

# Tutorial 4

Web Development Technologies

# Reminders

- Everyone should have XAMPP installed for this tutorial and for part 3 of the first Assignment.
- Install the plugins for PHP given in the first tutorial.
- **Assignment 1 due date is the 29<sup>th</sup> of January for the three parts.**

# Agenda

- HTTP
- HTML
- PHP
- JSON
- XML
- Class Exercise (registration/login page)

# HTTP



- GET request vs POST request

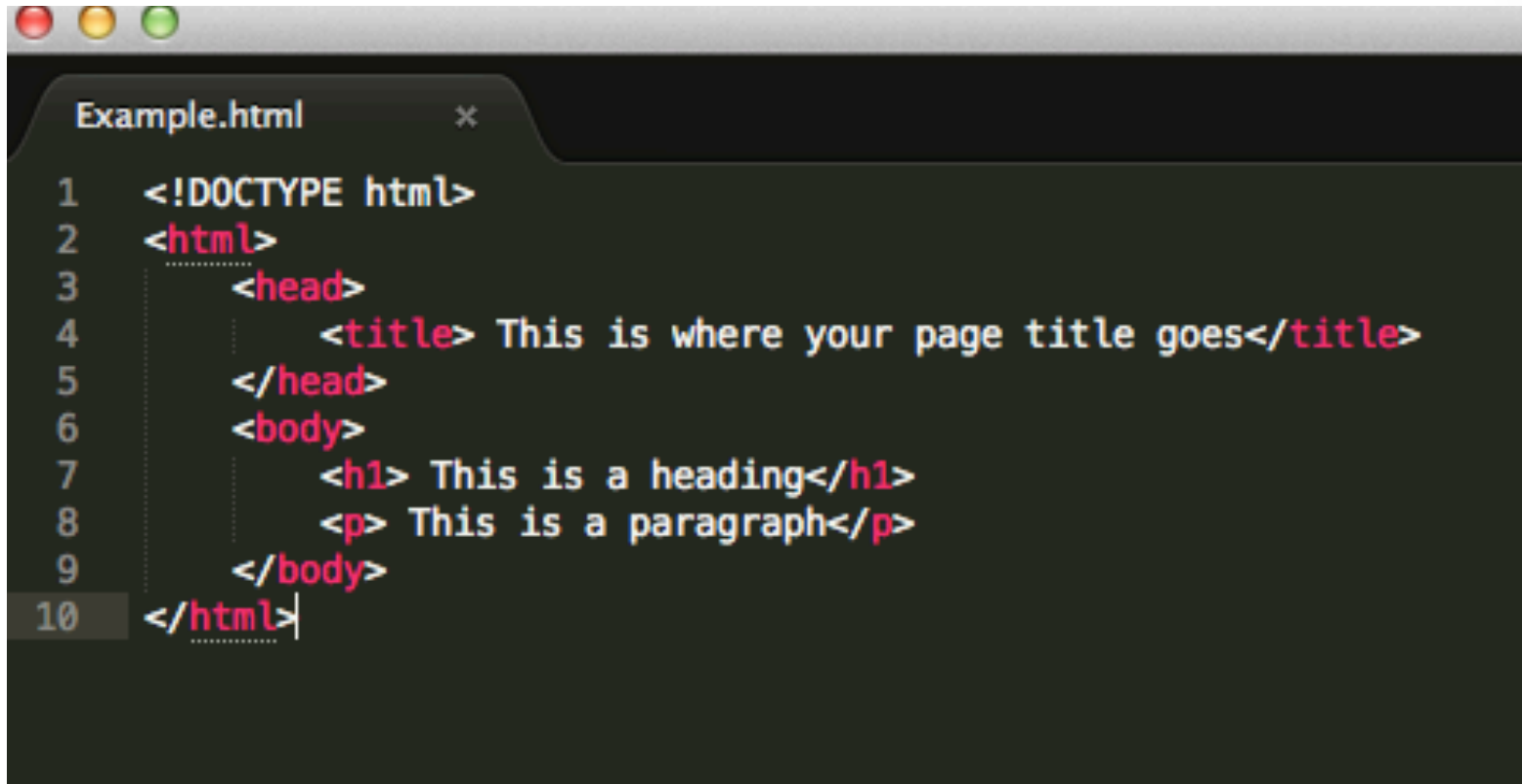
# HTTP

- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.
- HTTP works as a request-response protocol between a client and server.
- GET request appears in the query string (URL).
- POST request appears in the http message body.
- POST requests are never cached.
- POST requests do not remain in browser history.

# HTML

- HTML stands for Hyper Text Markup Languages, which is the most widely used language on the Web to develop web pages.
- HTML is a Markup Language which means you use it to simply “markup” a text document with tags that tells a Web browser how to structure it to display.
- HTML extension is .html

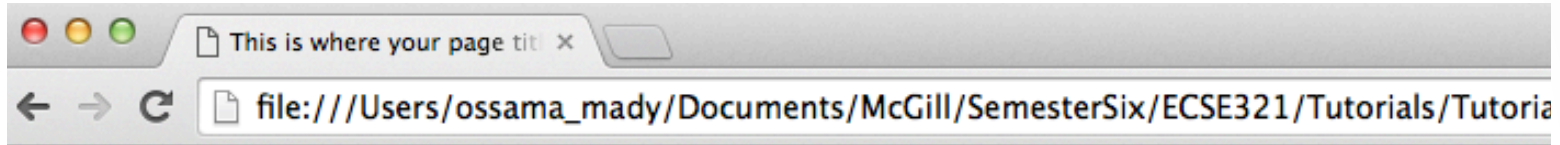
# HTML

A screenshot of a code editor window titled "Example.html". The editor has a dark background and shows a valid HTML document structure. The code is as follows:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title> This is where your page title goes</title>
5      </head>
6      <body>
7          <h1> This is a heading</h1>
8          <p> This is a paragraph</p>
9      </body>
10 </html>
```

The code is color-coded: opening and closing tags are in red, and content text is in white. Line numbers 1 through 10 are on the left. The window has standard macOS window controls (red, yellow, green buttons) at the top left.

# HTML

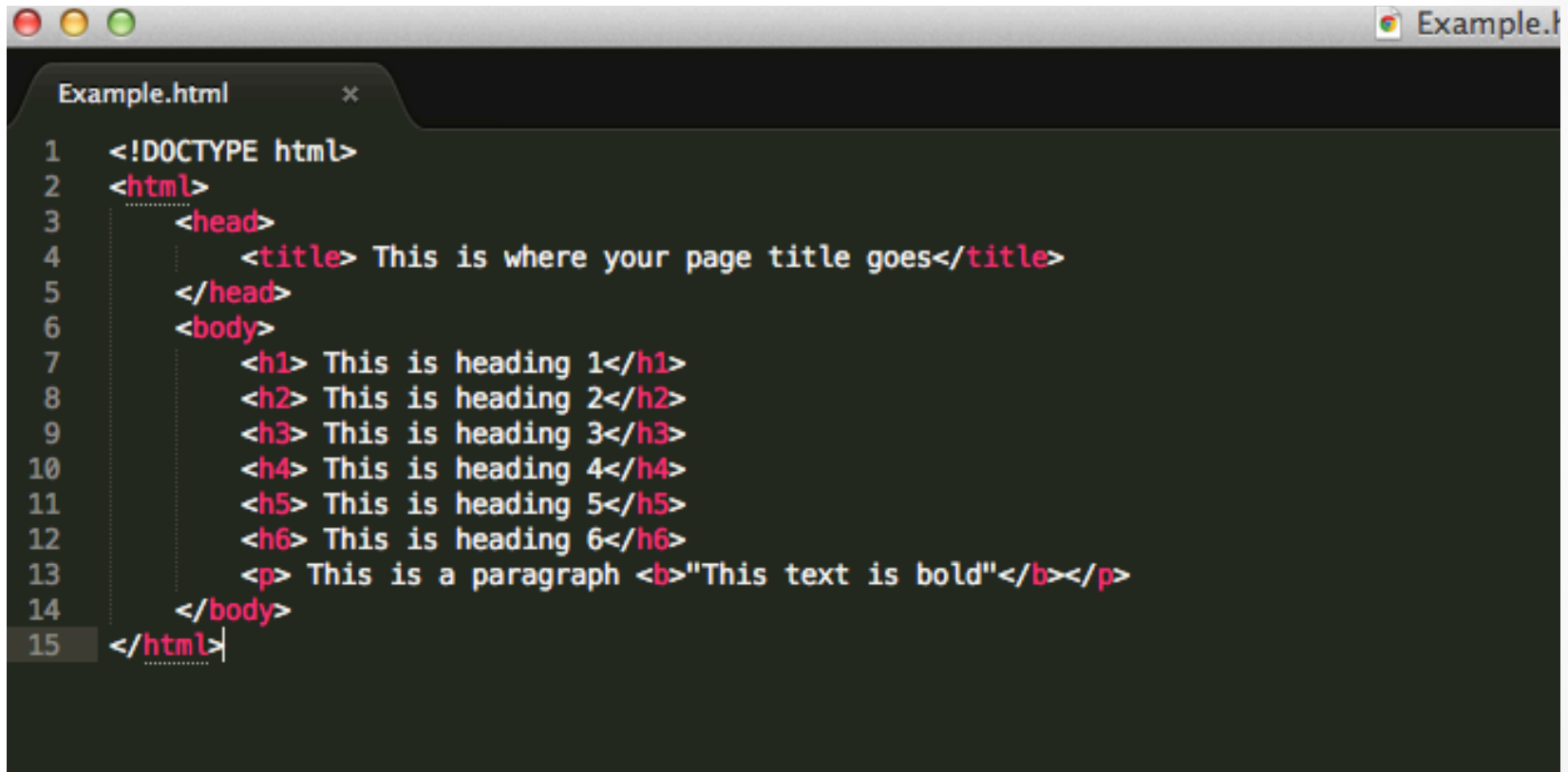


## **This is a heading**

This is a paragraph

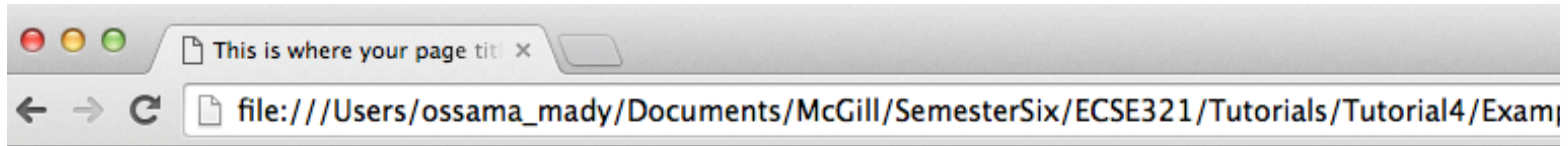


# HTML

A screenshot of a code editor window titled 'Example.html'. The editor has a dark background with light-colored text. The code is an HTML document structure with line numbers 1 through 15 on the left. The code includes a DOCTYPE declaration, an opening <html> tag, a <head> section with a <title> tag, and a <body> section containing six heading tags (<h1> through <h6>) and a paragraph tag (<p>) with bold text. The code is color-coded: tags are in red, text content is in white, and the closing tag for the paragraph is in blue. The window has standard macOS window controls (red, yellow, green buttons) in the top left corner and a title bar with the text 'Example.html' and a close button (X) in the top right corner.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title> This is where your page title goes</title>
5      </head>
6      <body>
7          <h1> This is heading 1</h1>
8          <h2> This is heading 2</h2>
9          <h3> This is heading 3</h3>
10         <h4> This is heading 4</h4>
11         <h5> This is heading 5</h5>
12         <h6> This is heading 6</h6>
13         <p> This is a paragraph <b>"This text is bold"</b></p>
14     </body>
15 </html>
```

# HTML



**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

**This is heading 5**

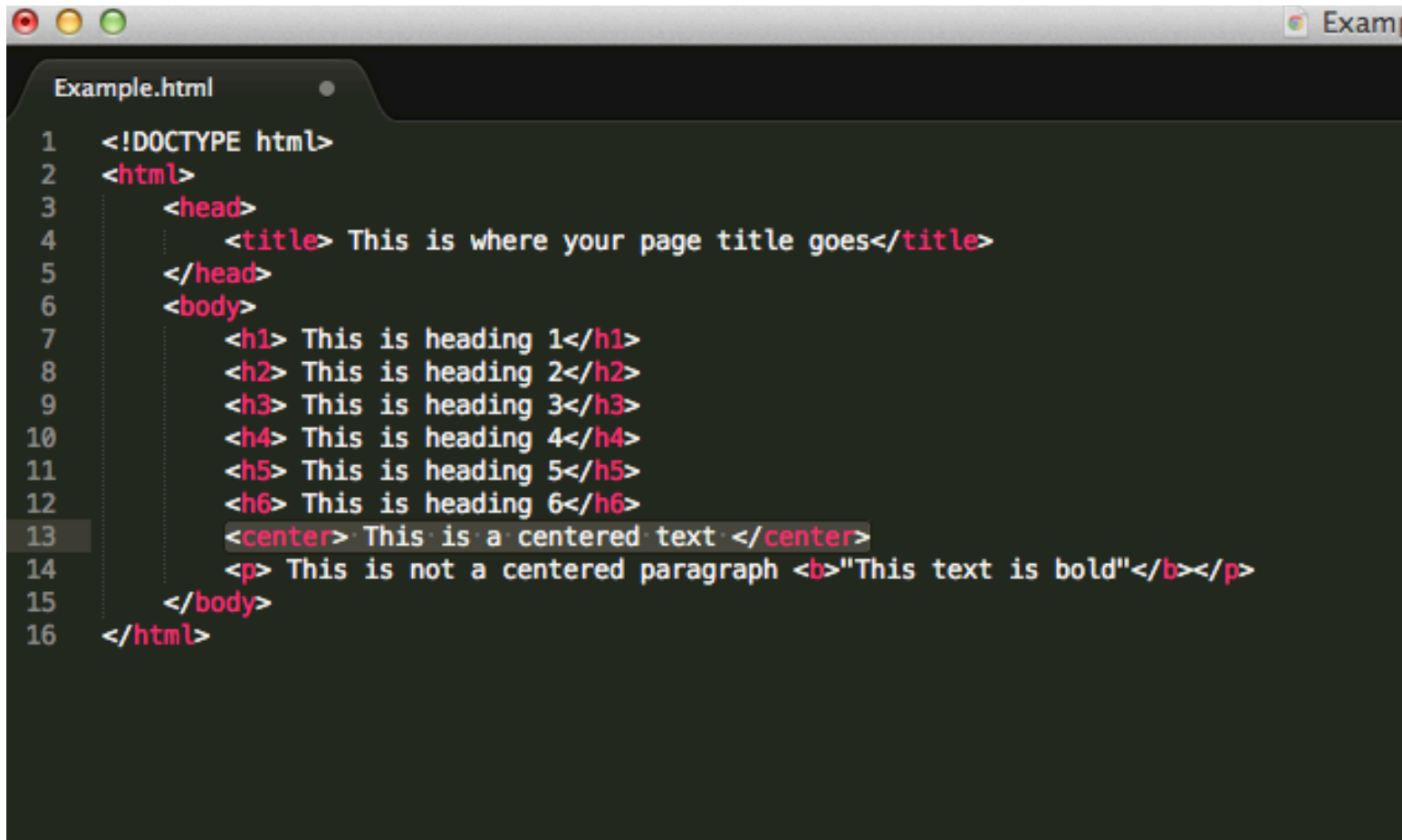
**This is heading 6**

This is a paragraph "**This text is bold**"

# HTML

- HTML is a collection of tags
- Tags can be categorized into two types of tags
  1. Tags with attributes and text in between  
`<TagName attr1="Value" attr2="Value"> Text </TagName>`
  2. Tags with attributes only  
`<TagName attr1="Value"/>`
- The text has to be contained within the opening and the closing tags of the corresponding element.

# HTML

A screenshot of a code editor window titled 'Example.html'. The editor has a dark background with light-colored text. The code is an HTML document structure with line numbers 1 through 16 on the left. The code includes a DOCTYPE declaration, an opening <html> tag, a <head> section with a <title> tag, and a <body> section containing six <h1> through <h6> heading tags, a <center> tag with text, and a <p> tag with bold text. The code is as follows:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title> This is where your page title goes</title>
5      </head>
6      <body>
7          <h1> This is heading 1</h1>
8          <h2> This is heading 2</h2>
9          <h3> This is heading 3</h3>
10         <h4> This is heading 4</h4>
11         <h5> This is heading 5</h5>
12         <h6> This is heading 6</h6>
13         <center> This is a centered text </center>
14         <p> This is not a centered paragraph <b>"This text is bold"</b></p>
15     </body>
16 </html>
```

# HTML

---

**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

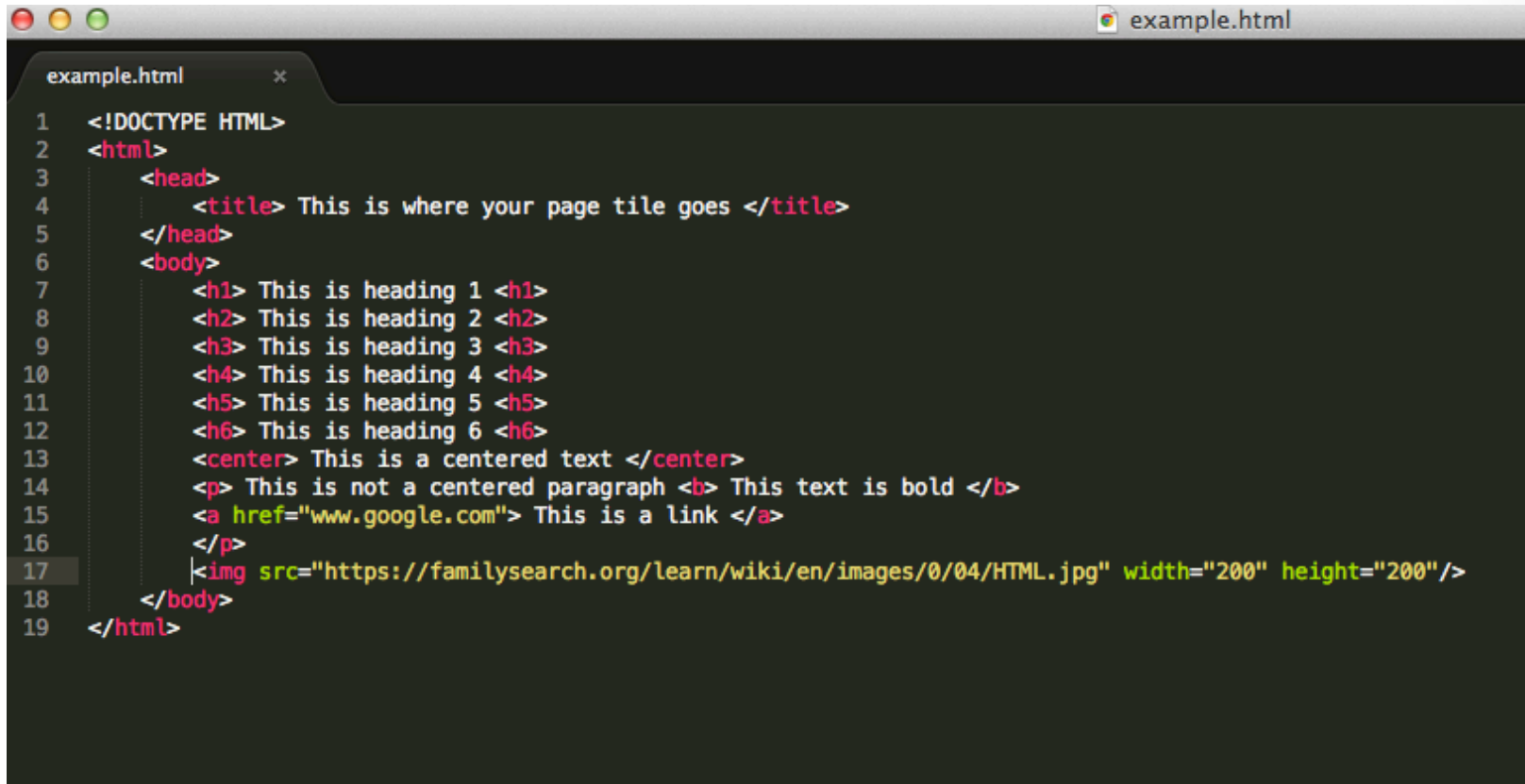
**This is heading 5**

**This is heading 6**

This is a centered text

This is not a centered paragraph "**This text is bold**"

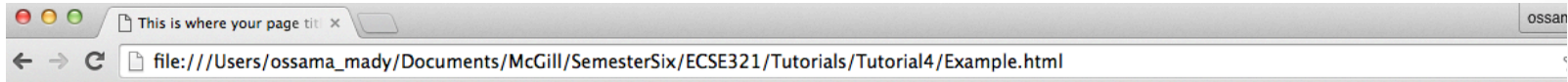
# HTML

A screenshot of a code editor window titled 'example.html'. The editor shows a complete HTML document structure. The code is as follows:

```
1  <!DOCTYPE HTML>
2  <html>
3      <head>
4          <title> This is where your page tile goes </title>
5      </head>
6      <body>
7          <h1> This is heading 1 </h1>
8          <h2> This is heading 2 </h2>
9          <h3> This is heading 3 </h3>
10         <h4> This is heading 4 </h4>
11         <h5> This is heading 5 </h5>
12         <h6> This is heading 6 </h6>
13         <center> This is a centered text </center>
14         <p> This is not a centered paragraph <b> This text is bold </b>
15         <a href="www.google.com"> This is a link </a>
16         </p>
17         
18     </body>
19 </html>
```

The code is color-coded: DOCTYPE and tags are in grey, elements in blue, text in white, and attributes in yellow. Line 17 is highlighted with a dark background.

# HTML



**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

**This is heading 5**

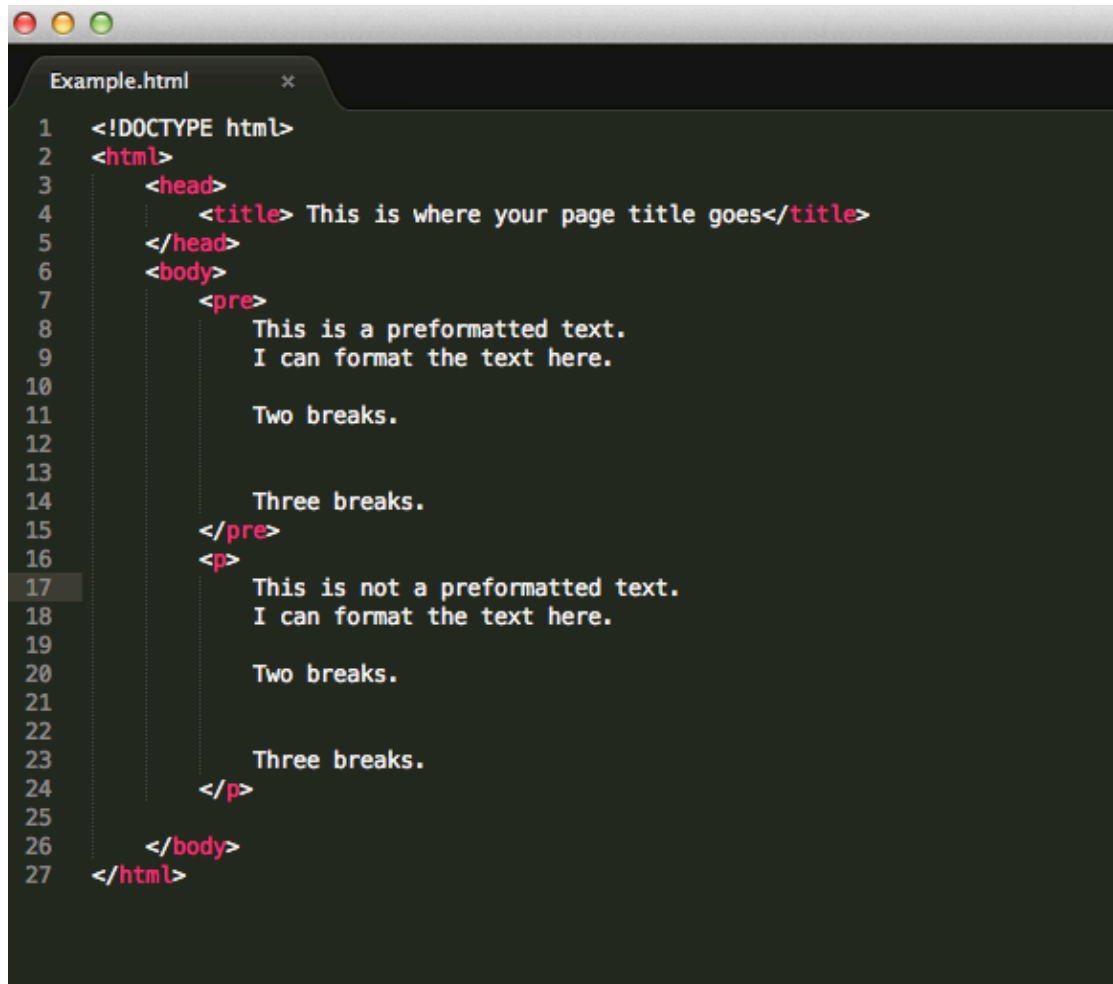
**This is heading 6**

This is a centered text

This is not a centered paragraph "**This text is bold**" [This is a link](#)



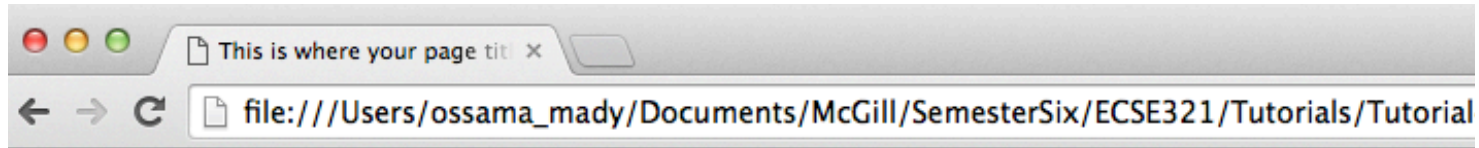
# HTML

A screenshot of a code editor window titled 'Example.html'. The editor has a dark background with light-colored text. The code is an HTML document structure. Line 1: <!DOCTYPE html>. Line 2: <html>. Line 3: <head>. Line 4: <title> This is where your page title goes</title>. Line 5: </head>. Line 6: <body>. Line 7: <pre>. Line 8: This is a preformatted text. Line 9: I can format the text here. Line 10: Line 11: Two breaks. Line 12: Line 13: Line 14: Three breaks. Line 15: </pre>. Line 16: <p>. Line 17: This is not a preformatted text. Line 18: I can format the text here. Line 19: Line 20: Two breaks. Line 21: Line 22: Line 23: Three breaks. Line 24: </p>. Line 25: Line 26: </body>. Line 27: </html>. The code is color-coded: tags are in red, text is in white, and the preformatted text block is in a monospace font. The line numbers 1 through 27 are on the left side of the editor.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> This is where your page title goes</title>
5   </head>
6   <body>
7     <pre>
8       This is a preformatted text.
9       I can format the text here.
10
11       Two breaks.
12
13
14       Three breaks.
15     </pre>
16     <p>
17       This is not a preformatted text.
18       I can format the text here.
19
20       Two breaks.
21
22
23       Three breaks.
24     </p>
25
26   </body>
27 </html>
```



# HTML



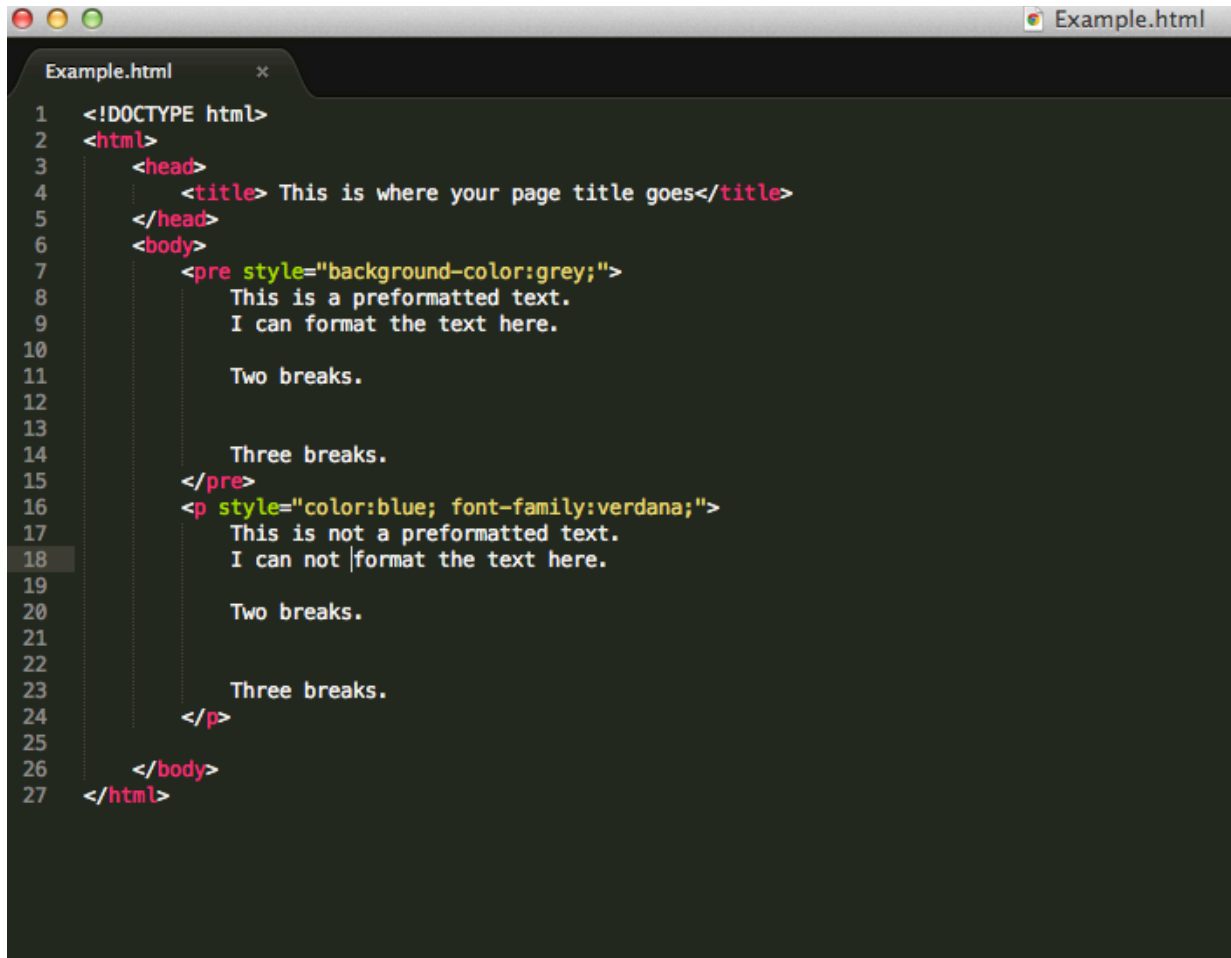
```
This is a preformatted text.  
I can format the text here.
```

```
Two breaks.
```

```
Three breaks.
```

This is not a preformatted text. I can not format the text here. Two breaks. Three breaks.

# HTML (adding some styles)

A screenshot of a code editor window titled "Example.html". The editor shows HTML code with line numbers from 1 to 27. The code defines an HTML document with a title and two paragraphs. The first paragraph is wrapped in a 

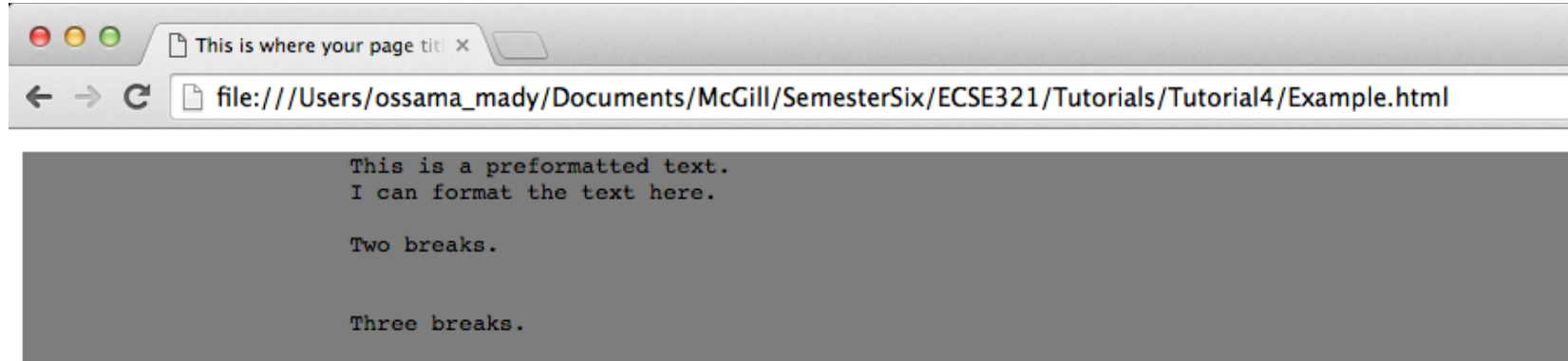
```
 tag with a grey background color. The second paragraph is wrapped in a 

tag with blue color and Verdana font family. Both paragraphs contain text with line breaks.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> This is where your page title goes</title>
5   </head>
6   <body>
7     <pre style="background-color:grey;">
8       This is a preformatted text.
9       I can format the text here.
10
11       Two breaks.
12
13
14       Three breaks.
15     </pre>
16     <p style="color:blue; font-family:verdana;">
17       This is not a preformatted text.
18       I can not format the text here.
19
20       Two breaks.
21
22
23       Three breaks.
24     </p>
25   </body>
26 </html>
```

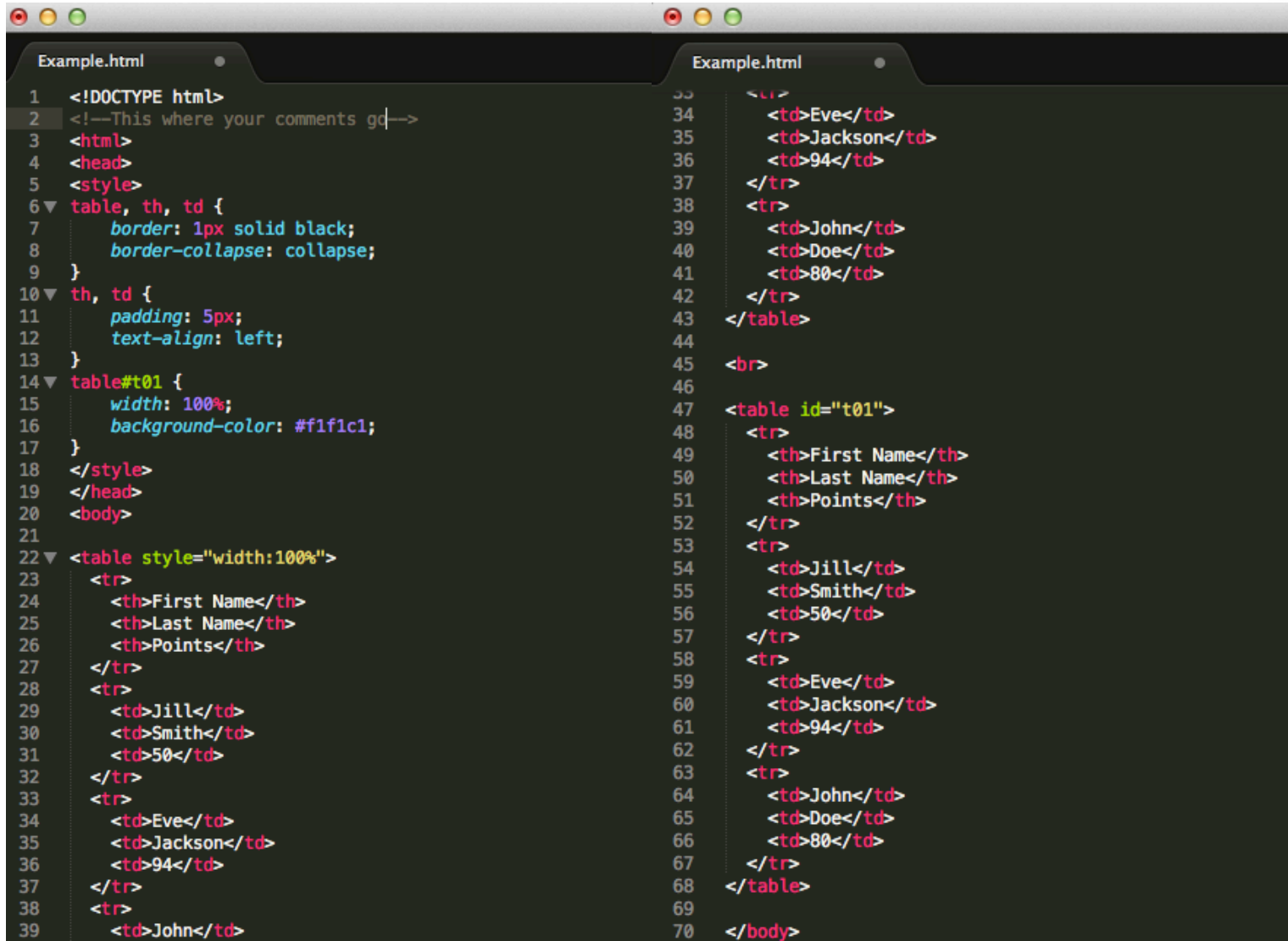

```

# HTML (adding some styles)



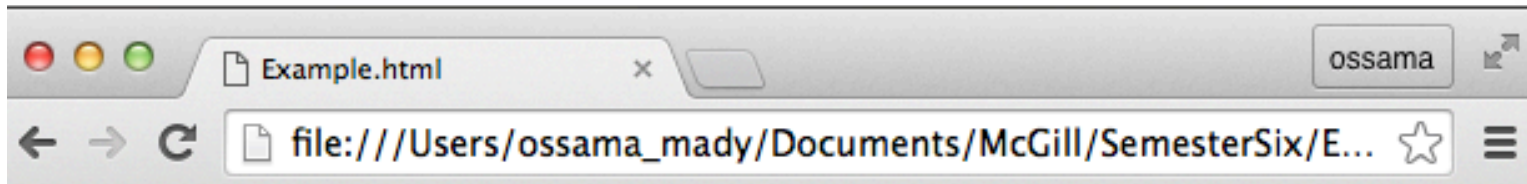
This is not a preformatted text. I can not format the text here. Two breaks. Three breaks.

# HTML



```
Example.html
1 <!DOCTYPE html>
2 <!--This where your comments go-->
3 <html>
4 <head>
5 <style>
6 table, th, td {
7     border: 1px solid black;
8     border-collapse: collapse;
9 }
10 th, td {
11     padding: 5px;
12     text-align: left;
13 }
14 table#t01 {
15     width: 100%;
16     background-color: #f1f1c1;
17 }
18 </style>
19 </head>
20 <body>
21
22 <table style="width:100%">
23     <tr>
24         <th>First Name</th>
25         <th>Last Name</th>
26         <th>Points</th>
27     </tr>
28     <tr>
29         <td>Jill</td>
30         <td>Smith</td>
31         <td>50</td>
32     </tr>
33     <tr>
34         <td>Eve</td>
35         <td>Jackson</td>
36         <td>94</td>
37     </tr>
38     <tr>
39         <td>John</td>
40
Example.html
33 </tr>
34     <td>Eve</td>
35     <td>Jackson</td>
36     <td>94</td>
37 </tr>
38 <tr>
39     <td>John</td>
40     <td>Doe</td>
41     <td>80</td>
42 </tr>
43 </table>
44
45 <br>
46
47 <table id="t01">
48     <tr>
49         <th>First Name</th>
50         <th>Last Name</th>
51         <th>Points</th>
52     </tr>
53     <tr>
54         <td>Jill</td>
55         <td>Smith</td>
56         <td>50</td>
57     </tr>
58     <tr>
59         <td>Eve</td>
60         <td>Jackson</td>
61         <td>94</td>
62     </tr>
63     <tr>
64         <td>John</td>
65         <td>Doe</td>
66         <td>80</td>
67     </tr>
68 </table>
69
70 </body>
```

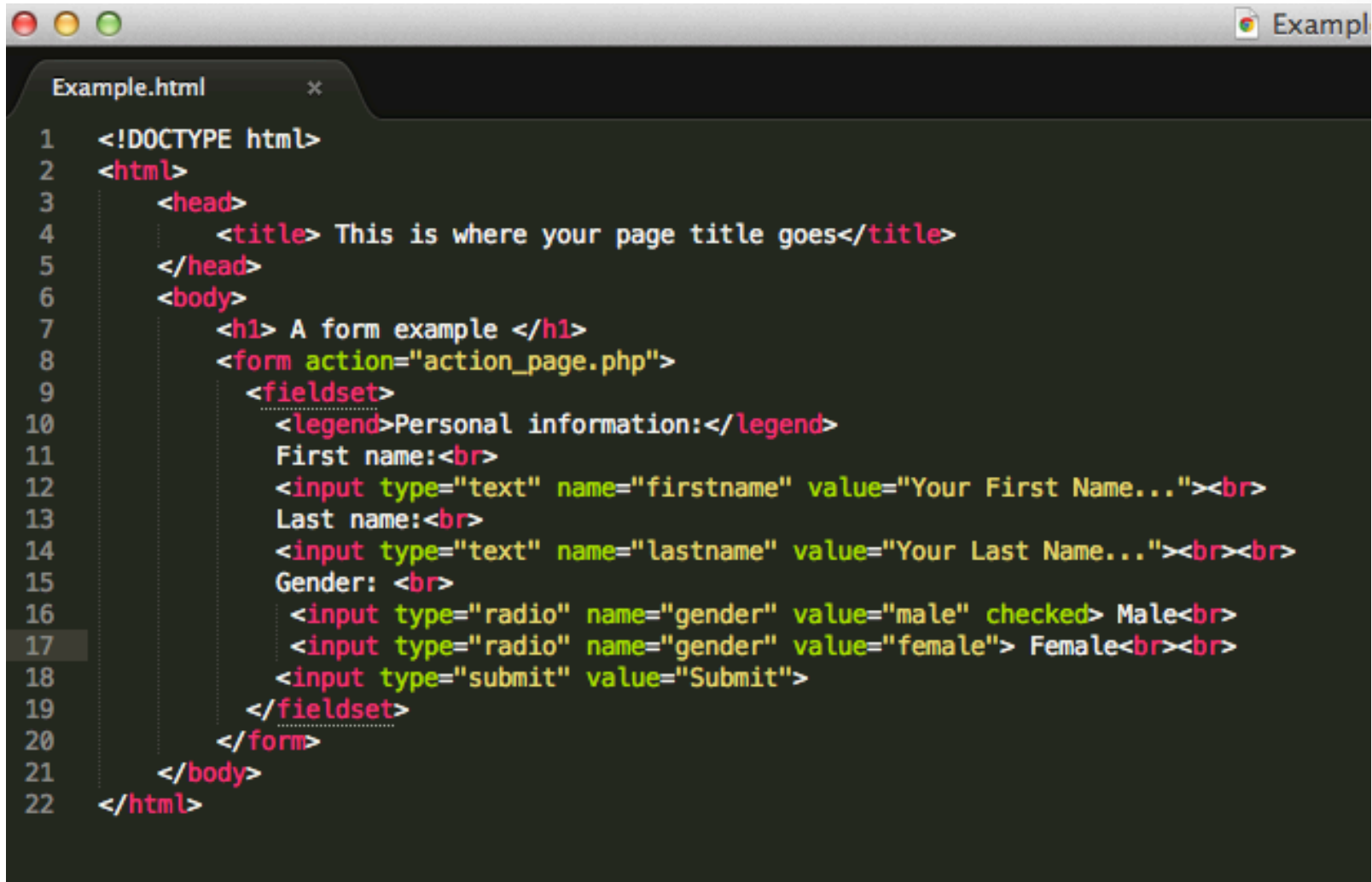
# HTML



First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

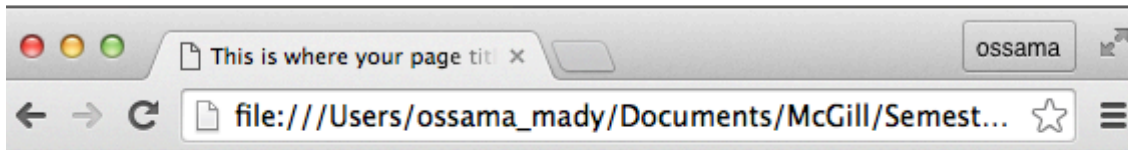
First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

# HTML FORMS

A screenshot of a code editor window titled "Example.html". The editor displays HTML code for a form. The code includes a DOCTYPE declaration, a head section with a title, and a body section with a heading and a form. The form contains a fieldset with a legend "Personal information:" and three input fields: a text field for "First name", a text field for "Last name", and a radio button group for "Gender" with options "Male" (checked) and "Female". A "Submit" button is also present. The code is syntax-highlighted with various colors for tags, attributes, and text.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title> This is where your page title goes</title>
5    </head>
6    <body>
7      <h1> A form example </h1>
8      <form action="action_page.php">
9        <fieldset>
10         <legend>Personal information:</legend>
11         First name:<br>
12         <input type="text" name="firstname" value="Your First Name..."><br>
13         Last name:<br>
14         <input type="text" name="lastname" value="Your Last Name..."><br><br>
15         Gender: <br>
16         <input type="radio" name="gender" value="male" checked> Male<br>
17         <input type="radio" name="gender" value="female"> Female<br><br>
18         <input type="submit" value="Submit">
19       </fieldset>
20     </form>
21   </body>
22 </html>
```

# HTML FORMS



## A form example

Personal information:

First name:

Last name:

Gender:  
☒ Male  
☐ Female

# HTML FORMS

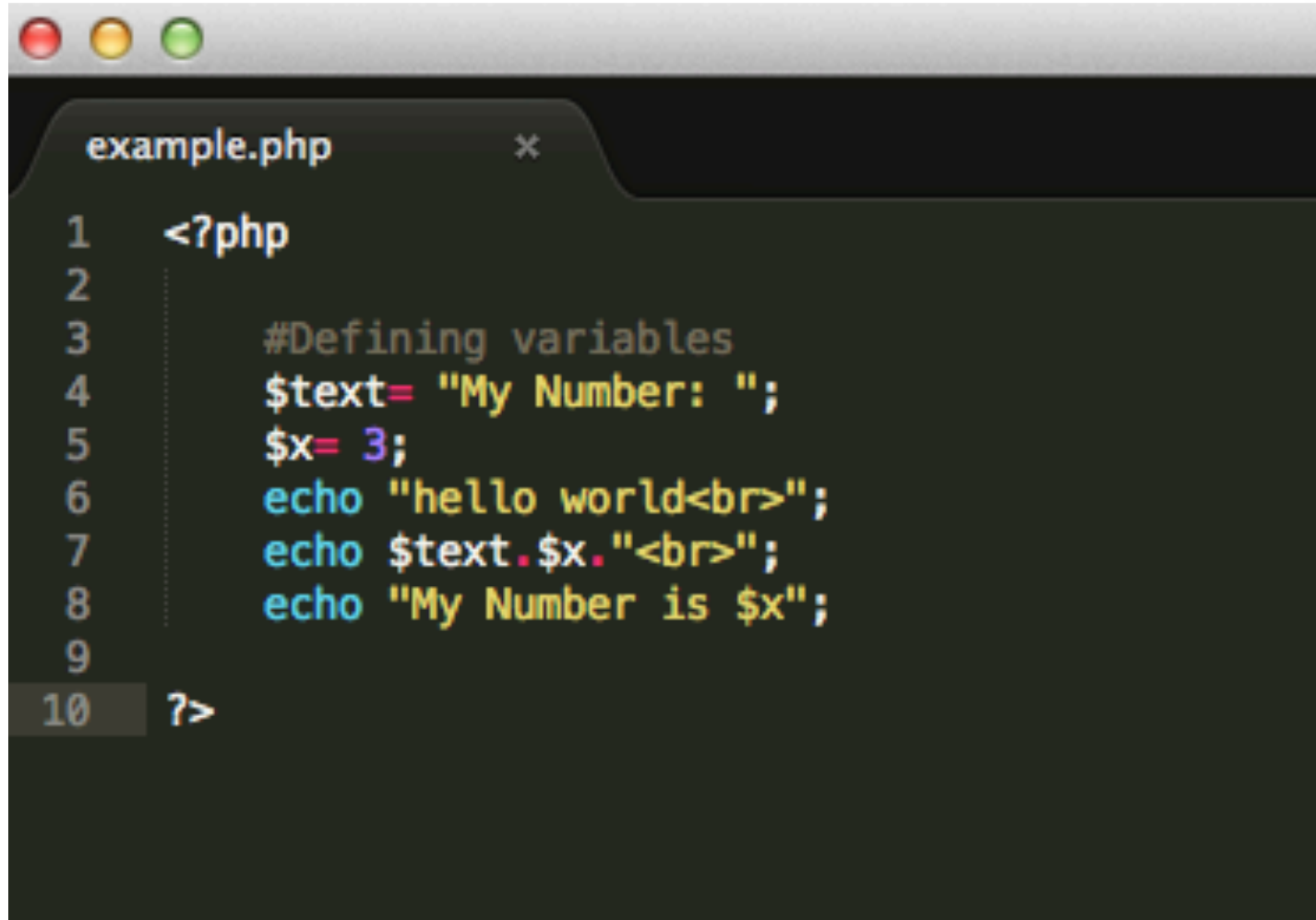
- Text Fields  
`<input name="email" type="text" />`
- Password Fields  
`<input name="password" type="password" />`
- Hidden Fields  
`<input name="id" value="123" />`
- Checkboxes  
`<input checked="checked" name="remember" type="checkbox" />`
- Radio Buttons  
`<input name="gender" type="radio" value="F" />`  
`<input name="gender" type="radio" value="M" />`
- Drop-Down Menus  
`<select name="state">`  
    `<option value=""></option>`  
    `<option value="MA"></option>`  
    `<option value="NY"></option>`  
`</select>`
- Text Areas  
`<textarea name="comments"></textarea>`



# PHP

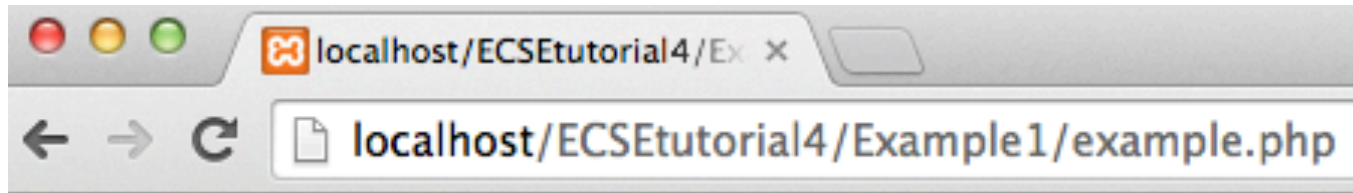
- The PHP HyperText Preprocessor is a programming language that allows web developers to create dynamic content that interacts with databases.
- PHP is a loosely typed language, which means it automatically converts the datatype of the variable depending on its value. You don't specify the data value beforehand like in C, Java..etc
- Any file that contains php code should be stored with the extension .php

# PHP



```
1  <?php
2
3      #Defining variables
4      $text= "My Number: ";
5      $x= 3;
6      echo "hello world<br>";
7      echo $text.$x."<br>";
8      echo "My Number is $x";
9
10  ?>
```

# PHP

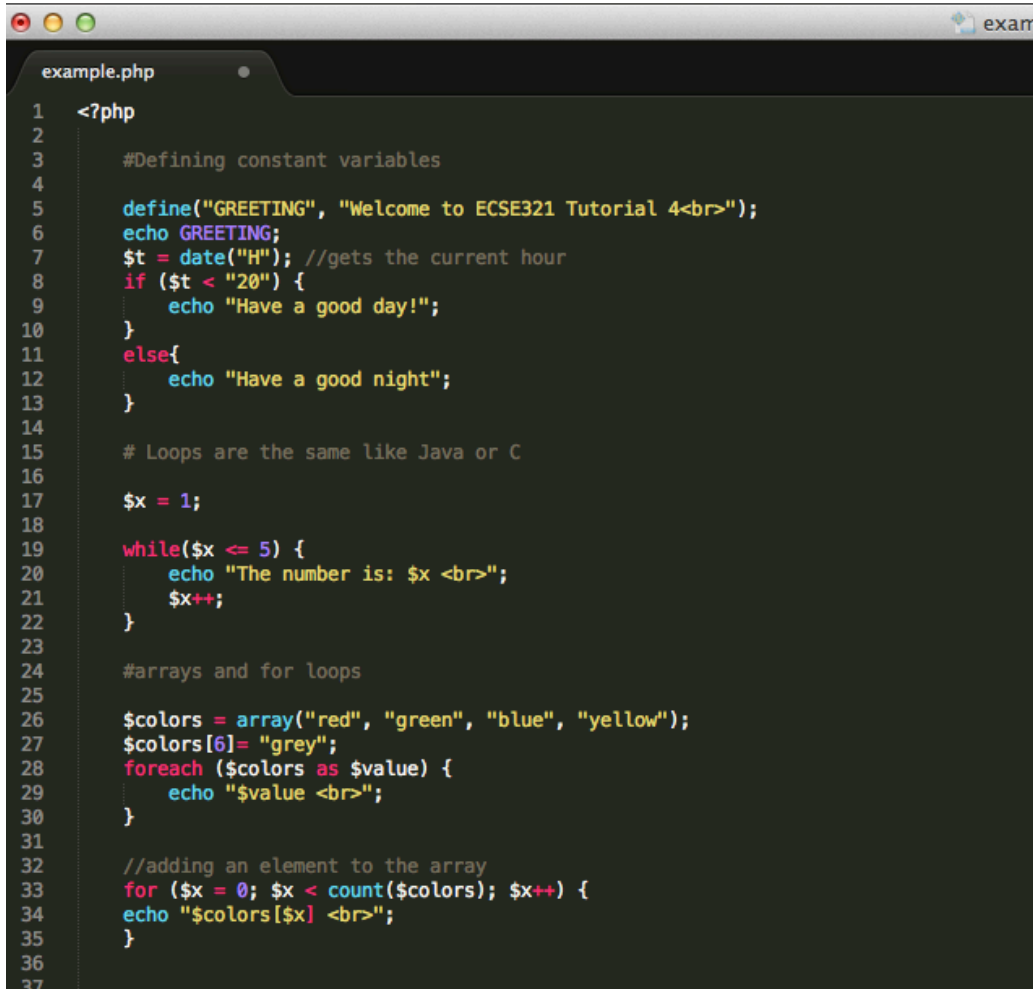


hello world

My Number: 3

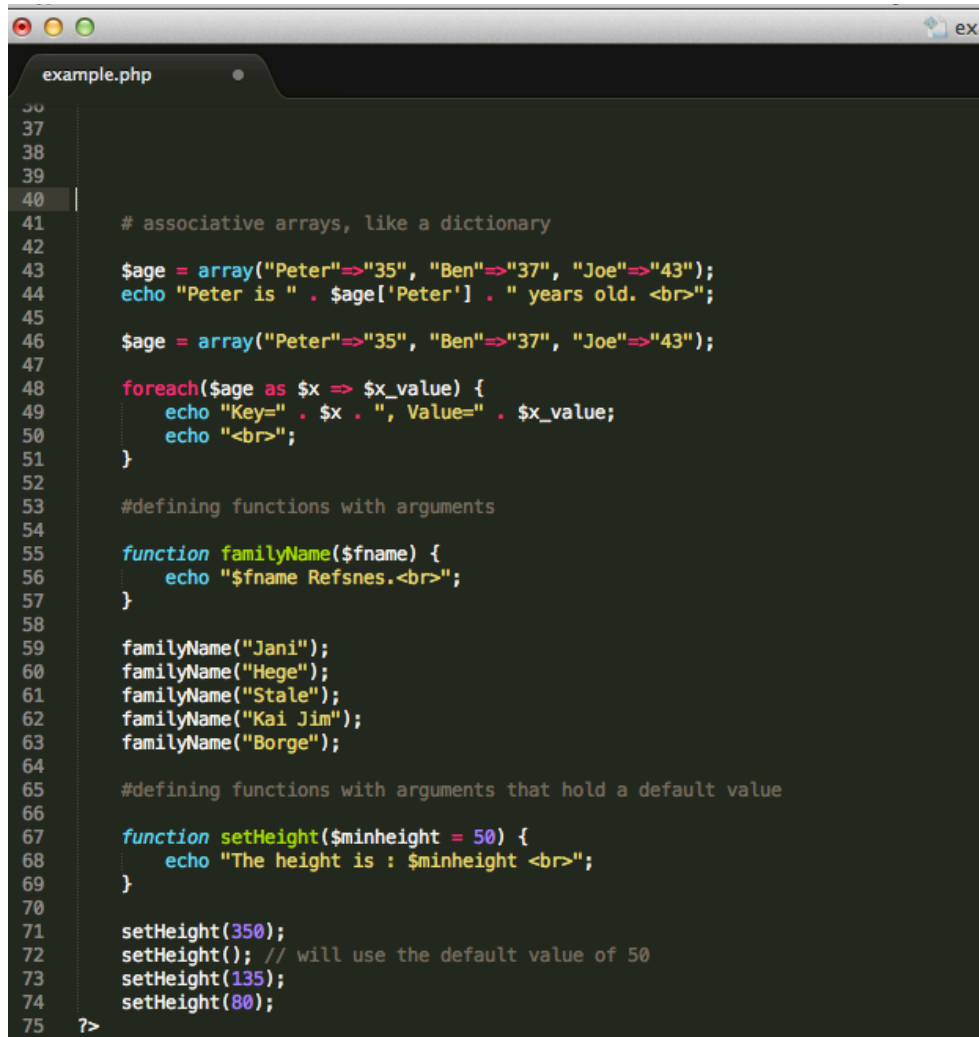
My Number is 3

# PHP



```
1  <?php
2
3  #Defining constant variables
4
5  define("GREETING", "Welcome to ECSE321 Tutorial 4<br>");
6  echo GREETING;
7  $t = date("H"); //gets the current hour
8  if ($t < "20") {
9      echo "Have a good day!";
10 }
11 else{
12     echo "Have a good night";
13 }
14
15 # Loops are the same like Java or C
16
17 $x = 1;
18
19 while($x <= 5) {
20     echo "The number is: $x <br>";
21     $x++;
22 }
23
24 #arrays and for loops
25
26 $colors = array("red", "green", "blue", "yellow");
27 $colors[6]= "grey";
28 foreach ($colors as $value) {
29     echo "$value <br>";
30 }
31
32 //adding an element to the array
33 for ($x = 0; $x < count($colors); $x++) {
34     echo "$colors[$x] <br>";
35 }
36
37
```

# PHP



```
36
37
38
39
40
41 # associative arrays, like a dictionary
42
43 $age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
44 echo "Peter is " . $age['Peter'] . " years old. <br>";
45
46 $age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
47
48 foreach($age as $x => $x_value) {
49     echo "Key=" . $x . ", Value=" . $x_value;
50     echo "<br>";
51 }
52
53 #defining functions with arguments
54
55 function familyName($fname) {
56     echo "$fname Refsnes.<br>";
57 }
58
59 familyName("Jani");
60 familyName("Hege");
61 familyName("Stale");
62 familyName("Kai Jim");
63 familyName("Borge");
64
65 #defining functions with arguments that hold a default value
66
67 function setHeight($minheight = 50) {
68     echo "The height is : $minheight <br>";
69 }
70
71 setHeight(350);
72 setHeight(); // will use the default value of 50
73 setHeight(135);
74 setHeight(80);
75 ?>
```

# PHP (Working With HTML)

```
Example.html x example.php
1 <html>
2 <body>
3 Welcome <?php echo $_GET["firstname"]." ".$_GET["lastname"]; ?> to ECSE321 tutorial 4<br>
4 </body>
5 </html>
6
7 <!-- or you can use this -->
8 <?
9 echo "<html>
10 <body>
11 Welcome " . $_GET["firstname"]." ".$_GET["lastname"]." to ECSE321 tutorial 4
12 </body>
13 </html>"
14 ?>
15
```

# PHP (form validation)

- How to Validate your input fields ?
- Form Validation Example, can be found on <https://github.com/shabbir-hussain/ecse321tutW16>  
Under Tutorial4files/example5

# JSON and XML

- XML and JSON are used to exchange and store data for later use.
- They just carry the data in a specified form, they DO NOT present it like HTML does.
- JSON files extension is .json
- XML files extension is .xml



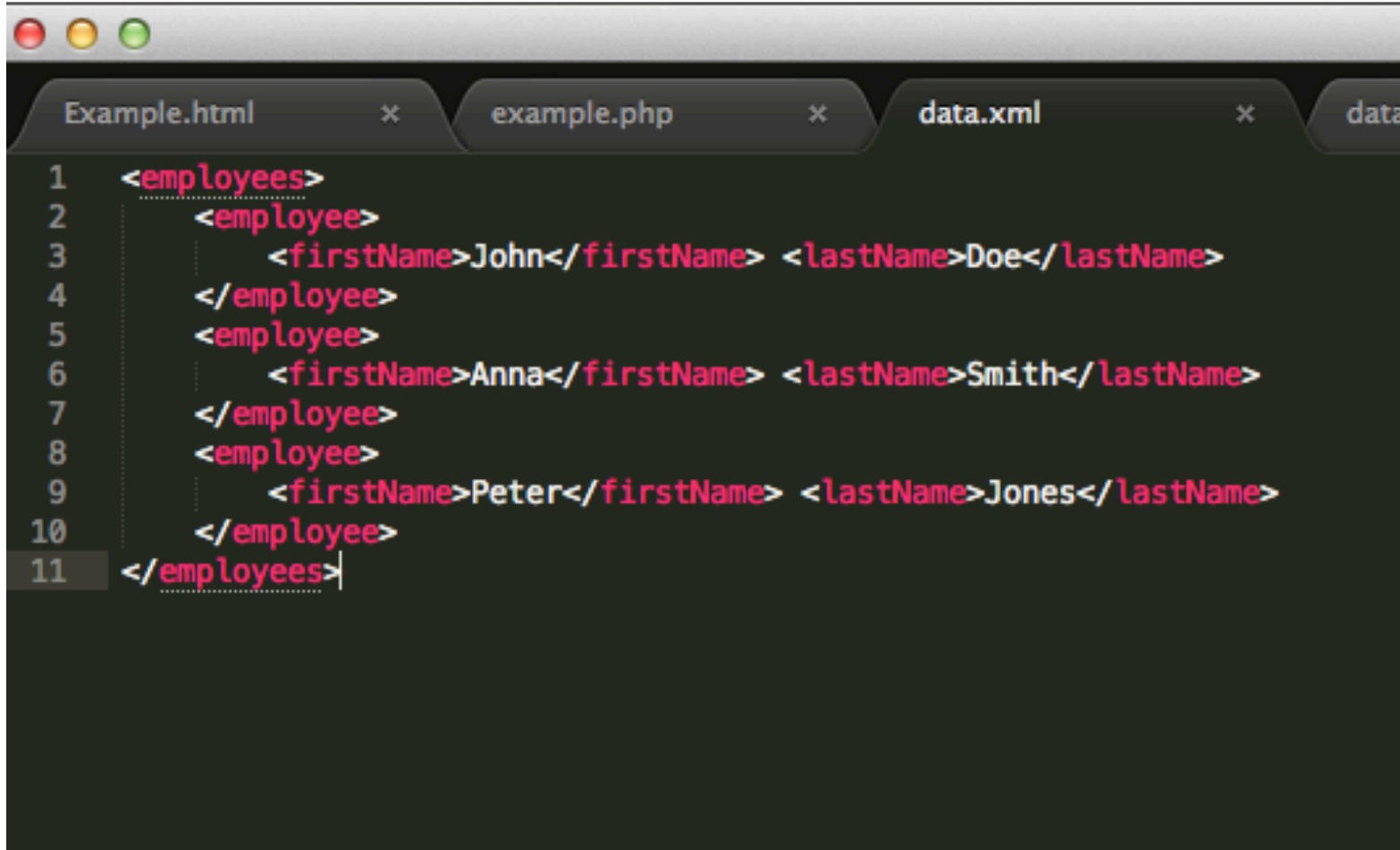
# JSON



The image shows a code editor window with four tabs: Example.html, example.php, data.xml, and data.json. The data.json tab is active, displaying a JSON object with an array of employee data. The code is as follows:

```
1 {"employees": [  
2   {"firstName": "John", "lastName": "Doe"},  
3   {"firstName": "Anna", "lastName": "Smith"},  
4   {"firstName": "Peter", "lastName": "Jones"}  
5 ]}
```

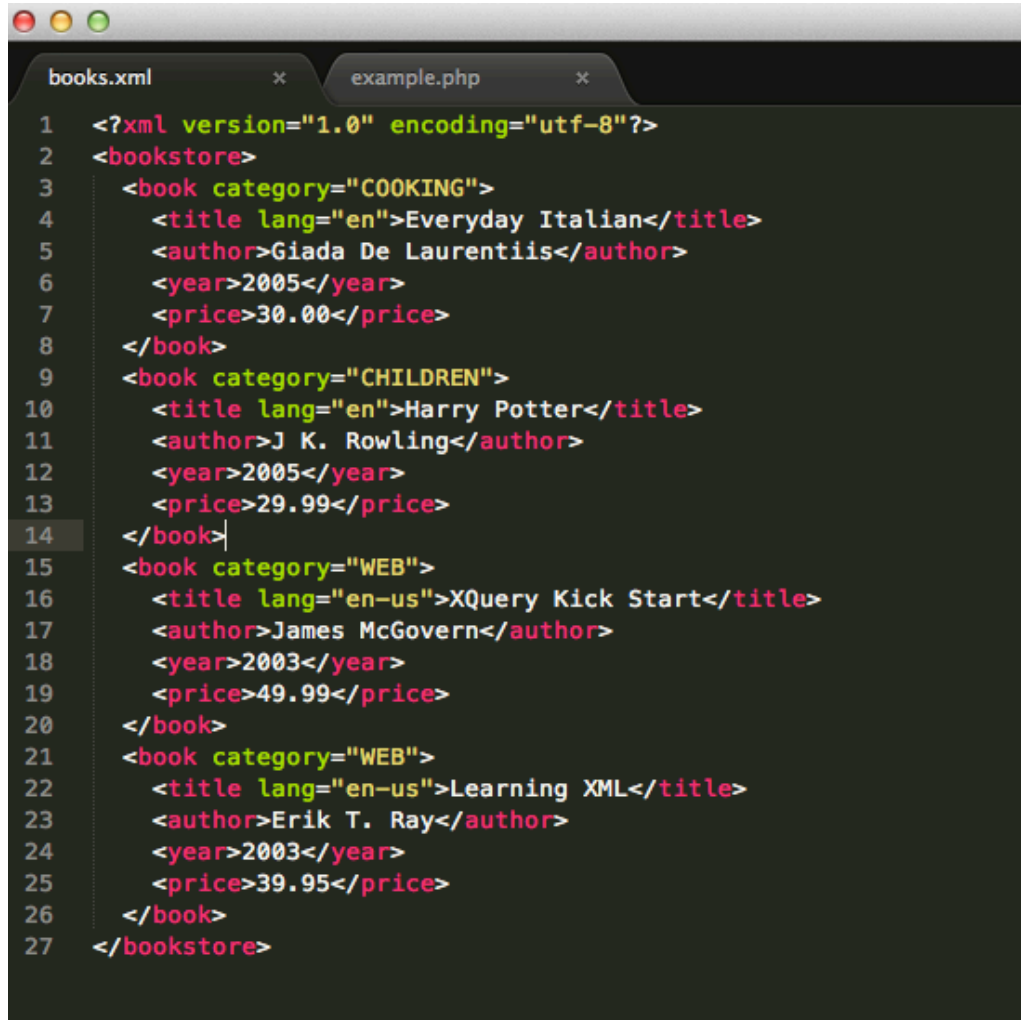
# XML



A screenshot of a code editor window with a dark theme. The window has three tabs: 'Example.html', 'example.php', and 'data.xml'. The 'data.xml' tab is active, showing XML code. The code is as follows:

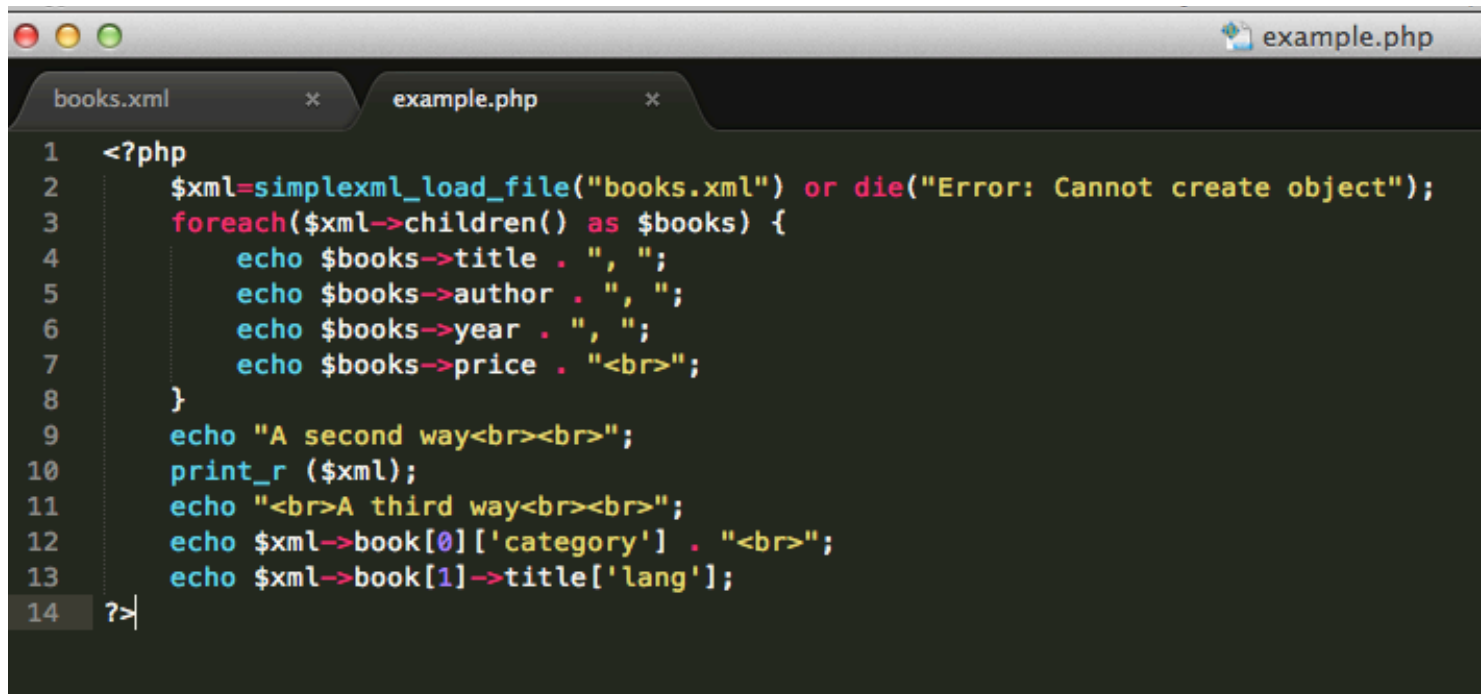
```
1 <employees>
2   <employee>
3     <firstName>John</firstName> <lastName>Doe</lastName>
4   </employee>
5   <employee>
6     <firstName>Anna</firstName> <lastName>Smith</lastName>
7   </employee>
8   <employee>
9     <firstName>Peter</firstName> <lastName>Jones</lastName>
10  </employee>
11 </employees>
```

# PHP with XML



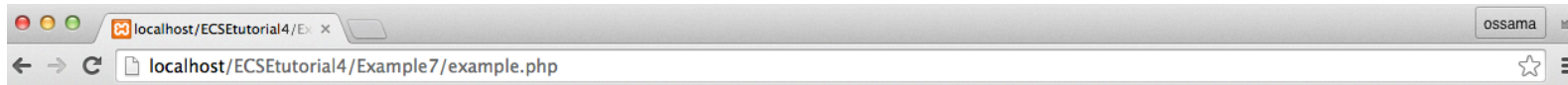
```
books.xml x example.php x
1  <?xml version="1.0" encoding="utf-8"?>
2  <bookstore>
3      <book category="COOKING">
4          <title lang="en">Everyday Italian</title>
5          <author>Giada De Laurentiis</author>
6          <year>2005</year>
7          <price>30.00</price>
8      </book>
9      <book category="CHILDREN">
10         <title lang="en">Harry Potter</title>
11         <author>J K. Rowling</author>
12         <year>2005</year>
13         <price>29.99</price>
14     </book>
15     <book category="WEB">
16         <title lang="en-us">XQuery Kick Start</title>
17         <author>James McGovern</author>
18         <year>2003</year>
19         <price>49.99</price>
20     </book>
21     <book category="WEB">
22         <title lang="en-us">Learning XML</title>
23         <author>Erik T. Ray</author>
24         <year>2003</year>
25         <price>39.95</price>
26     </book>
27 </bookstore>
```

# PHP with XML

A screenshot of a code editor window with two tabs: 'books.xml' and 'example.php'. The 'example.php' tab is active, showing PHP code that uses SimpleXML to load and parse an XML file. The code is color-coded: PHP tags are blue, strings are yellow, and keywords are red. The code includes error handling with 'die()' and uses 'foreach' to iterate through XML children. It also demonstrates direct array access on the SimpleXML object.

```
1  <?php
2      $xml=simplexml_load_file("books.xml") or die("Error: Cannot create object");
3      foreach($xml->children() as $books) {
4          echo $books->title . ", ";
5          echo $books->author . ", ";
6          echo $books->year . ", ";
7          echo $books->price . "<br>";
8      }
9      echo "A second way<br><br>";
10     print_r ($xml);
11     echo "<br>A third way<br><br>";
12     echo $xml->book[0]['category'] . "<br>";
13     echo $xml->book[1]->title['lang'];
14  ?>
```

# PHP with XML



Everyday Italian, Giada De Laurentiis, 2005, 30.00  
Harry Potter, J K. Rowling, 2005, 29.99  
XQuery Kick Start, James McGovern, 2003, 49.99  
Learning XML, Erik T. Ray, 2003, 39.95  
A second way

SimpleXMLElement Object ( [book] => Array ( [0] => SimpleXMLElement Object ( [attributes] => Array ( [category] => COOKING ) [title] => Everyday Italian [author] => Giada De Laurentiis [year] => 2005 [price] => 30.00 ) [1] => SimpleXMLElement Object ( [attributes] => Array ( [category] => CHILDREN ) [title] => Harry Potter [author] => J K. Rowling [year] => 2005 [price] => 29.99 ) [2] => SimpleXMLElement Object ( [attributes] => Array ( [category] => WEB ) [title] => XQuery Kick Start [author] => James McGovern [year] => 2003 [price] => 49.99 ) [3] => SimpleXMLElement Object ( [attributes] => Array ( [category] => WEB ) [title] => Learning XML [author] => Erik T. Ray [year] => 2003 [price] => 39.95 ) )

A third way

COOKING  
en

# Pair Programming EXERCISE

- Create a registration page which takes the user's first name, last name, email and password. (to be stored in xml file)
- Create a login page which takes the user's email and password, and matches them with the data stored in the xml file. If no match was found, an error message should be given to the user.