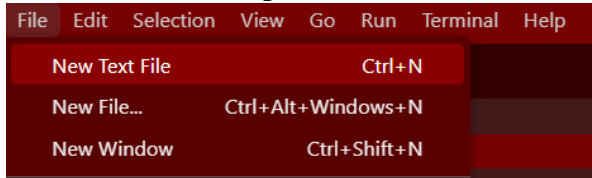


Упражнение 2 - ЕФП

Задача 1. Създайте вашата първа програма „hello world“.

1. От меню File изберете New Text File

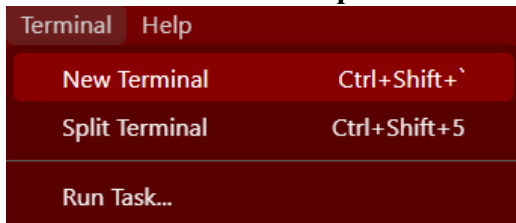


2. Запишете следния код във файла:

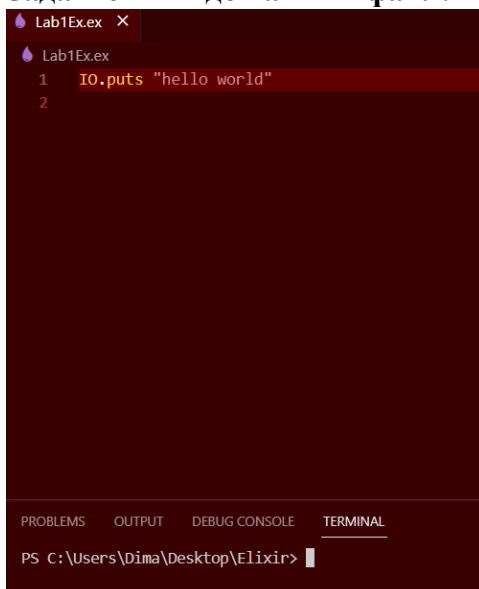
```
IO.puts "hello world"
```

3. Запазете файла с име: Lab1Ex. Разширението се добавя автоматично *.ex

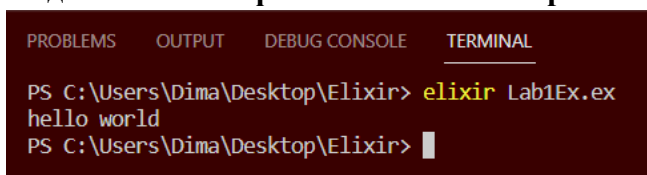
4. От меню Terminal изберете:



5. Задайте пътя до вашият файл:



6. За да изпълните файла запишете в терминала: **elixir името_на_файла.ex**



Задача 2. Създайте вашата първа програма, като използвате интерактивния режим за работа. За целта **запишете в терминала - iex.bat**, ако сте на Windows.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Dima\Desktop\Elixir> iex.bat
Interactive Elixir (1.13.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> █
```

А изход от този режим **Ctrl+C**

```
PS C:\Users\Dima\Desktop\Elixir> iex.bat
Interactive Elixir (1.13.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> 40+2
42
iex(2)> Terminate batch job (Y/N)? █
```

Задача 3. case, cond и if

```
iex(8)> x=1
1
iex(9)> case 10 do
...(9)> ^x->"won,t match"
...(9)> _->"Will match"
...(9)> end
"Will match"
iex(10)> case {1, 2, 3} do
...(10)> {1, x, 3} when x>0->
...(10)> "will match"
...(10)> _->
...(10)> "Won't match"
...(10)> end
"will match"
```

Задача 4. Изрази в защитни клаузи (Expressions in guard clauses)

```

iex(11)> hd(1)
** (ArgumentError) errors were found at the given arguments:
iex(11)> case 1 do
...(11)> x when hd(x)->"Won't match"
...(11)> x->"Got: #{x}"
...(11)> end
"Got: 1"
iex(12)> case :ok do
...(12)> :error ->"Won't match"
...(12)> end
** (CaseClauseError) no case clause matching: :ok

iex(12)> f= fn
...(12)> x, y when x>0 ->x+y
...(12)> x, y->x * y
...(12)> end
#Function<43.65746770/2 in :erl_eval.expr/5>
iex(13)> f.(1, 3)
4
iex(14)> f.(-1, 3)
-3

```

Задача 5.cond

```

iex(16)> cond do
...(16)> 2+2==5->
...(16)> "This will not be true"
...(16)> 2 * 2==3->
...(16)> "Nor this"
...(16)> 1+1==2->
...(16)> "But this will"
...(16)> end
"But this will"

```

```

iex(17)> cond do
...(17)> 2+2==5->
...(17)> "This is never true"
...(17)> 2*2==3->
...(17)> "Nor this"
...(17)> true->
...(17)> "This is always true (equivalent to else)"
...(17)> end
"This is always true (equivalent to else)"

```

Задача 6. If и unless

```
iex(18)> if true do
...(18)> "This works!"
...(18)> end
"This works!"
iex(19)> unless true do
...(19)> "This will never be seen"
...(19)> end
nil
```

```
iex(20)> if nil do
...(20)> "This won't be seen"
...(20)> else
...(20)> "This will"
...(20)> end
"This will"
iex(21)> "This will"
"This will"
```

Задача do/ end блокове

```
iex(22)> if true do
...(22)> a=1+2
...(22)> a+10
...(22)> end
13
iex(23)> if true, do: (
...(23)> a=1+2
...(23)> a+10
...(23)> )
13
iex(24)> if true, do: 1+2
3
```

```
iex(25)> is_number if true do
...(25)> 1+2
...(25)> end
** (CompileError) iex:25: undefined function is_number/2 (there is no such import)

iex(25)> is_number(if true do
...(25)> 1+2
...(25)> end)
true
```

Задачи

Задача 1. Да се напише програма, която въвежда от клавиатурата реално число r . Програмата да изчислява и отпечата периметъра на окръжност с радиус r .

Задача 2. Да се състави програма, която въвежда от клавиатурата две различни реални числа. Програмата да изведе по-голямото от тях.

Задача 3. Да се състави програма, която въвежда последователно от клавиатурата три реални числа и извежда най-голямото от тях.

Входни данни:

a	b	c
1	2	3
2	1	3
3	2	1
3	1	2
1	3	2
2	3	1

Задача 4. Да се напише програма, която въвежда от клавиатурата 3 реални, положителни числа и ги сортира във възходящ ред. Въвеждането на числата става на един ред, разделени с интервал. Сортираните числа се извеждат на един ред, разделени с интервал.

Задачи за самостоятелна работа

Задача 5. Напишете програма, която намира корена на линейното уравнение $a \cdot x + b = 0$. На изхода се извежда стойността на x и подходящо съобщение.

Задача 6. Напишете програма, която определя в кой квадрант лежи точката A с координати (x_a, y_a) . Предполага се, че x не лежи на никоя от координатните оси.

Задача 7. Да се напише програма, която въвежда символ от клавиатурата. Да се провери и отпечата дали въведеният символ е число.

Упътване: Използвайте кодовете на символите. Числата от 0 до 9 имат поредни кодове в ASCII таблицата.

Задача 8. Да се напише програма, която въвежда от клавиатурата цяло трицифрено число. Да се провери дали сумата от цифрите на числото е четно число.

Задача 9. Да се напише програма, която въвежда от клавиатурата цяло четирицифрено число. Да се провери дали произведението на цифрите е кратно на 3.

Задача 10. Да се напише програма, която въвежда от клавиатурата цяло четирицифрено число. Да се провери дали числото е симетрично, т.е. записът му отляво надясно и отдясно наляво е еднакъв.

Задача 11*. Да се напише програма, която въвежда от клавиатурата три цели числа, различни от нула. Програмата да извежда разликата между максималното четно число и минималното нечетно число между тях. Ако въведените числа са само нечетни, да извежда минималното от тях. Ако са само четни – максималното от тях.

Задача 12. Да се напише програма, която въвежда от клавиатурата трицифрено число n . Програмата проверява дали цифрите му са различни. Ако са различни, извежда „yes“ и произведението на първата и трета цифра на числото. В противен случай, извежда „no“ и числото изписано в обратен ред.