

Summer 2019 / Web Class
Remote Learning Assignment - Week 4

Assignment 1: Callback Function

Complete the function below to show a delayed result in console.

```
function delayedResult(n1, n2, delayTime, callback){  
    // your code here  
}  
delayedResult(4, 5, 3000, function(result){  
    console.log(result);  
}); // 9 (4+5) will be shown in the console after 3 seconds  
delayedResult(-5, 10, 2000, function(result){  
    window.alert(result);  
}); // 5 (-5+10) will be shown in an alert dialog after 2 seconds
```

Hint: You should use [window.setTimeout\(\)](#) for time scheduling.

Summer 2019 / Web Class
Remote Learning Assignment - Week 4

Assignment 2: Callback Function and Advanced HTML DOM

Complete the functions below to make AJAX call to a URL by GET method, and show the response data in the page. You may render UI with any style.

```
function ajax(src, callback){
    // your code here
}
function render(data){
    // your code here.
    // document.createElement() and appendChild() methods are preferred.
}
ajax("https://cwpeng.github.io/live-records-samples/data/products.json", function(response){
    render(response);
}); // you should get product information in JSON format and render data in the page
```

Reminder: You cannot connect to arbitrary URL because of [Same-Origin Policy](#) which is implemented by browser. So, feel free to use the URL provided by me for testing, otherwise, you can refer to an advanced topic: [CORS](#).

Summer 2019 / Web Class
Remote Learning Assignment - Week 4

Assignment 3: Re-Implement Layout by React (Front-End Only)

Re-implement what we have done in week 1, and assignment 4 in week 2 by React thinkings and rules. You may like to follow the steps below:

1. Prepare your React project. To be simple, you can find and download Minimal HTML Template in the [official getting started page](#).
2. Re-create RWD page layout, the same as what we have done in week 1.
3. Re-add effects to the page, the same as what we have done in assignment 4 in week 2. Follow React rules to handle events and manipulate HTML DOM by setting state.
4. You should test your page on a local server or Github Page.

In the HTML file, you can add only single <div> in <body>, any other HTML elements should be rendered dynamically by React Components.

Reminder: Don't hesitate to call for help if you get stuck in something more than one day.

Summer 2019 / Web Class
Remote Learning Assignment - Week 4

Assignment 4: Build a Website with MySQL (Backend Only)

You already have experience of building an HTTP server by Node.js and Express. Now, it's time to add a MySQL database server to your website for a simple member sign-up and sign-in features.

1. Build a simple HTTP server by Node.js and Express as usual.
2. Setup MySQL server and database schema:
 - a. Install MySQL server in your computer.
 - b. Create a database named assignment.
 - c. Create a user table named user.
 - d. User table should contain at least 3 columns: id, email, password.
 - e. Column id should be a primary key and increase automatically.
3. Try connecting to MySQL server from your HTTP server by [mysql module](#) or [mysql-connector-nodejs module](#).
4. Provide a home page, including sign-up and sign-in forms, for people to take action.
 - a. When people submit sign-up form, you should add a new row in the user table if there is no the same email in the existing rows.
 - b. When people submit sign-in form, you should check if the pair of email and password matches any existing row in the user table.
5. If people sign-up or sign-in successfully, redirect them to a member page including any simple welcome message. **Done.**
6. If people failed to sign-up or sign-in, just keep them in home page with a proper wrong message. **Done.**

Reminder: In this assignment, you should hand in an additional data file in SQL format by [mysqldump tool](#).