

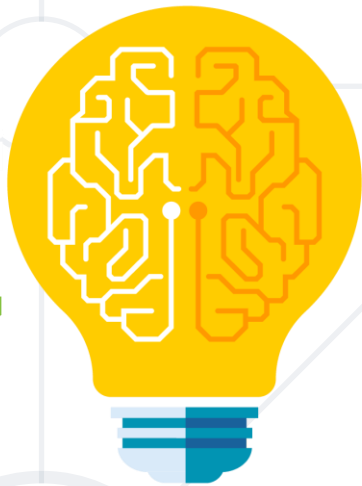
Първи стъпки в програмирането

Работа с конзола, аритметични операции с числа



СофтУни

Преподавателски екип



SoftUni



Софтуерен университет

<https://softuni.bg>

1. Какво е програмиране?
2. Първа програма с **JavaScript** и **Visual Studio Code**
3. Променливи и типове данни
4. Четене на потребителски вход
5. Прости операции
 - работа с текст и числа
 - аритметични операции
6. Печатане на конзолата





Какво е програмиране?

Какво е програмиране?



- Компютърна наука
- Използва **команди**, за да **комуникираме** с компютъра
- Командите се подреждат и изпълняват **една след друга**
- Поредицата от команди образува **компютърна програма**

Какво е език за програмиране?

- Компютърните програми се пишат на език за програмиране
- Пример: JavaScript, C#, Java, Python, PHP, C, C++
- Използва се **среда за програмиране (среда за разработка)**
- Пример: Visual Studio Code, IntelliJ IDEA, Visual Studio, PyCharm, Code Blocks



Какво е компютърна програма?

- Програмата е **последователност от команди**
- Може да съдържа пресмятания, проверки, повторения
- Програмите се пишат в текстов формат
 - Текстът на програмата се нарича **сорс код**
- Сорс кодът се **компилира** до изпълним файл



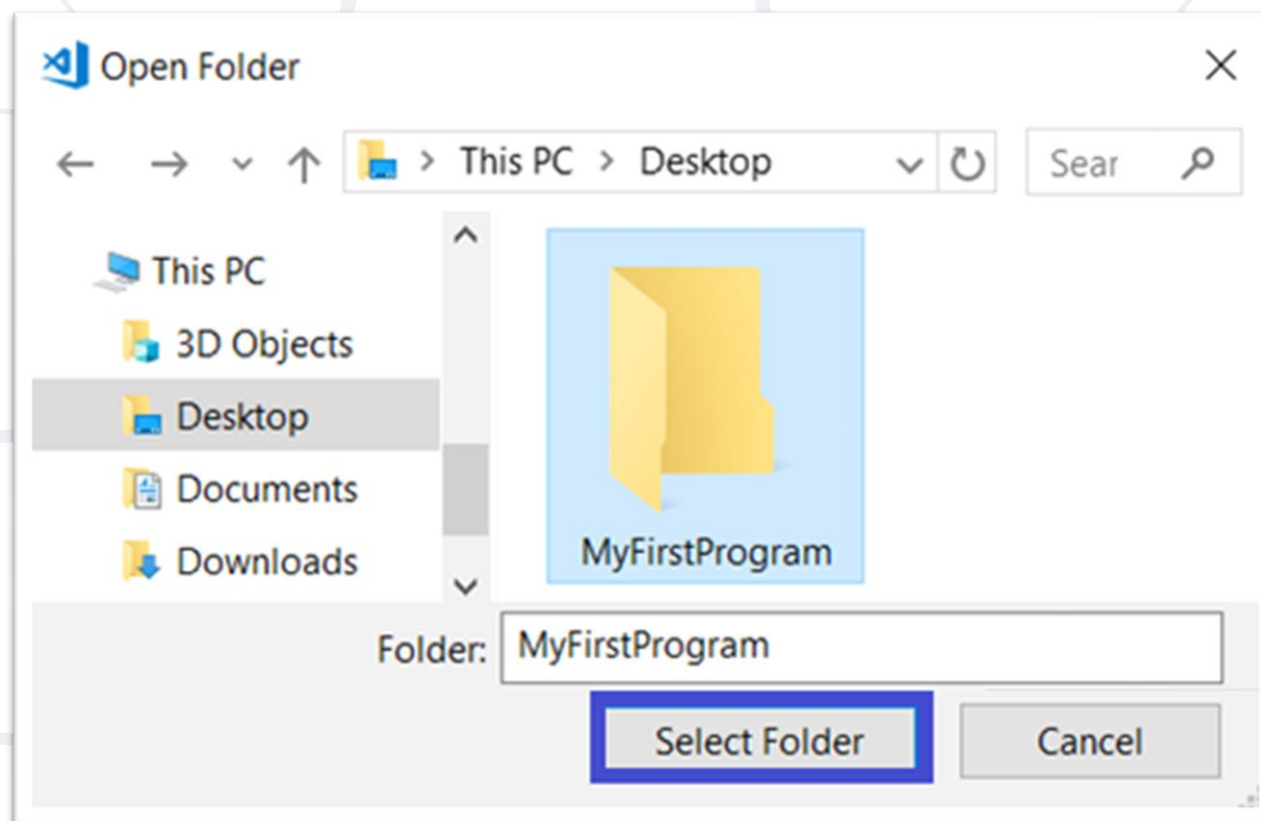


**Да направим първите си конзолни
програми**

- Средата за разработка е нужна, за да програмирате
 - **Integrated Development Environment (IDE)**
 - **Visual Studio Code** е среда за разработка на езика JavaScript
- Инсталирайте си **Visual Studio Code**
 - **Инструкции за инсталация**
- Приложението е **мултиплатформено** (Linux, Mac OS, Windows)

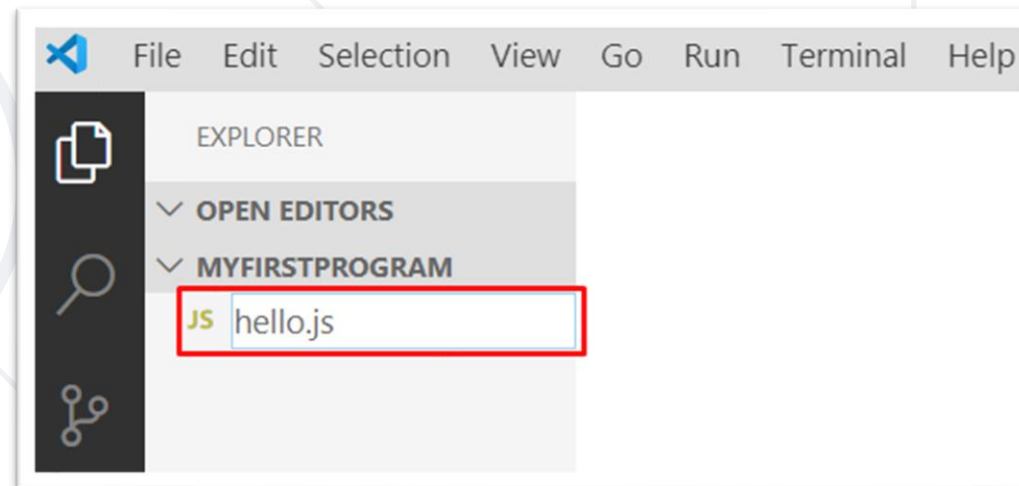
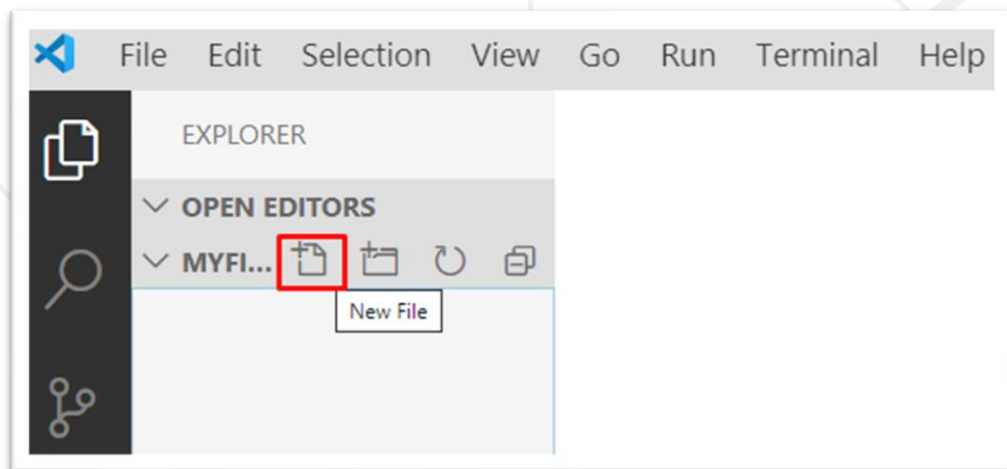
Създаване на конзолна програма

- Създайте и изберете временната папка, в която ще създадем нашата първа програма

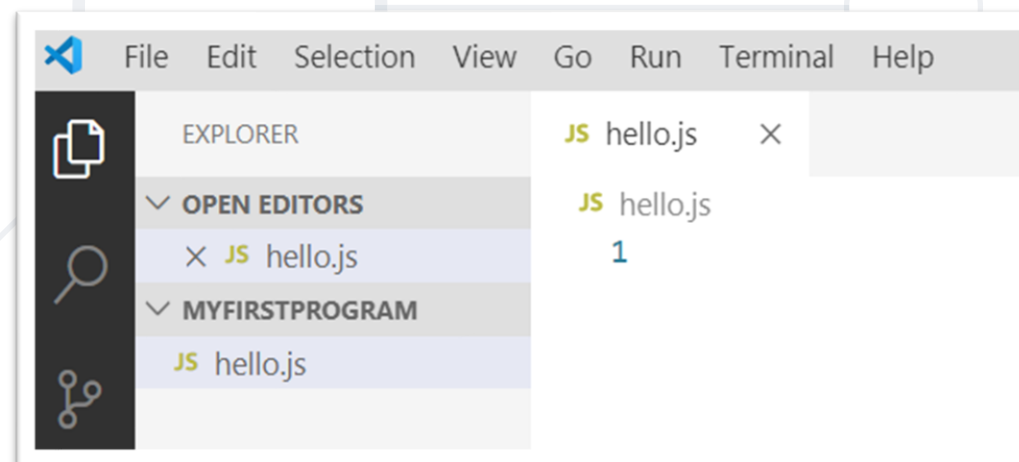


Създаване на конзолна програма

- Добавете файла, в който ще пишем кода за нашата програма



- Сорс кодът на програмата ще напишем в празния файл **"hello.js"**, който вече създадохме



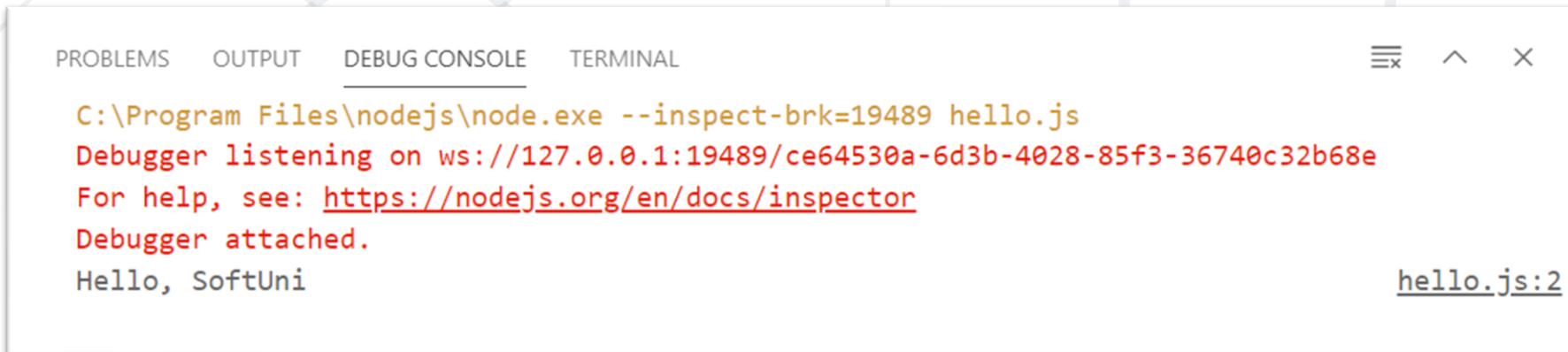
- Напишете следния код:

```
function hello() {  
    console.log("Hello SoftUni");  
}  
hello();
```



Стартиране на програмата

- За стартиране на програмата натиснете:
 - **Ctrl + F5**
 - **Debug → Start Without Debugging**
- Ако няма грешки, програмата ще се изпълни
- Резултатът ще се изпише на конзолата:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Program Files\nodejs\node.exe --inspect-brk=19489 hello.js
Debugger listening on ws://127.0.0.1:19489/ce64530a-6d3b-4028-85f3-36740c32b68e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Hello, SoftUni                                     hello.js:2
```

- Бъркане на **малки** и **главни** букви:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Debugging with inspector protocol because Node.js v8.9.4 was detected.
node .\vscode\Demo.js
c:\Users\User\Documents\VS Code Projects\Loops\.vscode\Demo.js:2
  Console.log("Hello")
  ^
ReferenceError: Console is not defined
    at hello (c:\Users\User\Documents\VS Code Projects\Loops\.vscode\Demo.js:2:5)
    at Object.<anonymous> (c:\Users\User\Documents\VS Code Projects\Loops\.vscode\Demo.js:5:1)
```

- Липсваща кавичка " или липсваща скоба (или)

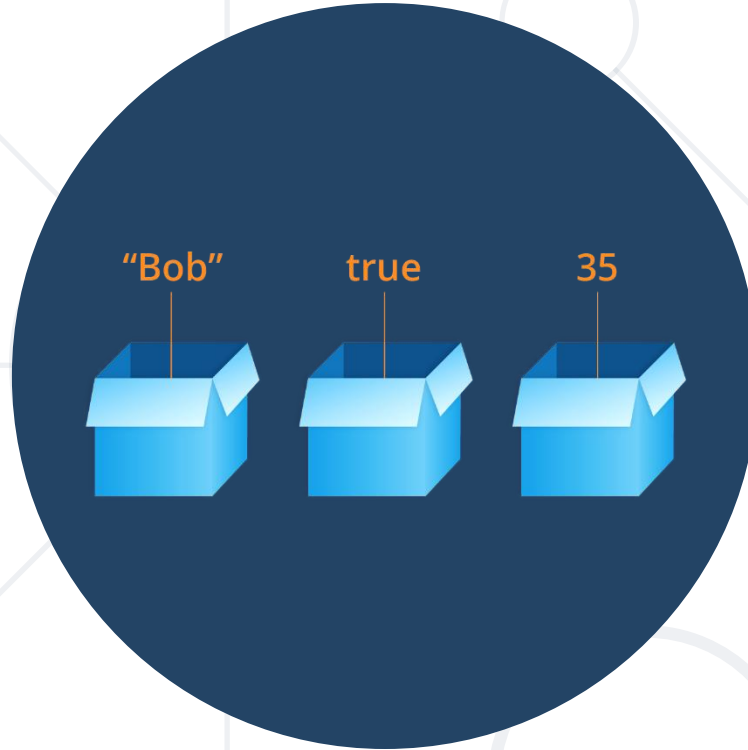
```
console.log("Hello JavaScript);
```

```
console.log("Hello JavaScript";
```

- Напишете програма, която принтира числата от **1** до **10**, всяко на нов ред

- Решение:

```
function numsFrom1to10() {  
    console.log(1);  
    console.log(2);  
    console.log(3);  
    ...  
    console.log(10);  
}  
  
numsFrom1to10();
```



Променливи и типове данни

- Компютрите са машини, които обработват данни
 - Данните се записват в компютърната памет в променливи
 - Променливите имат **тип, име и стойност**
- Дефиниране на променлива и присвояване на стойност:

Инициализация

Име на променлива

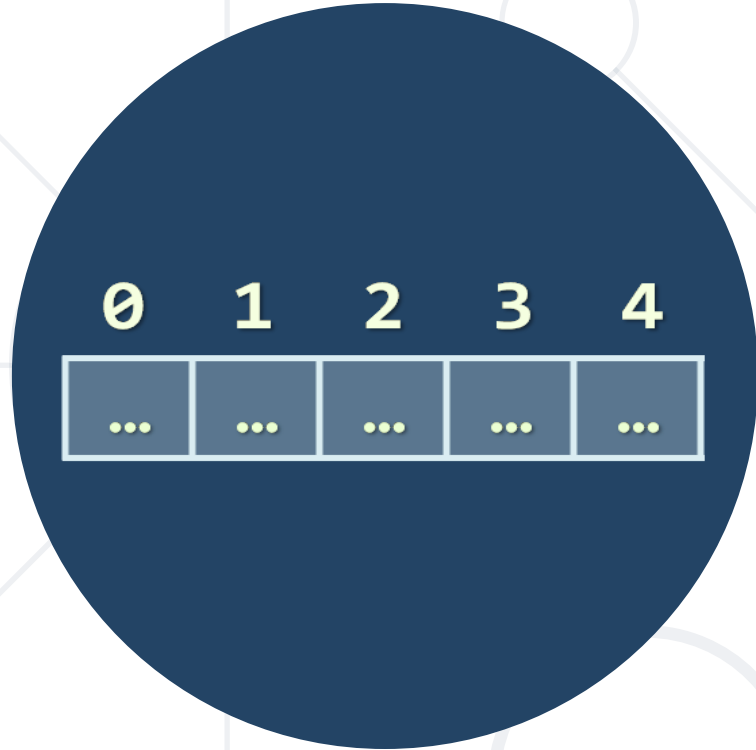
```
let count = 5;
```

Стойност (от тип число)

Типове данни

- Променливите съхраняват стойност от даден тип
 - Число, буква, текст (низ), дата, списък, ...
- Типове данни – примери:
 - **Number** – число: **7**, **3.14**, **-1.5**, ...
 - **String** – текст: **"Hello"**, **"Здрасти"**, **"p@r0La"**, ...
 - **Undefined** – променлива, на която не е присвоена стойност





Масиви – четене от масив

Какво е масив?

В програмирането **масив** е **поредица от елементи**



- Можем да запазваме **много стойности** в една променлива
- Елементите са номерирани от **0** до **length-1**
- Масивите имат **дължина на променливата** (**Array.length**)

- Както от подаден текст можем да прочетем само една буква, така можем да прочетем само един елемент от даден масив.

```
function solve(input){  
  console.log(input[0]); //a  
  console.log(input[1]); //b  
  console.log(input[2]); //c  
}  
solve([ "a", "b", "c" ])
```



Получаване на потребителски вход

Работа с функции

Четене на текст

- Всичко, което **получаваме** като вход, идва под формата на **аргумент**
- Всичко, което **печатаме** на конзолата, се **преобразува в текст**
- Получаване на текст:

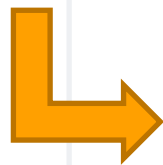
```
function readText(input) {  
  let str = input[0];  
}
```



Четене на текст (2)

- Пример:

```
function readName(input) {  
  let name = input[0];  
  console.log(name);  
}  
  
readName([ 'SoftUni' ]);
```



DEBUG CONSOLE

```
Debugging with inspector protocol  
node simpleCalculations.js  
SoftUni
```



Четене на числа

- Получаване на число:

```
function readNumber(input)
  let num = Number(input[0]);
}
```

- Пример: пресмятане на лице на квадрат със страна **a**:

```
function squareArea(input) {
  let a = Number(input[0]);
  let area = a * a;
  console.log(area);
}
```

Получаване на
число на един ред





Прости операции

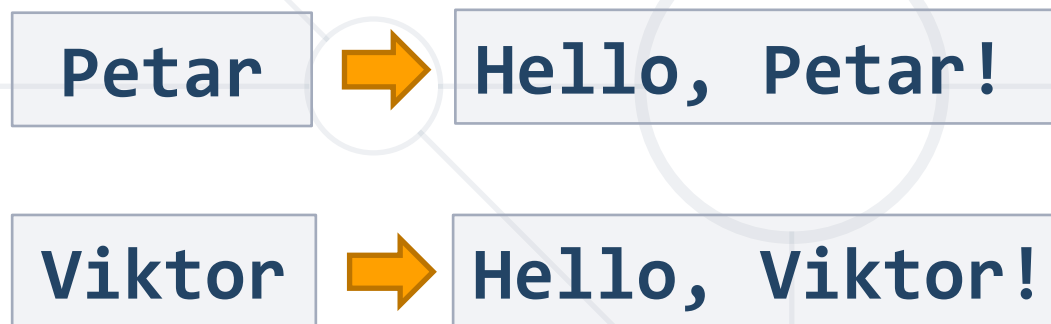
Работа с текст и числа

Поздрав по име – пример

- Да се напише функция, която:

- Получава като аргумент **име** на човек
- Отпечатва "Hello, <name>", където <name> е полученото преди това **име**

- Примерен вход и изход:



```
function greetingByName(input) {  
  let name = input[0];  
  let greeting = "Hello, " + name;  
  console.log(greeting);  
}  
greetingByName(['Svetlin Nakov']);
```

Операцията "+"
долепя текстовата
стойност и
променливата

DEBUG CONSOLE

...

Debugging with inspector protocol
node simpleCalculations.js
Hello, Svetlin Nakov

Съединяване на текст и число

- Съединяване на текст и число (**оператор +**):



```
let firstName = "Maria";  
let lastName = "Ivanova";  
let age = 19;  
let str = firstName + " " + lastName + " @ " + age;  
console.log(str);           // Maria Ivanova @ 19
```

```
let a = 1.5;  
let b = 2.5;  
let sum = "The sum is: " + a + b;  
console.log(sum);           // The sum is 1.52.5
```

Резултатът е
долепяне/конкатенация

Аритметични операции: + и -

- Събиране на числа (**оператор +**):

```
let a = 5;  
let b = 7;  
let sum = a + b;           // 12
```

- Изваждане на числа (**оператор -**):

```
function subtract(input)  
  let a = Number(input[0]);  
  let b = Number(input[1]);  
  let result = a - b;  
  console.log(result);
```



Аритметични операции: * и /

- Умножение на числа (**оператор ***):

```
let a = 5;  
let b = 7;  
let product = a * b;           // 35
```

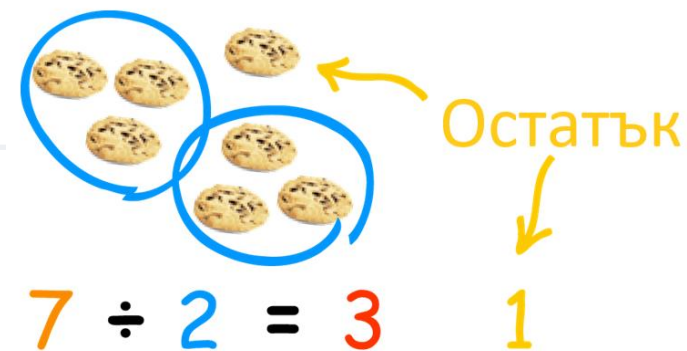
- Деление на числа (**оператор /**):

```
let a = 25;  
let i = a / 4;                 // 6.25 (дробно деление)  
let f = parseInt(a / 4.0);     // 6 (дробната част се отрязва)  
let infinity = a / 0;          // Infinity (безкрайност)  
let sqrt = Math.sqrt(-1);      // получава се NaN
```



- Модул/остатък от целочислено деление на числа (**оператор %**):

```
let a = 7  
let b = 2  
let product = a % b // 1
```



```
let odd = 3 % 2  
let even = 4 % 2  
let error = 3 % 0
```

```
// 1 – числото 3 е нечетно  
// 0 – числото 4 е четно  
// резултатът е NaN
```



Печатане на конзолата

- При печат на текст, числа и други данни, можем да ги съединим, използвайки интерполация:

``${arg1}, ${arg2}, ${arg3}``

- Пример:

```
function greet(input) {  
  let name = input[0];  
  console.log(`Hello, ${name}!`);  
}
```

Интерполация на стринг се извършва с back-tick(`)

Изразът `${name}` се замества с
със стойността, която стои
зад променливата `name`

- Да се напише програма, която:
 - Получава като аргумент име, фамилия, възраст на човек и град, въведени от потребителя
 - Отпечатва "You are <firstName> <lastName>, a <age>-years old person from <town>."
 - Примерен вход и изход:

Вход: `['Petar', 'Petrov', '24', 'Sofia']`

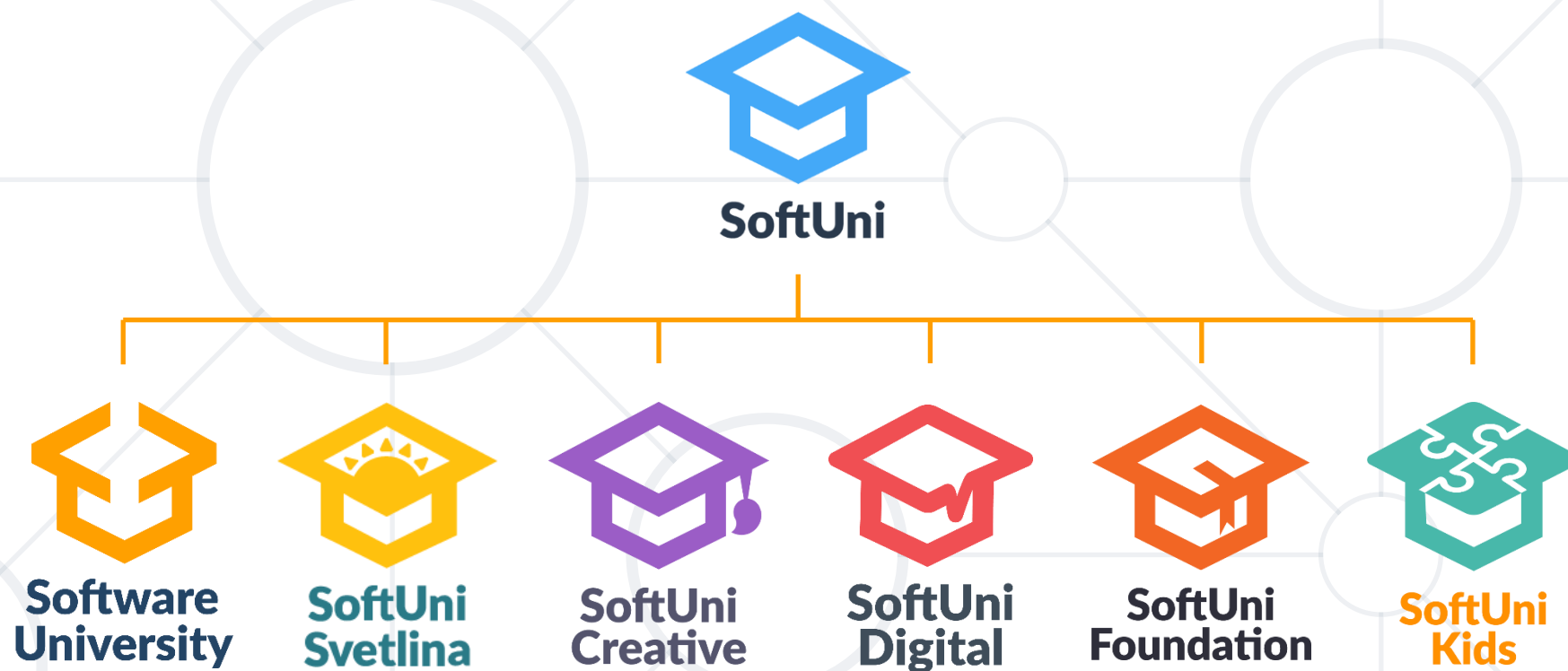
Изход: `You are Petar Petrov, a 24-years old person from Sofia.`

```
function personalInfo(input) {  
    let firstName = input[0];  
    let lastName = input[1];  
    let age = Number(input[2]);  
    let town = input[3];  
    console.log(`You are ${firstName} ${lastName}, a  
    ${age}-years old person from ${town}.`);  
}
```

- Компютърната програма е поредица команди
- На JavaScript командите се пишат във функции
- Печатаме с командата **console.log(...)**
- Въвеждане на текст и числа
- Аритметичните операции с числа:
+, -, *, /, (), %
- Отпечатване на текст по шаблон



Въпроси?



- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни – <https://softuni.org>
- © Софтуерен университет – <https://softuni.bg>



- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
 - softuni.bg
- Фондация "Софтуерен университет"
 - softuni.foundation
- Софтуерен университет @ Facebook
 - facebook.com/SoftwareUniversity
- Дискуссионни форуми на СофтУни
 - forum.softuni.bg



Software University

