun Allert
- ☐ Wolfgang Schreiner

3. Wofür stand der Name CBL? (bitte **string** eingeben)

4. Wer lehrt nebenberuflich bei KWM?

- ☐ Josef Altmann
- ☐ Mirjam Augstein
- ☐ Wolfgang Hoffelner
- ☐ Tanja Jadin
- ☐ Carrie Kovacs
- ☐ Christina Ortner
- ☐ Johannes Schönböck
- ☐ Wolfgang Schreiner

5. Welche Kurse werden bei KWM gelehrt?

- ☐ Serverseitige Web-Programmierung
- ☐ Web und Hypermedia Grundlagen
- ☐ Deep Learning
- ☐ Lernpsychologie
- ☐ Datenbanken
- ☐ Compilerbau

Abschicken

SaxParser:

WEB KWM Quiz

Ersteller*in: Masterstudent, Max



Schwierigkeitsgrad: Mittel

Online: von 1. 4. 2021 (7: 15) bis 30. 6. 2021 (23: 59)

1. In welchem Jahr wurden die ersten KWM Studierenden aufgenommen? (bitte integer eingeben)

2. Wer war der/die erste Studiengangsleiter/in des KWM-Vorgängers CBL?

- ☐ Josef Altmann
- ☐ Heidrun Allert
- ☐ Wolfgang Schreiner

3. Wofür stand der Name CBL? (bitte string eingeben)

4. Wer lehrt nebenberuflich bei KWM?

- ☐ Josef Altmann
- ☐ Mirjam Augstein
- ☐ Wolfgang Hoffelner
- ☐ Tanja Jadin
- ☐ Carrie Kovacs
- ☐ Christina Ortner
- ☐ Johannes Schönböck
- ☐ Wolfgang Schreiner

5. Welche Kurse werden bei KWM gelehrt?

- ☐ Serverseitige Web-Programmierung
- ☐ Web und Hypermedia Grundlagen
- ☐ Deep Learning
- ☐ Lernpsychologie
- ☐ Datenbanken
- ☐ Compilerbau

Abschicken