**First Steps In Coding**

Problems for in-class and homework exercises for the course ["Programming Basics" @ SoftUni](https://softuni.org/).

Test your solutions in the online **judge system**: <https://judge.softuni.org/Contests/Compete/Index/3487>

# Empty Visual Studio Code project

Create empty project in **Visual Studio Code**. We will combine the solutions of all tasks in the form of separate files in this project. This option is extremely handy when we want to work on several projects and quickly switch between them, or we want to logically merge several interrelated projects. This helps us to keep the task solutions separate and to use them for other tasks or revisions.

1. Start Visual Studio Code
2. Create a new folder that will hold the individual solutions. A dialog box will open in which you will need to select its directory. It is recommended that you name the folder according to the topic of the assignment, for example, "**First-Steps-in-Calculations**".



Next, select the folder as your workspace, to add the JavaScript solution files of your tasks to it.



The panel on the left will look like this:



## Hello SoftUni

Write a **function** that prints the text "Hello SoftUni".

### Hints and Guidelines

1. Create a **new JavaScript file** in the existing folder and name it appropriately. It is recommended to name each script file as the name of the task whose solution it contains.





1. The content of the new file will open in the window on the **right**.



1. Go to the **hello.js** file and create the hello()function. You can help yourself with the image below:



1. Run the program with **Ctrl + F5**. To get the result we need to "call" the function:



1. We can see the result in the console below:



1. Test the solution to this problem in the online [Judge](https://judge.softuni.org/Contests/Compete/Index/2399#0) system:



## Nums 1...10

Write a function that **prints the numbers from 1 to 10** on separate lines on the console.

### Hints and Guidelines

1. Create a **new JavaScript file** in the existing folder and name it "Nums-1-To-10". Use the body of the function"nums1To10()":

function nums1To10 () {

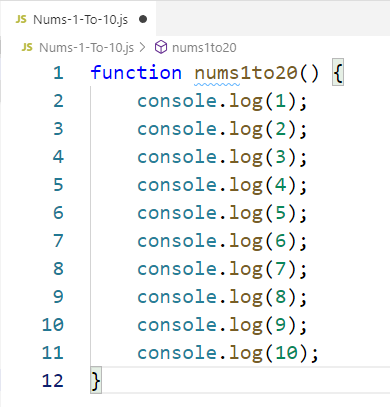
// Your code goes here

}

// Call the function to execute the code inside

nums1To10();

1. Write 10 commands console.log(), one after another, to print the numbers from 1 to 10.



1. Run the program with **Ctrl + F5**. To get the result you need to "call" the function.

## Square Area

Write a function squareArea(side), that **takes an integer number** and **calculates the square area given its side** (the side equals the given number).

**Sample Input and Output**

|  |  |
| --- | --- |
| **Input** | **Output** |
| squareArea(5) | 25 |

### Hints and Guidelines

1. Take an input (a number in string form) and store it in the variable **a**, converting it to a numeric type.
2. Initialize a variable that holds the calculated area value, obtained from the formula (**a \* a**) and print the result, remembering that to get a result locally, you must call the function and give it input.

Картина, която съдържа текст

Описанието е генерирано автоматично

## Inches to Centimeters

Write a function inchesToCm(inch) that **reads a floating-point number from the console** and converts it **from inches to centimeters**.

To do this, **multiply the inches by 2.54** (1 inch = 2.54 centimeters).

**Sample Input and Output**

|  |  |
| --- | --- |
| **Input** | **Output** |
| inchesToCm(5) | 12.7 |

|  |  |
| --- | --- |
| **Input** | **Output** |
| inchesToCm(7) | 17.78 |

**Attention:** depending on the regional settings of the operating system, it is possible to use a **decimal comma** instead of a **decimal point** (US settings). If the program expects a decimal point and enters a number with a decimal comma or vice versa (a decimal point is entered when a decimal comma is expected), it will not be able to be executed.

It is recommended **that you change the settings on your computer** to use a **decimal point**:





## Greeting by Name

Write a function greetings(name) that **receives a person’s name** and prints "Hello, <name>!", where <name> is the name is received as input.

### Hints and Guidelines

1. First, create a **new JavaScript file** in the existing folder and name it appropriately. It is recommended to name each script file as the name of the task whose solution it contains.



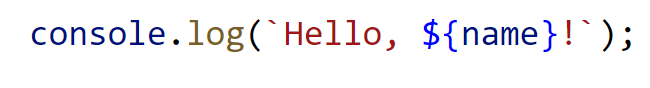


1. The content of the new file will open in the window on the right.



Create the variable **name** and save in it the name that is passed from (**input**).

Display the output on the console using the following template:



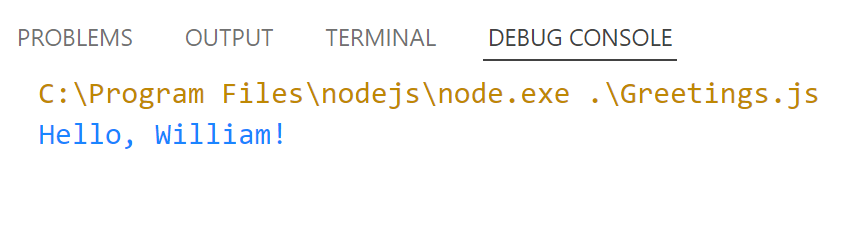
How does the example work? The **console.log ()** method allows us to write in parentheses a series of characters that hold the value of a variable - **${name}** and print it on the console.

Note that for this to work, the text must be enclosed by the **`** symbol, which recognizes plain text and a variable. To recognize a variable as such, it must begin with the symbol **$** and be enclosed in curly braces: **{ }**.

1. Run the program with **Ctrl + F5** and test with different input examples.
2. To get a result we need to "call" the function and pass it input data:

Картина, която съдържа текст

Описанието е генерирано автоматично



1. Test your solution in judge, without including the calling of the function.

## Concatenate Data

Write a function concatenateData(firstName, lastName, age, town) and print a message of the following type:

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

### Hints and Guidelines

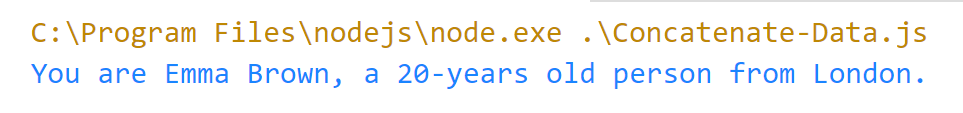
1. Add another **JavaScript file** named "**Concatenate-Data**" to the current Visual Studio Code solution.
2. Take the input from the console:

Картина, която съдържа текст

Описанието е генерирано автоматично

1. Display the formatted output on the console.
2. Run the program with **Ctrl + F5** and test with different input examples. To display the result locally to the console, you need to call the function and pass the input data in the order in which you expect to receive it:





## Projects Creation

Write a function projectCreation(architect, projects) that **calculates how many hours** it will take an architect to **complete** several construction projects. One project takes **three hours** to complete.

**Input Data**

The function receives **2 arguments**:

1. **Name of the architect - text**
2. **Number of projects to complete – an integer in the range [0 … 100]**

**Output Data**

On the console is printed:

* **"The architect {name of architect} will need {numbers of hour needed} hours to complete {number of projects} project/s."**

**Sample Input and Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| projectCreation("George", 4) | The architect George will need 12 hours to complete 4 project/s. | projectCreation("Sarah", 9) | The architect Sarah will need 27 hours to complete 9 project/s. |

## Pet Shop

Write a function calculate(dogPackages, catPackages) that **calculates the costs needed** to buy food for dogs and cats. The food is bought from a pet shop, a packet of **dog food costs 2.50 USD** and a packet of **cat food costs 4 USD**.

**Input Data**

The function receives **2 arguments**:

1. **Number of dog food packages – an integer in the range [0… 100]**
2. **Number of cat food packages – an integer in the range [0… 100]**

**Output Data**

On the console is printed:

**"{final amount} USD."**

**Sample Input and Output**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | **Output** | |  | **Input** | **Output** |
| calculate(5, 4) | | 28.5 USD. | calculate(13, 9) | 68.5 USD. |

## Yard Greening

Betty has **several houses** on the Mediterranean Sea coast and **wants to green the yards of some of them**, thus creating a **cozy atmosphere and comfort for its guests**. She has hired a company for this purpose.

Write a function calculate(meters) that **calculates the amount needed** that Betty will have to pay to the project contractor. The price per **square meter is 7.61 USD including VAT**. Because her yard is **quite large**, the contractor company offers an **18% discount on the final price**.

**Input Data**

The function receives only **1 argument**:

1. **Square meters to be landscaped – a floating-point number in the range [0.00 … 10000.00]**

**Output Data**

**Two lines** are printed on the console:

* **"The final price is: {final service price} USD."**
* **"The discount is: {discount} USD."**

**Sample Input and Output**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| **calculate(550)** | The final price is: 3432.11 USD.  The discount is: 753.39 USD. | We calculate the cost of landscaping the entire yard:  **550** \* **7.61** = **4185.50** USD.  We deduct the discount (18% = 0.18) from the total amount:  **0.18** \* **4185.5** = **753.39** USD.  We calculate the final price of the service:  **4185.50** – **753.39** 🡪 3432.11 USD. |
| **Input** | **Output** | **Comments** |
| **calculate(150)** | The final price is: 936.03 USD.  The discount is: 205.47 USD. | We calculate the cost of landscaping the entire yard:  **150** \* **7.61** = **1141.50** USD.  We deduct the discount (18% = 0.18) from the total amount**:**  **0.18** \* **1141.50** = **205.47** USD.  We calculate the final price of the service:  **1141.50** – **205.47** 🡪 936.03 USD. |