

Лаб: Условни конструкции

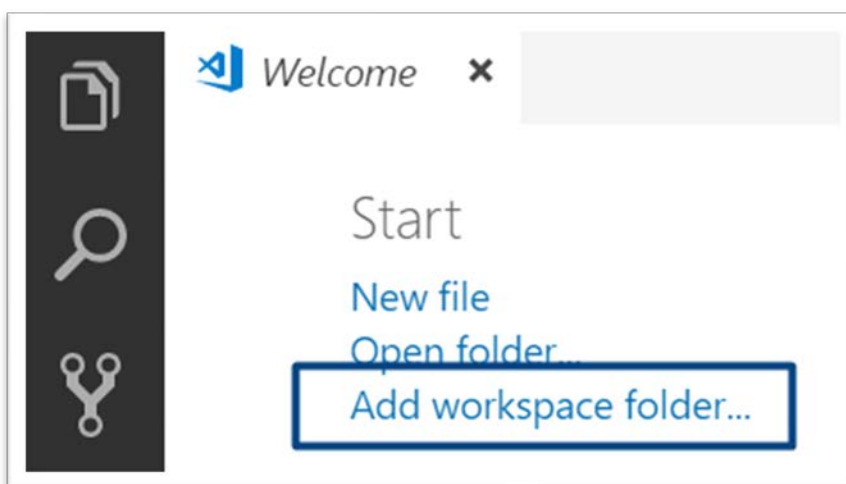
Задачи за упражнение в клас и за домашно към курса "[Основи на програмирането](#)" @ СофтУни.

Тествайте решенията си в Judge системата: <https://judge.softuni.bg/Contests/Index/2401#0>

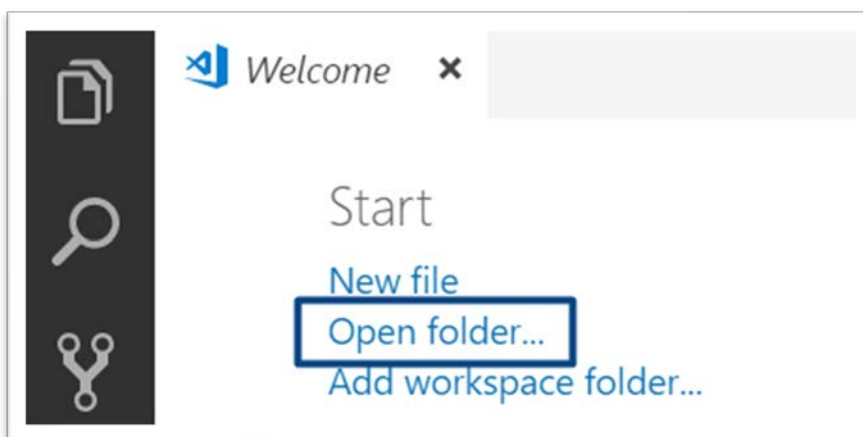
0. Празно Visual Studio Code проект (Blank Project)

Създайте празен проект във Visual Studio Code. Ще обединяваме решенията на всички задачи, под формата на отделни файлове в този проект. Тази възможност е изключително удобна, когато искаме да работим по няколко проекта и бързо да превключваме между тях или искаме да обединим логически няколко взаимосвързани проекта. Това ни помага да запазваме решенията на задачите отделно и да ги пазим, за да ги използваме за други задачи или преговор.

1. Стартирайте Visual Studio Code.
2. Създайте нова папка, която ще държи отделните решения. Ще се отвори диалогов прозорец, в който ще трябва да изберете нейната директория. Препоръчително е да именувате папката според темата на заданието, пример "**Conditional-Statements**"



3. След това изберете папката, като работна среда, за да добавите файловете с JavaScript решенията на своите задачи в нея.



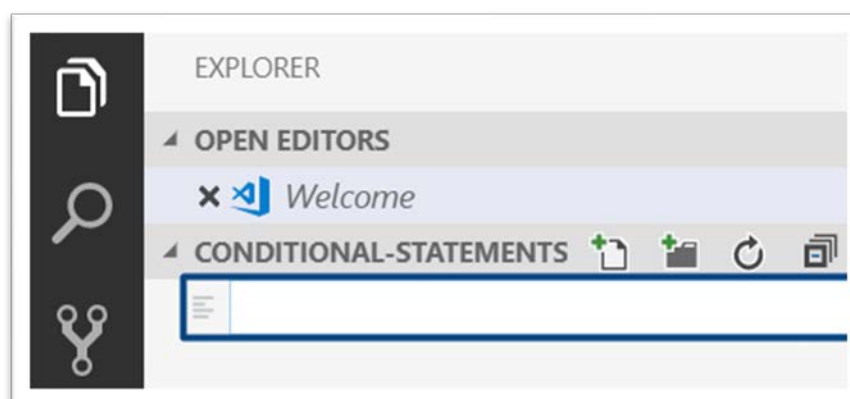
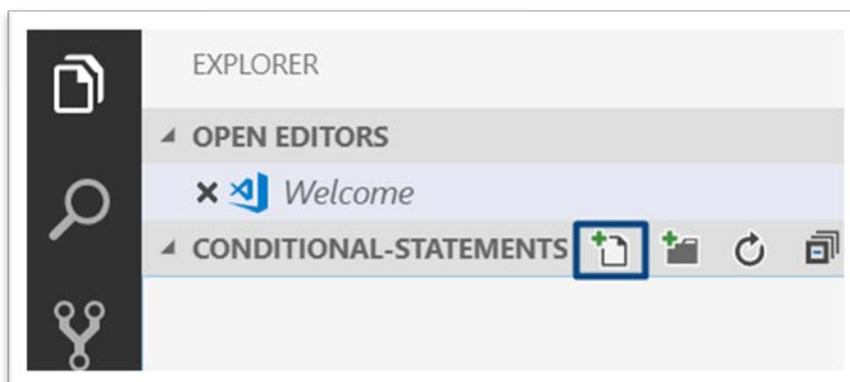
1. Проверка за отлична оценка

Първата задача от тази тема е да се напише **функция**, която **чете оценка**, получена като аргумент и отпечатва "**Excellent!**", ако оценката е **5.50** или по-висока.

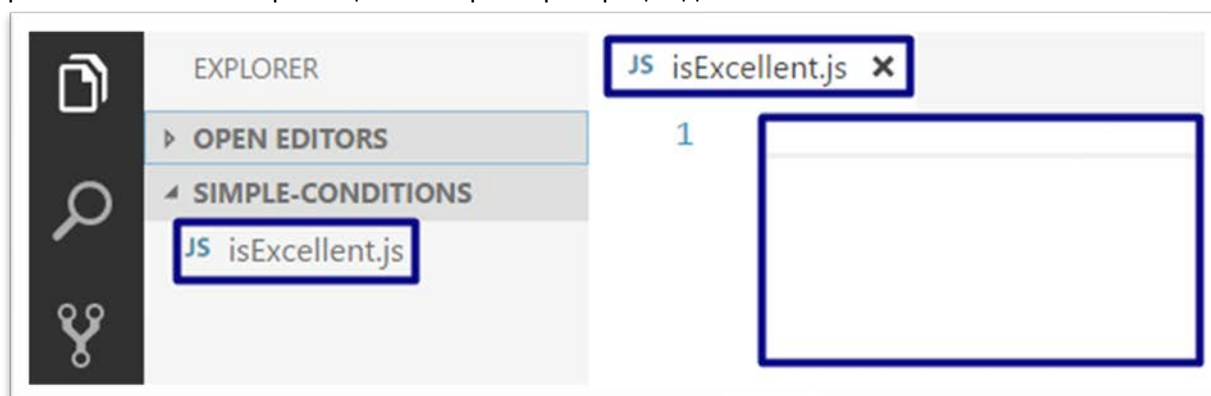
ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
(["6"])	Excellent!	(["5"])	(няма изход)	(["5.50"])	Excellent!	(["5.49"])	(няма изход)

Насоки

1. Създайте **нов JavaScript файл** в съществуващата папка и го именувайте подходящо. Препоръчително е всеки скриптов файл да се казва, както името на задачата чието решение съдържа.



2. Съдържанието на новият файл ще се отвори в прозореца вдясно.



3. Отидете във файла **isExcellent.js** и създайте функцията **isExcellent(input):**.

```
function isExcellent(input) {  
  
}
```

4. Направете проверка за стойността на оценката. Ако тя е по-голяма или равна на 5.50 отпечатайте изхода по условие:

```
function isExcellent(input) {  
    let grade = Number(input[0]);  
    if (grade >= 5.50) {  
        console.log("Excellent!");  
    }  
}
```

5. Извикайте функцията с различни входни стойности и я стартирайте с **Ctrl + F5**:

```
isExcellent(["5.50"]);
```

PROBLEMS OUTPUT DEBUG CONSOLE

```
C:\Program Files\nodejs\node.  
Excellent!
```

```
isExcellent(["5.49"]);
```

PROBLEMS OUTPUT DEBUG CONSOLE TEI

```
C:\Program Files\nodejs\node.exe
```

2. Намиране на по-голямото число

Да се напише функция, която получава **две цели числа** и отпечатва **по-голямото** от двете.

Примерен вход и изход

вход	изход	вход	изход	вход	изход	вход	изход
(["5", "3"])	5	(["3", "5"])	5	(["10", "10"])	10	(["-5", "5"])	5

Насоки

1. Сравнете, дали първото число **num1** е по-голямо от второто **num2**. Отпечатайте по-голямото число.

```
function greaterNumber(input) {  
    let num1 = Number(input[0]);  
    let num2 = Number(input[1]);  
    if (num1 > num2) {  
        console.log(num1);  
    } else {  
        console.log(num2);  
    }  
}
```

3. Четно или нечетно

Да се напише функция, която получава **цяло число** като аргумент и отпечата на конзолата, дали е **четно** или **нечетно**.

Примерен вход и изход

ВХОД	ИЗХОД
(["2"])	even

ВХОД	ИЗХОД
(["3"])	odd

ВХОД	ИЗХОД
(["25"])	odd

ВХОД	ИЗХОД
(["1024"])	even

Насоки

1. Първо добавете **нов JavaScript файл** към съществуващия проект
2. Проверете, дали числото е четно, като го разделите на 2 и проверите, дали има остатък от делението. Отпечатайте изхода по условие – текста **"even"** или **"odd"**.

```
function oddOrEven(input) {
  let num = Number(input[0]);
  if (num % 2 == 0) {
    console.log("even");
  } else {
    console.log("odd");
  }
}
```

4. Число от 100 до 200

Да се напише функция, която **получава цяло число**, като аргумент и проверява, дали е **под 100**, **между 100 и 200** или **над 200**. Да се отпечата съответно съобщения, като в примерите по-долу:

Примерен вход и изход

вход	изход	вход	изход	вход	изход
(["95"])	Less than 100	(["120"])	Between 100 and 200	(["210"])	Greater than 200

5. Познай паролата

Да се напише функция, която **получава парола** (един ред с произволен текст), като аргумент и проверява, дали **съвпада** с фразата "s3cr3t!P@ssw0rd". При съвпадение да се изведе "Welcome". При несъвпадение да се изведе "Wrong password!".

Примерен вход и изход

вход	изход	вход	изход	вход	изход
(["qwerty"])	Wrong password!	(["s3cr3t!P@ssw0rd"])	Welcome	(["s3cr3t!p@ss"])	Wrong password!

6. Лица на фигури

Да се напише функция, която **получава като аргументи вида и размерите на геометрична фигура** и пресмята лицето ѝ. Фигурите са четири вида: квадрат (square), правоъгълник (rectangle), кръг (circle) и триъгълник (triangle). Първият аргумент е вида на фигурата (square, rectangle, circle или triangle). Ако фигурата е **квадрат**, следващия аргумент е едно число - дължина на страната му. Ако фигурата е **правоъгълник**, следващите два аргумента са две числа - дължините на страните му. Ако фигурата е **кръг**, следващия аргумент е число - радиусът на кръга. Ако фигурата е **триъгълник**, следващите два аргумента са две числа - дължината на страната му и дължината на височината към нея. Резултатът да се закръгли до **3 цифри след десетичната точка**.

Примерен вход и изход

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
(["square", "5"])	25.000	(["rectangle", "7", "2.5"])	17.500	(["circle", "6"])	113.097	(["triangle", "4.5", "20"])	45.000

Примерна изпитна задача

7. Магазин за детски играчки

Петя има магазин за детски играчки. Тя получава голяма поръчка, която трябва да изпълни. С парите, които ще спечели иска да отиде на екскурзия. Да се напише функция, която пресмята печалбата от поръчката.

Цени на играчките:

- Пъзел - 2.60 лв.
- Говореща кукла - 3 лв.
- Плюшено мече - 4.10 лв.
- Миньон - 8.20 лв.
- Камионче - 2 лв.

Ако поръчаните играчки са **50 или повече** магазинът прави **отстъпка 25% от общата цена**. От спечелените пари Петя трябва да даде **10% за наема** на магазина. Да се пресметне, дали парите ще ѝ стигнат да отиде на екскурзия.

Вход

Функцията получава **6 аргумента**:

1. Цена на екскурзията - реално число в интервала [1.00 ... 10000.00]
2. Брой пъзели - цяло число в интервала [0... 1000]
3. Брой говорещи кукли - цяло число в интервала [0 ... 1000]
4. Брой плюшени мечета - цяло число в интервала [0 ... 1000]
5. Брой миньони - цяло число в интервала [0 ... 1000]
6. Брой камиончета - цяло число в интервала [0 ... 1000]

Изход

На конзолата се отпечатва:

- Ако парите са достатъчни се отпечатва:
 - "Yes! {оставащите пари} lv left."
- Ако парите НЕ са достатъчни се отпечатва:
 - "Not enough money! {недостигащите пари} lv needed."

Резултатът трябва да се форматира до втория знак след десетичната запетая.

Примерен вход и изход

Вход	Изход	Обяснения
(["40.8", "20", "25", "30", "50", "10"])	Yes! 418.20 lv left.	<p>Сума: $20 * 2.60 + 25 * 3 + 30 * 4.10 + 50 * 8.20 + 10 * 2 = 680$ лв.</p> <p>Брой на играчките: $20 + 25 + 30 + 50 + 10 = 135$</p> <p>135 > 50 => 25% отстъпка; 25% от 680 = 170 лв. отстъпка</p> <p>Крайна цена: $680 - 170 = 510$ лв.</p> <p>Наем: 10% от 510 лв. = 51 лв.</p> <p>Печалба: $510 - 51 = 459$ лв.</p> <p>459 > 40.8 => $459 - 40.8 = 418.20$ лв. остават</p>
Вход	Изход	Обяснения
(["320", "8", "2", "5", "5", "1"])	Not enough money! 238.73 lv needed.	<p>Сума: 90.3 лв.</p> <p>Брой на играчките: 21</p> <p>21 < 50 => няма отстъпка</p> <p>Наем: 10% от 90.3 = 9.03 лв.</p> <p>Печалба: $90.3 - 9.03 = 81.27$ лв.</p> <p>81.27 < 320 => $320 - 81.27 = 238.73$ лв. не достигат</p>