# Web Technologies: Midterm exam topics

Note: This collection of topics/questions doesn't cover the entire pool of topics, but rather serves for demonstration purpose.

## 1. Basics: Protocols & Tech-stack

a. Explain underlying mechanisms and protocols that are needed to issue an HTTP GET request to a web server.

b. What does `keep-alive` mean in the context of HTTP and in which version was it first usable?

c. What is the purpose of HTTP-headers and which information can be conveyed by them? List two concrete examples.

d. Explain the concept of status codes in HTTP. List two status codes and explain their meaning.

## 2. HTML & CSS

a. Answer the following questions with `true` / `false`.

```
i. HTML is a touring complete programming language.
ii. HTML files need to be created in a hex-editor.
iii. HTML received some modernisation with the release of HTML5. This is
the currently used version of HTML.
iv. Tags in HTML can be self-closing. That means that there is no
dedicated closing tag to an opening tag.
```

b. Create a bare-bones HTML page with the following content:

- First-order heading with the text "HTML is fun".
- Paragraph with arbitrary text
- Link with the title "Search engine" that opens the URL "https://duckduckgo.com/" in a new tab.

c. Formulate a tag to embed the file `sunset.png` in the file `view.html` according to the given file structure below.

```
index.html
view.html
css
    |-> main.css
    |-> mobile.css
js
    |-> calculate.css
img
    |-> sunset.png
```

d. Write down a tag to include the file `mobile.css` according to the file structure in question c.

e. Explain the CSS box model and outline differences between inline- and block elements.

f. Given the following `body` on a page:

```
<body>
    <header>
        <h1>My Personal Blog</h1>
        <p>... by Jon Doe</p>
        <div id="navbar">
            <a class="nav-item" href="index.html">
                Home
            </a>
            <a class="nav-item" href="about.html">
                About
            </a>
            <a class="nav-item" href="portfolio.html">
                Portfolio
            </a>
        </div>
    </header>

    <!-- main content -->
    <div>
        <p class="blogpost">
            <h3>First post</h3>
            <p>Some content here</p>
            <!-- post tags -->
            <span>interesting</span>
            <span>new</span>
            <span>life-advice</span>
        </p>
        <p class="blogpost">
            <h3>Second post</h3>
            <p>Some content here</p>
            <!-- post tags -->
            <span>interesting</span>
            <span>new</span>
            <span>life-advice</span>
        </p>
    </div>
</body>
```

TODO: rewrite HTML, translate question

## 3. Basic JavaScript

a. Explain the difference between `var` and `let` in JavaScript

**Listing 1**

```
    <h1>ToDolist-app</h1>
    <input type="text" placeholder="Enter todo item here .." id="new-todo-
txt" label="new-todo-txt">
    <input type="button" value="Add to list" id="add-item-btn">
    <input type="button" value="Clear list" id="clear-list-btn">

    <div id="todo-list">
        <div class="item">
            <p>Text of the item goes here</p>
            <input type="button" value="Delete item" class="delete-item-
btn">
        </div>
    </div>
```

b. Write down a tag you would use to include a JavaScript file that exists in `js/custom/cms.js`.

c. Implement the functionality of the button with id `add-item-btn`. Clicking on the button should grab the text from element with id `new-todo-txt` and insert it into the `div` with id `todo-list`. Inside `todo-list`, there is already a sample item in listing 1 that shows you what the inserted item should look like. Also, the text in `new-todo-txt` should be cleared after the item has been added.

d. Ensure that a click on `delete-item-btn` deletes the respective ToDo-item and `clear-list-btn` clears the entire ToDo-list.

TODO: question e onwards

## 4. Advanced JavaScript

## 5. node.js

a. Answer the following questions with `true` / `false`.

```
i. Node.js transpiles JS code during execution to C code, so that it can
get executed by the browser.
ii. Node.js is usually used for implementing application backend systems.
iii. Google Chrome and node.js use the same JavaScript engine.
iv. The tool `npm` is responsible for managing external packages and
dependencies in node applications.
v. Node.js can also be used without `npm.
```

b. Write a web service for management of ToDo lists that contains the following functionality:

- Accessible through port 8080
- `/` returns the text `ToDo list service is up and running`
- Create, change name and delete ToDo-lists
- Create, change & delete elements of ToDo-lists
- Retrieve all elements of a given ToDo list
- Retrieve all elements of all ToDo lists

- Pay attention to not hurt any of the REST principles when designing your API!