# Intro to Coding with AI – Exercises

Exercises and homework assignments for “Introduction to Programming with AI” course at SoftUni.

## Run JS Code in the Browser

Web browsers have built-in **JavaScript runtime**, available in the **developer console**. The developer console, known as “**Chrome DevTools**” or “**Firefox DevTools**” is a powerful toolset for Web developers, built into modern Web browsers. **DevTools** allow inspecting the web app loaded in the browser, trace its communication with the server, inspect network resources loaded, access JavaScript runtime, cookies, storage, etc.

Your task is to open the **Chrome** Web browser and **run a few JavaScript commands** in its Developer Console:

|  |
| --- |
| let width = 3;  let height = 8;  let area = width \* height;  console.log(area); |

To open the developer console (**Chrome DevTools**), press **[F12]** (in Windows) or **[Cmd ⌘ + Alt ⌥ + I]** (in macOS).

A screenshot of a computer

AI-generated content may be incorrect.

You can **experiment** with running more commands in the browser like these:

|  |
| --- |
| let radius = prompt("Enter a circle radius");  alert("Circle area: " + Math.PI \* radius \* radius); |

## Run JS Code in the Terminal

The “**System Terminal**” in the Operating System (OS) is a **command-line interface** (CLI), used to invoke system commands, e. g. list files, start an app, view running processes, change file permissions, etc. A command-line interface means that **users send commands** on the console, and the system **responds to each command** interactively. This style of communication with the OS is also known as an “**interactive shell**”.

Your task now is to open the **system terminal**, start **Node.js** and execute a few JavaScript commands in the Node.js interpreter:

|  |
| --- |
| s |

First, **open the system terminal** in your OS (in Windows / Linux / macOS / other). Start the “**Terminal**” app:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Next, type a **sample terminal command**, e.g.:

|  |
| --- |
| date |

A screenshot of a computer program

AI-generated content may be incorrect.

Type a sample **invalid command** to see how it will produce **an error message** like this:

|  |
| --- |
| give me a coffee |

A screenshot of a computer

AI-generated content may be incorrect.

Now, **start Node.js**. Type the following command:

|  |
| --- |
| node |

If Node.js is properly installed, the **Node.js JavaScript command interpreter** will start and will show the version of Node.js, which is currently running in the terminal:

A screenshot of a black screen

AI-generated content may be incorrect.

If **Node.js** is not properly installed, you will see an error message: ***'node' is not recognized as an internal or external command***. In this case, you should install the Node.js runtime: <https://nodejs.org>.

Now **type a few JavaScript commands** in the interactive Node.js runtime. For example:

|  |
| --- |
| let width = 3;  let height = 8;  let area = width \* height;  console.log(area); |

A screenshot of a computer program

AI-generated content may be incorrect.

**Try a few more JS commands** on your own. Experiment, explore, and have fun.

## Run JS Code in VS Code

**VS Code** is modern development environment (IDE) for coding in multiple languages, with powerful AI-assisted coding tools, based on **GitHub Copilot**. Your task is to **write and run a few lines of JS code** in VS Code.

First, **start VS Code** and open a **new window**: press **[Shift + Ctrl + N]** keys:

A screenshot of a computer

AI-generated content may be incorrect.

Create a **new folder** named "**Intro-Coding**" and open the folder:

A screen shot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Now you have an **empty VS Code project** that will hold your source code:

A screenshot of a computer

AI-generated content may be incorrect.

Create a **new JavaScript file** inside your project: hello.js.

A screenshot of a computer

AI-generated content may be incorrect.

Type a **sample JS command**:

|  |
| --- |
| console.log("Hello JS"); |

This is how your VS Code environment will look like:

A screenshot of a computer

AI-generated content may be incorrect.

Now, it’s time to **run the code** and see the execution results. In VS Code you can **run the code** by creating a “Run and Debug” configuration and then press [Ctrl+F5]. This approach is good for more experienced developers. For newbies, we recommend **installing an extension** called "**Code Runner**", which runs the file open in the current editor and supports many languages (like JavaScript, TypeScript, Python and C#).

**Install the "Code Runner" extension** to VS Code:

A screenshot of a computer program

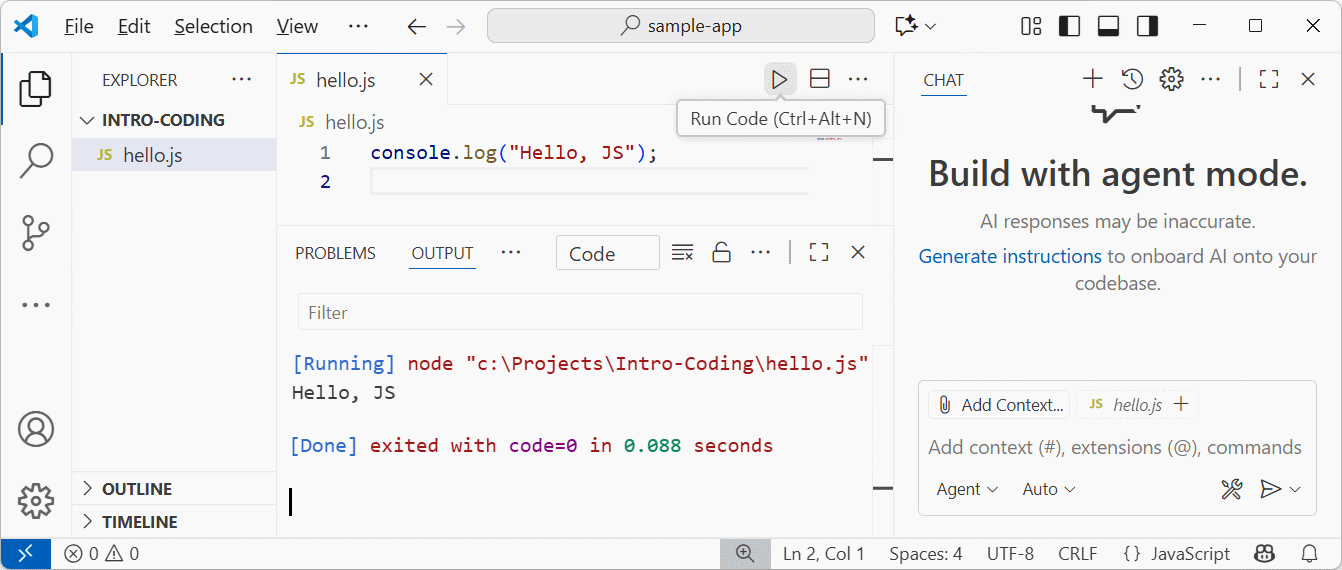
AI-generated content may be incorrect.

This extension will insert a **[Run] button** at the top of the file editor in VS Code:

A screenshot of a computer

AI-generated content may be incorrect.

Now running JS code file is quite simple: just **click the [Run]** button.



The **output** from the JavaScript file execution is visualized in the **[Output] window** in VS Code.

Now, intentionally introduce an **error** in the code: **put a wrong command** and run the code to see what happens. For example, remove the quotes when printing a text message:

A screenshot of a computer

AI-generated content may be incorrect.

This is how JavaScript works: when a **wrong command** is attempted to be executed, the JavaScript runtime shows an **error message**. Errors during the code execution are either handled by the code (with a special language syntax), or error messages are printed on the JS console.

Now, use the built-in **[Explain] function** in the code editor to **explain the problem** and how to fix it:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

## Sum 3 Numbers

**Functions** in JS are **named pieces of code**, which can be invoked multiple times. Functions can take input **parameters** and return a **result** as an output.

Your task now is to **write a function to sum given 3 numbers**. This is how the function should work:

|  |
| --- |
| let sum = sumNumbers(5, 2, 3);  console.log("Sum =", sum);  // Sum = 10  let sum = sumNumbers(10, -5, 0.2);  console.log("Sum =", sum);  // Sum = 5.2 |

First, create a **new file** to hold your code. Use a **meaningful name** (self-explainable). Example: sum-numbers.js.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Try to **write the function by hand**:

A screenshot of a computer

AI-generated content may be incorrect.

Or ask **GitHub Copilot** to write it by typing a **simple prompt**:

A screenshot of a computer

AI-generated content may be incorrect.

To **accept** the code suggestion, press **[Tab]** key.

## Submit Code to SoftUni Judge

Now, let’s **submit your code** for evaluation to **SoftUni Judge**.

Open this contest in SoftUni Judge: <https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264>. First **login in SoftUni Judge** with your personal SoftUni username. Then **submit your code** for evaluation:

A screenshot of a computer program

AI-generated content may be incorrect.

Below you will see the list of **submissions** for the currently selected task.

A screenshot of a computer program

AI-generated content may be incorrect.

To view the most recent results, press the **[⟳ Refresh] button**.

Now intentionally **submit wrong code** to see how SoftUni Judge will process it:

|  |
| --- |
| function sumNumbers(a, b, c) {  // This code is intentionally wrong:  return a + b;  } |

A screenshot of a computer program

AI-generated content may be incorrect.

The **execution results** will look like this:

A screenshot of a computer

AI-generated content may be incorrect.

## Visualize “Sum 3 Numbers” with HTML

Now, let’s play a bit. Writing functions without using them in real-world projects is boring. Can we **visualize** the “Sum 3 Numbers” problem?

We can create a **simple HTML page** that invokes our number summator function and displays the results in a well-looking visual way. We can use this **prompt** in GitHub Copilot coding agent:

|  |
| --- |
| Create a simple HTML with CSS to enter data and call the function. |

A screenshot of a computer

AI-generated content may be incorrect.

The above prompt will generate a new file “index.html” in your VS Code project.

A screenshot of a computer

AI-generated content may be incorrect.

This file will hold an **HTML page**, with an input form to **enter 3 numbers** and a **button** to calculate their sum and a **field for the output**. To **open** the generated **HTML**, reveal the file in the file explorer:

A screenshot of a computer

AI-generated content may be incorrect.

Now **double-click** on the HTML file:

A close-up of a paper

AI-generated content may be incorrect.

This is an example of the result:

A screenshot of a computer

AI-generated content may be incorrect.

Note that **the HTML page may look different**, because it is AI-generated and each **prompt run** may generate a **different HTML code** as an output.

## Configure Code Runner to Open HTML Files

To simplify **opening HTML files** in VS Code, it is recommended to **configure your “Code Runner”** extension.

By default, VS Code and Code Runner extension **cannot run HTML pages**:



Attempts to run the HTML file will produce an **error message**:

A screenshot of a computer

AI-generated content may be incorrect.

How to **fix** this? You can **configure** **your “Code Runner”** extension to **handle correctly HTML files**.

Open Code Runner **extension settings**:

A screenshot of a computer error

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Open the **Executor Map**:

A screenshot of a computer

AI-generated content may be incorrect.

Add a configuration line that assigns "**html**" files to open in the **Chrome** Web browser:

A screenshot of a computer program

AI-generated content may be incorrect.

Now Code Runner in VS Code will **open Web pages** in the Web browser, when **[Run] button** is clicked:

A screenshot of a computer program

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Sum Hours and Minutes (with Copilot)

Write a function sumTimes(h1, m1, h2, m2) to **sum times** in 24-hour format “hh:mm”. Visualize the work of your function with **AI-generated HTML page**.

### Input and Output

The **input** comes as 4 numbers: **h1**, **m1**, **h2**, **m2**, which hold a **start time** (hours and minutes) and a **time interval to add** (hours and minutes). The **output** consists of **two text lines**, like it is shown in the examples below.

Sample **input** and **output** from your function sumTimes(h1, m1, h2, m2):

|  |  |
| --- | --- |
| **Input** | **Output** |
| sumTimes(10, 30, 0, 5) | The time is 10:30 now.  After 00:05 the time will be 10:35. |
| sumTimes(10, 30, 0, 45) | The time is 10:30 now.  After 00:45 the time will be 11:15. |
| sumTimes(23, 58, 1, 15) | The time is 23:58 now.  After 01:15 the time will be 01:13. |

### Hints

First, create a new file "sum-time.js".

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Next, use **GitHub Copilot** in **[Edit] mode** to write the function:

A screenshot of a chat

AI-generated content may be incorrect.

This is an example of how the **generated function code** may look like:

A screen shot of a computer code

AI-generated content may be incorrect.

**Run** and **test** the code to ensure that it works correctly:

A screenshot of a computer

AI-generated content may be incorrect.

### Explain the Code

In software development it is always good to **know how your code works**. Yes, the AI agent can write the code, and in most cases, it is correct, but you as a developer should **check the code** and confirm its correctness. Your task now is to read the code, understand its logic, and explain how the code works, step by step. You can help yourself with **GitHub Copilot in [Ask] mode**:

A screenshot of a chat

AI-generated content may be incorrect. A screenshot of a computer screen

AI-generated content may be incorrect.

### Judge Submission

Now your code solves correctly the problem, and you can explain how it works. Now you can submit your solution for automated grading in SoftUni Judge.

**Submit** your function to the **judge** here: <https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264>.

A screenshot of a computer program

AI-generated content may be incorrect. A screenshot of a program

AI-generated content may be incorrect.

### Visualize with HTML

Now, let’s visualize the function sumTimes(h1, m1, h2, m2) as HTML Web page. First, we need to **redesign the function** to return its output instead of printing it on the console. Why? Because we cannot easily visualize the console output messages, but we can **visualize a function returned output**.

For simple changes in a single file, it is recommended to use **Copilot in [Edit] mode**.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Now, instead of just invoking the function, we **invoke the function and print its returned result**:

|  |
| --- |
| console.log(sumTimes(23, 58, 1, 15)); |

Now, let’s **visualize the function in HTML page** using a prompt like this:

|  |
| --- |
| Visualize the function sumTimes(h1, m1, h2, m2) with a HTML page `sum-time.html`: enter the input parameters, invoke the function and visualize the function output. |

Use **Copilot in Agent mode** with a **premium LLM model** to create the new HTML page to invoke the existing JS code:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

## Sum 3 Hours and Minutes (with ChatGPT)

Now solve a similar problem, without GitHub Copilot. Use **ChatGPT in “Learn” mode** instead.

Write a function sum3Times(h1, m1, h2, m2, h3, m3) to **sum 3 times** in 24-hour format “hh:mm”.

### Input and Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| sum3Times(10, 30, 0, 5, 0, 2) | 10:30 + 00:05 + 00:02 = 10:37 |
| sum3Times(11, 50, 0, 45, 1, 55) | 11:50 + 00:45 + 01:55 = 14:30 |
| sum3Times(21, 58, 9, 15, 16, 51) | 21:58 + 09:15 + 16:51 = 00:04 |

### Hints

First, create a **new JS file** to hold your code: sum-3-times.js

Next, create a prompt in **ChatGPT in “Learn mode”**:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

**Talk with ChatGPT**, answer the questions, think, ask questions and follow the instructions from ChatGPT. Finally, write the correct code for the function sum3Times(h1, m1, h2, m2, h3, m3) to solve the problem. Example:

A screenshot of a computer program

AI-generated content may be incorrect.

**Run and test your code** to ensure it works as expected.

### Judge Submission

**Submit your solution** to the **judge**: <https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264>.

A screenshot of a computer

AI-generated content may be incorrect.

If your code is correct, you will get 100 / 100 result:

A screenshot of a computer

AI-generated content may be incorrect.

### Visualize with HTML

Now, let’s visualize the function sum3Times(h1, m1, h2, m2, h3, m3) as HTML Web page, where the user enters 3 times (hour + minutes), the function is invoked and the result form it is visualized. This is a sample Copilot prompt you can use:

|  |
| --- |
| Visualize the function sum3Times(h1, m1, h2, m2, h3, m3) with a HTML page `sum-3-times.html`: enter the input parameters, invoke the function and visualize the function output. |

A white background with black text

AI-generated content may be incorrect.

This is a **sample** HTML page, generated as a result:

A screenshot of a computer

AI-generated content may be incorrect.

## Inches to Centimeters

Write a function to **convert inches to centimeters**. The output should be rounded to **2 decimal places**.

### Input and Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| inchesToCentimeters(1) | 2.54 |
| inchesToCentimeters(3.5) | 8.89 |

### Hints

Create a **new JS file** with a reasonable name: inches2centimeters.js.

Use **ChatGPT**, **Claude** or **GitHub Copilot** to write the function, using a prompt.

|  |
| --- |
| function inchesToCentimeters(inches) {  // TODO: convert inches to centimeters, with rounding to 2 decimal digits, and return the result  } |

Test the function with a sample input:

|  |
| --- |
| console.log(inchesToCentimeters(3.5));  // The expected result is: 8.89 |

### Judge Submission

**Submit your solution** to the **judge**: <https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264>.

### Visualize with HTML

Visualize the function with HTML, using a **GitHub Copilot prompt**:

A screenshot of a web browser

AI-generated content may be incorrect.

## Info Message

Write a function that takes as input **first name**, **last name**, **age** and **town** and prints a message like this:

|  |
| --- |
| You are <firstName> <lastName>, a <age>-years old person from <town>. |

### Input and Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| infoMessage("Steve", "Nak", 35, "Sofia") | You are Steve Nak, a 35-years old person from Sofia. |
| infoMessage("Maria", "Green", 24, "London") | You are Maria Green, a 24-years old person from London. |

### Hints

Create a **new JS file** with a reasonable name.

Use **ChatGPT**, **Claude** or **GitHub Copilot** to write the function, using a prompt.

### Judge Submission

**Submit your solution** to the **judge**: <https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264>.

### Visualize with HTML

Visualize the function with HTML, using a **GitHub Copilot prompt**.

## Vegetables Price

You want to **order vegetables**. You are given the quantity of **tomatoes** and the quantity of **cucumbers** to be ordered. The **prices** are fixed:

* Tomatoes: 2.4 EUR / kg (without VAT)
* Cucumbers: 1.9 EUR / kg (without VAT)

Write a JS function to **calculate the vegetables price** without VAT and the total price with 20% VAT. Return the results as text, in the format given in the examples below. Round all numbers to **2 decimal places**.

### Input and Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| vegetablesPrice(2, 3) | Tomatoes: 4.80 EUR  Cucumbers: 5.70 EUR  Total: 10.50 EUR  20% VAT: 2.10 EUR  Total price (with VAT): 12.60 EUR |

### Hints

Create a **new JS file** with a reasonable name.

Use **ChatGPT**, **Claude** or **GitHub Copilot** to write the function, using a prompt.

### Judge Submission

**Submit your solution** to the **judge**: <https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264>.

### Visualize with HTML

Visualize the function with HTML, using a **GitHub Copilot prompt**.