NSWI142 Web Applications – Final Test

Name: *Stanley Hardworker* 24.12.2016

The questions does not have single correct answer. In fact, any combination of correct answers is possible, even an empty set. Please select only those answers which are completely correct. Formally, the correctness of the results is determined with respect to HTML5, CSS3, ECMAScript 5, and PHP 5.3 (or newer, but not PHP 7) specifications and web APIs, which are implemented in current versions of Firefox and Chrome.

Each question counted as either completely correct or completely wrong unless specified otherwise. You can receive up to 100 points from the test and additional points from the home assignments are added to your result. You may recieve up to 15 additional bonus points; however, there is no limit in case of negative points. The total amount of points will determine your grade as follows:

- 91 points or more: excellent (mark 1),
- 90 – 76 points: well done (mark 2),
- 75 – 60 points: OK (mark 3),
- 59 points or less: failed (mark 4).

---

1. *Consider the following example of HTML5 code. Select all correct statements below.* [3 points]

```
<article><h1>Heading A</h1>
  <section><h1>Heading B</h1>
    <section><h1>Heading C</h1></section>
    <section><h1>Heading D</h1></section>
  </section>
</article>
```

☐ Heading A, B, C a D are first-level headings, since they all use element <`h1`>.

☒ The only first-level heading is "Heading A". "Heading B" is second-level heading and C and D are third-level headings. The level is determined by <`section`> element.

☐ The browser will report an error, because the HTML fragment is not valid.

2. *Consider the following HTML form. Add/modify its code, so that the content of the form is sent encoded in URL query to* ***`http://example.cz/process.php`*** *script once button "Send" is clicked.* [4 points]

```
<form method="get" action="http://example.cz/process.php">
  Name: <input type="text" name="firstName" /><br/>
  Surname:  <input type="text" name="surname" /><br/>
  <input type="submit" value="Send" />
</form>
```

3. *Sketch a result of the following HTML code as it would appear in the browser.* [5 points]

```
<table>
  <tr>
    <td colspan="2">Adult</td><td>2</td>
  </tr>
  <tr><td>A1</td><td>34</td>
    <td rowspan="2">2</td></tr>
  <tr><td>A2</td><td>32</td></tr>
  <tr>
    <td colspan="2">Child</td><td>3</td>
  </tr>
  <tr><td>C1</td><td>4</td><td>1</td></tr>
  <tr><td>C2</td><td>8</td>
    <td rowspan="2">2</td></tr>
  <tr><td>C3</td><td>12</td></tr>
</table>
```

<table>
<tr><td colspan="2">Adult</td><td>2</td></tr>
<tr><td>A1</td><td>34</td><td rowspan="2">2</td></tr>
<tr><td>A2</td><td>32</td></tr>
<tr><td colspan="2">Child</td><td>3</td></tr>
<tr><td>C1</td><td>4</td><td>1</td></tr>
<tr><td>C2</td><td>8</td><td rowspan="2">2</td></tr>
<tr><td>C3</td><td>12</td></tr>
</table>

4. *What is the purpose/behaviour of the <`input`> HTML element with `type="hidden"` attribute?* [4 points]

☐ The `hidden` value is not allowed in the `type` attribute.

☐ It defines an input field of the form which is not initially visible to the user. The browsers display a small thematic icon instead of the hidden input fields and these fields are shown to the uses once the icon is clicked on.

☒ It defines an input field which is never visible to the user and the user cannot modify its value. The value is directly encoded in the HTML code, but it may be change by Javascript at client side.

5. *Select all fragments of code that **are not** valid according to HTML language specification.* [4 points]

☐ `<table id=1><tr><td>Martin</td></tr><tr><td>Newman</td></tr></table>`

☒ `<table id="1"><tr><td>Martin</tr></td><tr><td>Newman</tr></td></table>`

☐ `<table id="1"><tr><td>Martin<tr><td>Newman</table>`

6. *Consider the following fragment of HTML code. Find (possibly the shortest) CSS selector that would target all phone numbers inside a table cell.* [4 points]

```
<ul>
  <li class="phone">777000000</li>
</ul>
<table>
  <tr><td>Phone: </td><td><span class="phone">7771111111</span>
    (alternative: <i class="phone">777222222</i>)</td></tr>
  <tr><td>E-mail:</td><td class="email">css@guru.cz</td></tr>
</table>
```

Selector: *td .phone*

7. *Consider CSS selector "`.phone`". Select all statements that are true.* [3 points]

☒ "`.phone`" is equivalent to "`[class~=phone]`".

☐ "`.phone`" is equivalent to "`[class=phone]`".

☐ "`.phone`" functionality cannot be expressed (emulated) by another selector type.

8. *Which statements are true for the following CSS selector?* [4 points]

```
tr:not(:nth-child(-n+1)):not(:nth-last-child(-n+1))
```

☐ The selector targets every odd row of every table.

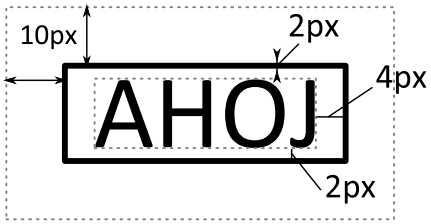
☐ The selector targets the first and the last row of every table.

☒ The selector targets every row of every table except the first and the last row.

☐ The selector targets an empty set since it is a contradiction.

9. *Consider the following CSS rule applied on an element <`div`>`Hello!`</`div`>. Depict, how the element will be rendered by a browser. Also include measurements of the most important distances and sizes in your sketch (in pixels).* [4 points]

```
div {
  margin: 10px;
  padding: 2px 4px 2px 4px;
  border: 2px solid black;
}
```



10. *Write down a CSS rule, which will move the text "Hello" in the following HTML fragment 5px upward from its normal position.* [5 points]

```
<div>Let us say <span class="fiveUp">Hello</span> to everybody</div>
```

CSS (including selector): `.fiveUp { position: relative; top: -5px; }`

11. *Select true statements about ECMAScript variables:* [2 points/answer]

- ☐ They have fixed data type, which is determined by the type of the value assigned to them when they are declared.
- ☒ They are declared by the `var` keyword.
- ☒ They does not allocate/provide memory for their value. Instead, they create a mnemonic association to an existing value, which was created as a literal or as a result of an expression.

12. *Consider the following fragment of ECMAScript. Which of the following statements will hold after the script is executed.* [5 points]

```
var foo = 1;
function barA() { foo = 2; }
function barB() { var foo = 3; }
barA();
barB();
```

- ☐ There is exactly one variable `foo` (in the global scope) with value 3.
- ☒ There is one global variable `foo` (with value 2) and one local variable `foo` inside a closure of the invocation of the function `barB()`; hence, it is not accessible from the rest of the script.
- ☐ There is one global variable `foo` (with value 1) and one local variable `foo` inside a closure of the invocation of the function `barB()`; hence, it is not accessible from the rest of the script.
- ☐ There is one global variable `foo` (with value 2) and one local variable `foo` associated with function `barB()` and since functions are also objects, the `foo` also appears as a property of object `barB`.
- ☐ The global variable `foo` is replace by local counterpart once function `barA()` is invoked. After that, the variable `foo` ceases to exist, since there are no references inside the closure of `barA` and thus the whole closure is released.

13. *Consider the following fragment of code. If a mouse click event occurs whilst the cursor is above element "**div2**", but not above "**div3**", the browser may display one or more alert messages. Select all message sequences which are theoretically possible.* [4 points]

```
<div id="div1">
  <div id="div2">
    <div id="div3">
      ...
</div></div></div>
<script type="text/javascript">
  document.getElementById("div1").onclick = function() {
    window.alert("div1 clicked");
  };
  document.getElementById("div2").onclick = function() {
    window.alert("div2 clicked");
  };
  document.getElementById("div3").onclick = function() {
    window.alert("div3 clicked");
  };
</script>
```

☐ "div2 clicked"

☐ "div1 clicked", "div2 clicked"

☐ "div1 clicked", "div2 clicked", "div3 clicked"

☐ "div1 clicked", "div2 clicked", "div3 not clicked"

☒ "div2 clicked", "div1 clicked"

☐ "div3 clicked", "div2 clicked", "div1 clicked"

☐ "div1 clicked", "div2 clicked", "div1 clicked"

14. *Consider the following fragment of HTML code. Write a Javascript fragment that will set the background color of the <`div`> element with `id="text"` to red using CSS properties.* [5 points]

```
<div id="text"> ... </div>
```

JavaScript: `document.getElementById("text").style.backgroundColor = "red";`

15. *Consider the following scenario: a web page contains a HTML form, which is supposed to insert new record into the database. The form is processed at the client-side by a Javascript code and its data are sent over to the server via asynchronous HTTP request (AJAX), where they are processed by a PHP script. Which of the following statements are true, if the whole solution is implemented correctly and in compliance with HTTP specification.* [2 points/answer]

☒ The asynchronous HTTP request must use POST (or possibly PUT) method.

☐ The PHP script must generate a HTML page, which is displayed after the form data are processed.

☐ The form data must be formatted in XML or in JSON.

☐ The server-side script must not save the data into the database. It can only verify that the data are valid. The form must be then submitted in a traditional way (using its submit event).

☒ The user may fully interact with the page while the asynchronous request is being processed. It may even happen that a browsing action is initiated (e.g., by clicking on a hyperlink or navigating backwards in history).

16. *Which of the following properties are true for HTTP 1.1?* [3 points]

☐ Each request requires a new TCP connection and the connection is terminated once the request is served.

☒ The protocol is stateless, each request is processed regardless of the previous and concurrent requests.

☒ The protocol supports partial queries (which request only a byte range of the target contents), which can be used to resume interrupted downloads, for instance.

17. *Which of the following data types exist in PHP?* [1 point/answer]

☐ number ☒ integer ☒ string ☐ function ☒ boolean ☒ array

18. *String literals in PHP may be written in quotes ("") or apostrophes ('').* [2 points/answer]

☐ There is no difference between both syntaxes, programmers may use any of them at their convenience.

☐ There is no difference between both syntaxes, but the literals in apostrophes were denoted as deprecated and should not be used any more.

☒ The literals in quotes replace escape sequences (e.g., `\n`, `\t`, ...) with their corresponding special characters.

☒ The literals in quotes replace variables (written in PHP syntax) with their respective contents (converted to string).

19. *A* ***index.php*** *script is processing a HTTP request with the following URL. Answer th questions below.* [3 points/answer]

```
index.php?page=welcome&offset=10&item[0]=a&item[1]=b&item[2]=c&last=
```

- How many items does the `$_GET` array have? *4*
- What is the type of `$_GET['item']` ? *array*
- What is the type and value of `$_GET['last']` ? *string("")*

20. *Consider the following fragment of interleaved HTML and PHP code. Outline the output of the PHP interpret (which will be received by the browser). Exact whitespace placement and formating is not important.* [4 points]

```
<html><body>
  <ul>
  <?php
    $data = array("first", "second", "third");
    for ($i = 1; $i < count($data); ++$i) { ?>
      <li><?= $data[$i]; ?></li>
  <? } ?>
  </ul>
</body></html>
```

```
<html><body>
  <ul>
    <li>second</li>
    <li>third</li>
  </ul>
</body></html>
```

Total _____ points received from the maximum of 100. Final grade: _____.