# AJAX | CLASSWORK

## Basic Exercises

### Exercise 1

Create a simple web page that has a button and an empty <div>. Give the button an id so you can set up what happens when it's clicked. Also, give the empty <div> an id of "result". When the button is clicked, it will make an AJAX request to load content from a .txt file and show that text inside the <div>

Create a .txt file named example.txt in the same directory as your HTML file. Insert some simple text content into it, which will be displayed when the button is clicked.

### Exercise 2

Extend Exercise 1 by adding another button with its own unique id. This new button, when clicked, will use AJAX to retrieve the current date and time from a PHP file, displaying this information in a new separate <div>.

Create a PHP file named datetime.php in the same directory as your HTML file. This PHP file should output the current date and time. The AJAX call triggered by the second button's click will fetch this information and show it in the new div.

## Intermediate Exercise

For this exercise, you'll create a songs.json file filled with details on at least 20 of your favorite songs, including the title, artist, genre, country, and year. Here's an example of how your JSON file might look like:

```
[
  {
    "title": "Bohemian Rhapsody",
    "artist": "Queen",
    "genre": "Rock",
    "country": "United Kingdom",
    "year": "1975"
  },
  {
    "title": "Imagine",
    "artist": "John Lennon",
    "genre": "Pop",
    "country": "United Kingdom",
    "year": "1971"
  },
…
]
```

After that, make a simple webpage where you'll show your songs. You can make it look nice with some basic design tools like Bootstrap or just HTML and CSS.

Use AJAX to load your song list from the songs.json file without needing to refresh the page. When someone visits your page, they should be able to see all the song details appear automatically.

## Challenge

For the challenge you're going to build a signup form on your website. Include an email field that checks, in real-time, whether the entered email already exists in your database. To do this, set up an AJAX request that triggers as the user types in the email field. This AJAX call will query a PHP script on your server which then looks into the database for that email. If the email exists, display a message directly below the email field saying, "This email is already in use" – all of this without the user having to click a submit button.

Remember, before jumping into coding, make sure your database is all set up and ready to go.

CodeFactory Vienna

– We create developers! –