

# Hands-on RegularExpressions

Exercise 1: Check if the string "hello" exists in the sentence "Hello, world!"

# ➤ Use `re.search()` to check for a match

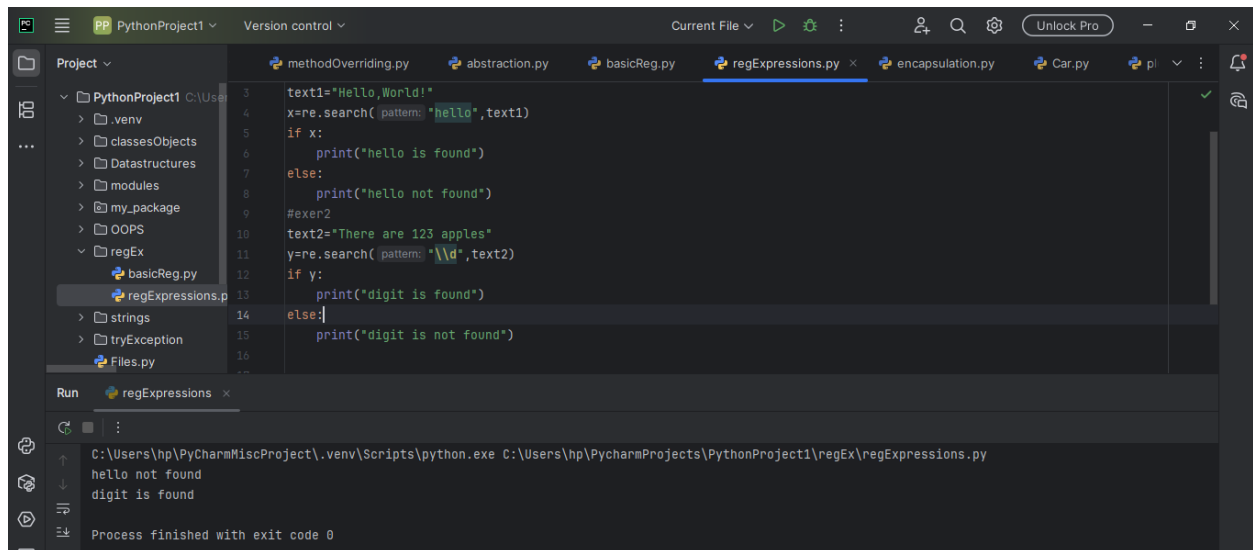
# Exercise 2: Search for any number in the string "There are 123 apples"

# ➤ Use `re.search()` to find the number

**Code:**

```
import re
#exer1
text1="Hello,World!"
x=re.search("hello",text1)
if x:
    print("hello is found")
else:
    print("hello not found")
#exer2
text2="There are 123 apples"
y=re.search("\d",text2)
if y:
    print("digit is found")
else:
    print("digit is not found")
```

**Output)**

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, search, and a 'Unlock Pro' button. The left sidebar displays a project tree for 'PythonProject1' with folders like 'venv', 'classesObjects', 'Datastructures', 'modules', 'my\_package', 'OOPS', 'regEx', 'strings', 'tryException', and 'Files.py'. The 'regEx' folder is expanded, showing files 'basicReg.py' and 'regExpressions.p'. The main editor window displays the code from the previous block, with line numbers 3 through 16. The bottom panel, titled 'Run', shows the execution command: 'C:\Users\hp\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\hp\PyCharmProjects\PythonProject1\regEx\regExpressions.py'. The output of the program is displayed below the command: 'hello not found' followed by 'digit is found'. At the bottom, it states 'Process finished with exit code 0'.

Exercise 3: Extract the first email from the text "Contact me at test@example.com for more info"

# ➤ Use `re.search()` with an appropriate email regex pattern

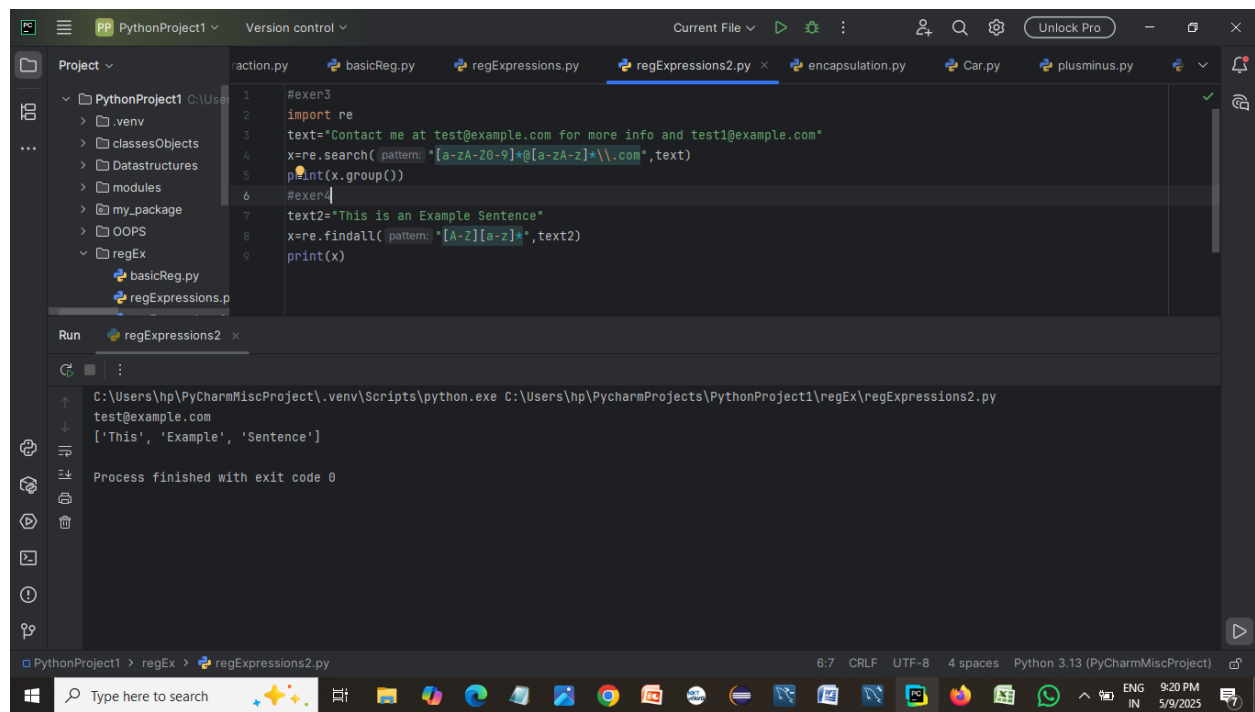
# Exercise 4: Extract all words starting with a capital letter from the string "This is an Example Sentence"

# ➤ Use `re.findall()` to find all such words

### Code )

```
#exer3
import re
text="Contact me at test@example.com for more info and test1@example.com"
x=re.search("[a-zA-Z0-9]*@[a-zA-z]*\\.com",text)
print(x.group())
#exer4
text2="This is an Example Sentence"
x=re.findall("[A-Z][a-z]*",text2)
print(x)
```

### Output )

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, a search icon, and a 'Unlock Pro' button. The 'Project' sidebar on the left shows a tree view of the project structure, including folders like '.venv', 'classesObjects', 'Datastructures', 'modules', 'my\_package', 'OOPS', and 'regEx', along with files 'basicReg.py' and 'regExpressions.p'. The main editor window displays the Python code from the previous block, with line numbers 1 through 9. The 'Run' toolbar at the bottom of the editor shows a play button and a terminal icon. The terminal window at the bottom displays the output of the code: 'test@example.com' on the first line and ['This', 'Example', 'Sentence'] on the second line, followed by the message 'Process finished with exit code 0'. The status bar at the very bottom indicates the file path 'PythonProject1 > regEx > regExpressions2.py', encoding 'UTF-8', and the Python version 'Python 3.13 (PyCharmMiscProject)'. The Windows taskbar at the bottom shows the search bar and various application icons, with the system clock displaying '9:20 PM 5/9/2025'.

Exercise 5: Check if a string contains a valid phone number format (e.g., 123-456-7890)

# ➤ Use `re.match()` or `re.search()` with a regex pattern for a phone number

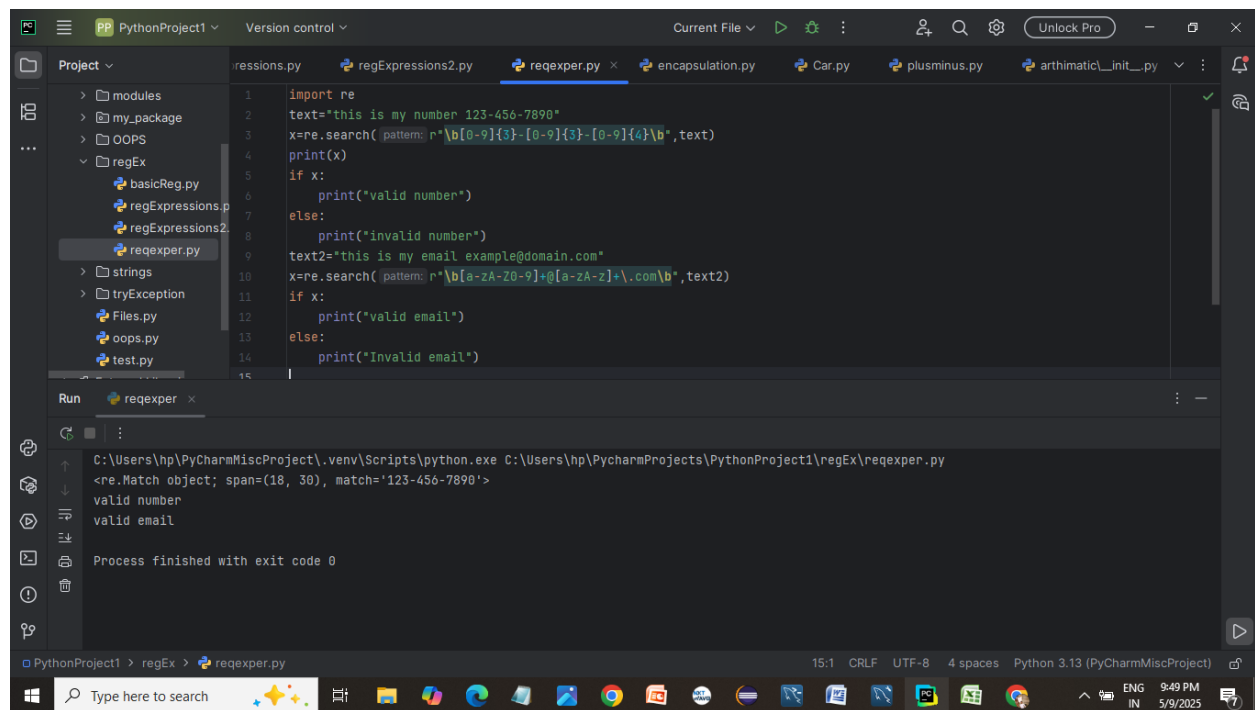
# Exercise 6: Validate if a string is a valid email address using a regex pattern

# ➤ Example: "example@domain.com"

### Code )

```
import re
text="this is my number 123-456-7890"
x=re.search(r"\b[0-9]{3}-[0-9]{3}-[0-9]{4}\b",text)
print(x)
if x:
    print("valid number")
else:
    print("invalid number")
text2="this is my email example@domain.com"
x=re.search(r"\b[a-zA-Z0-9]+@[a-zA-z]+\b",text2)
if x:
    print("valid email")
else:
    print("Invalid email")
```

### Output)



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `reqexper.py` with the following code:

```
1 import re
2 text="this is my number 123-456-7890"
3 x=re.search(pattern=r"\b[0-9]{3}-[0-9]{3}-[0-9]{4}\b",text)
4 print(x)
5 if x:
6     print("valid number")
7 else:
8     print("invalid number")
9 text2="this is my email example@domain.com"
10 x=re.search(pattern=r"\b[a-zA-Z0-9]+@[a-zA-z]+\b",text2)
11 if x:
12     print("valid email")
13 else:
14     print("Invalid email")
15
```

The Run window at the bottom shows the output of the script:

```
C:\Users\hp\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\hp\PyCharmProjects\PythonProject1\regEx\reqexper.py
<re.Match object; span=(18, 30), match='123-456-7890'>
valid number
valid email
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, the line length is 15:1, and the Python version is 3.13 (PyCharmMiscProject).

Exercise 7: Replace all occurrences of "apple" with "orange" in the string "I like apple pie and apple juice"

# ➤ Use `re.sub()` to replace the text

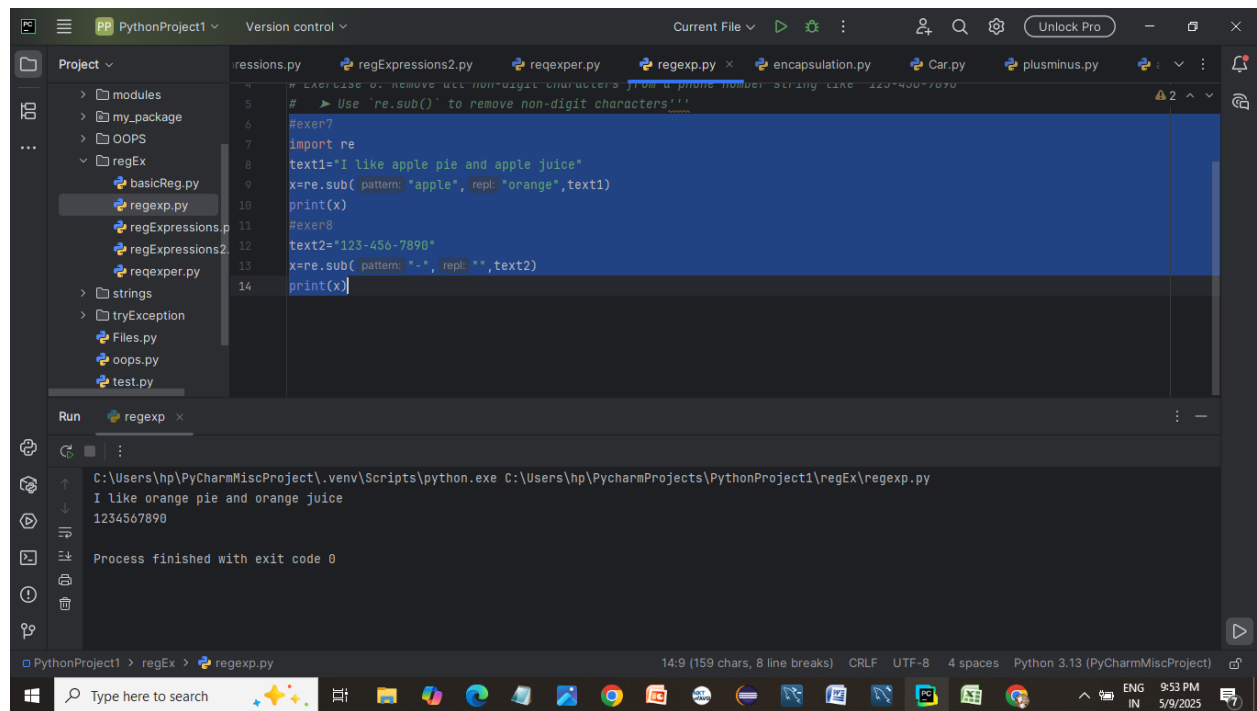
# Exercise 8: Remove all non-digit characters from a phone number string like "123-456-7890"

# ➤ Use `re.sub()` to remove non-digit characters

### Code )

```
#exer7
import re
text1="I like apple pie and apple juice"
x=re.sub("apple","orange",text1)
print(x)
#exer8
text2="123-456-7890"
x=re.sub("-", "", text2)
print(x)
```

### Output)



The screenshot shows the PyCharm IDE interface. The main editor window displays the Python code for Exercise 7 and Exercise 8. The code is as follows:

```
#exer7
import re
text1="I like apple pie and apple juice"
x=re.sub("apple","orange",text1)
print(x)
#exer8
text2="123-456-7890"
x=re.sub("-", "", text2)
print(x)
```

The Run window at the bottom shows the output of the code:

```
C:\Users\hp\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\hp\PyCharmProjects\PythonProject1\regEx\regexp.py
I like orange pie and orange juice
1234567890
Process finished with exit code 0
```

The status bar at the bottom indicates the file is `regEx\regexp.py` with 149 characters, 8 line breaks, CRLF line endings, UTF-8 encoding, 4 spaces indentation, and Python 3.13 interpreter.

Exercise 9: Find all the digits in the string "abc123def456ghi"

# ➤ Use `re.findall()` to extract all digits

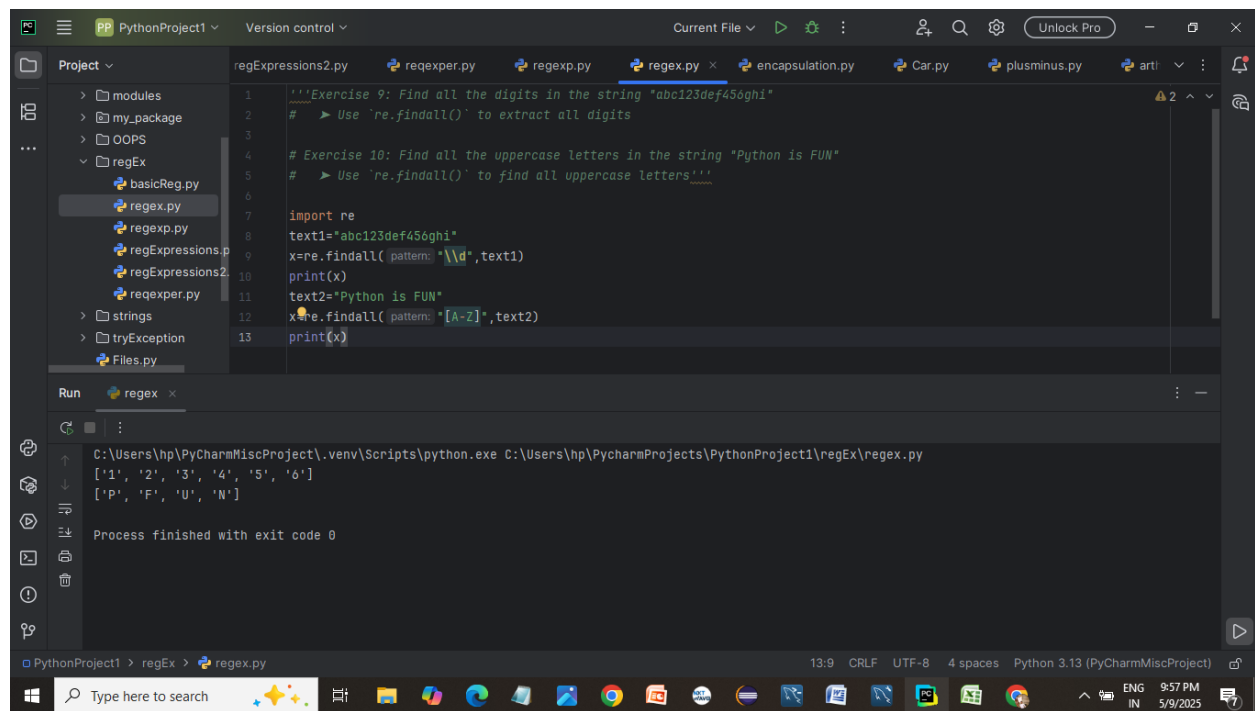
# Exercise 10: Find all the uppercase letters in the string "Python is FUN"

# ➤ Use `re.findall()` to find all uppercase letters

### Code )

```
import re
text1="abc123def456ghi"
x=re.findall("\\d",text1)
print(x)
text2="Python is FUN"
x=re.findall("[A-Z]",text2)
print(x)
```

### Output )



The screenshot shows the PyCharm IDE interface. The main editor window displays the Python code for Exercise 9 and Exercise 10. The code is as follows:

```
1 '''Exercise 9: Find all the digits in the string 'abc123def456ghi'
2 # ➤ Use 're.findall()' to extract all digits
3
4 # Exercise 10: Find all the uppercase letters in the string 'Python is FUN'
5 # ➤ Use 're.findall()' to find all uppercase letters'''
6
7 import re
8 text1="abc123def456ghi"
9 x=re.findall(pattern="\\d",text1)
10 print(x)
11 text2="Python is FUN"
12 x=re.findall(pattern="[A-Z]",text2)
13 print(x)
```

The Run console at the bottom shows the output of the code:

```
C:\Users\hp\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\hp\PyCharmProjects\PythonProject1\regEx\regex.py
['1', '2', '3', '4', '5', '6']
['P', 'F', 'U', 'N']
Process finished with exit code 0
```

The status bar at the bottom indicates the file is `regex.py` in the `regEx` directory of `PythonProject1`, using Python 3.13 (PyCharmMiscProject) with 4 spaces indentation, CRLF line endings, and UTF-8 encoding. The system clock shows 9:57 PM on 5/9/2025.

Exercise 11: Extract the area code from a phone number string like "(123) 456-7890"

# ➤ Use `re.search()` with capturing groups to extract the area code

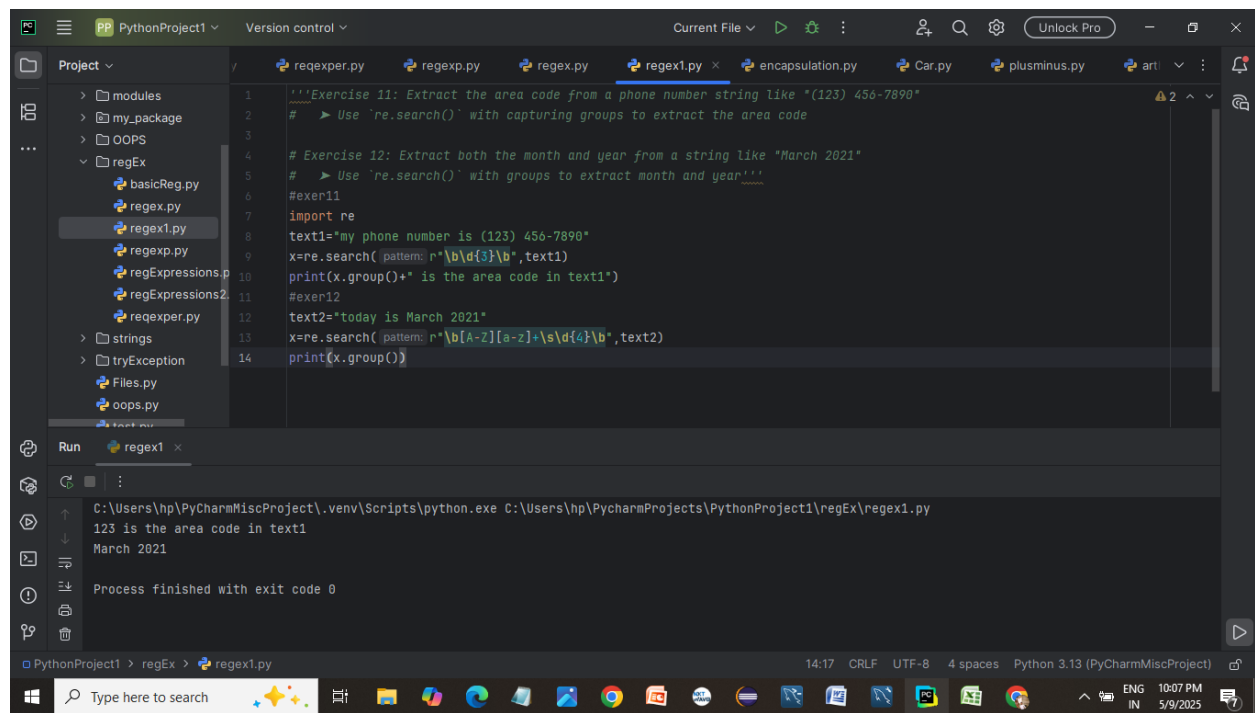
# Exercise 12: Extract both the month and year from a string like "March 2021"

# ➤ Use `re.search()` with groups to extract month and year

### Code )

```
#exer11
import re
text1="my phone number is (123) 456-7890"
x=re.search(r"\b\d{3}\b",text1)
print(x.group()+" is the area code in text1")
#exer12
text2="today is March 2021"
x=re.search(r"\b[A-Z][a-z]+\s\d{4}\b",text2)
print(x.group())
```

### Output)



The screenshot shows the PyCharm IDE interface. The main editor window displays the Python code for Exercise 11 and Exercise 12. The code is as follows:

```
'''Exercise 11: Extract the area code from a phone number string like "(123) 456-7890"
# ➤ Use 're.search()' with capturing groups to extract the area code

# Exercise 12: Extract both the month and year from a string like "March 2021"
# ➤ Use 're.search()' with groups to extract month and year'''

#exer11
import re
text1="my phone number is (123) 456-7890"
x=re.search(pattern=r"\b\d{3}\b",text1)
print(x.group()+" is the area code in text1")
#exer12
text2="today is March 2021"
x=re.search(pattern=r"\b[A-Z][a-z]+\s\d{4}\b",text2)
print(x.group())
```

The Run window at the bottom shows the output of the code:

```
C:\Users\hp\PyCharmMiscProject\venv\Scripts\python.exe C:\Users\hp\PyCharmProjects\PythonProject1\regEx\regex1.py
123 is the area code in text1
March 2021
Process finished with exit code 0
```

The status bar at the bottom indicates the file is 'regex1.py' in the 'regEx' directory of 'PythonProject1'. The encoding is UTF-8, and the line length is 14:17. The system clock shows 10:07 PM on 5/9/2025.

Exercise 13: Check if a string starts with "Hello" and ends with "world"

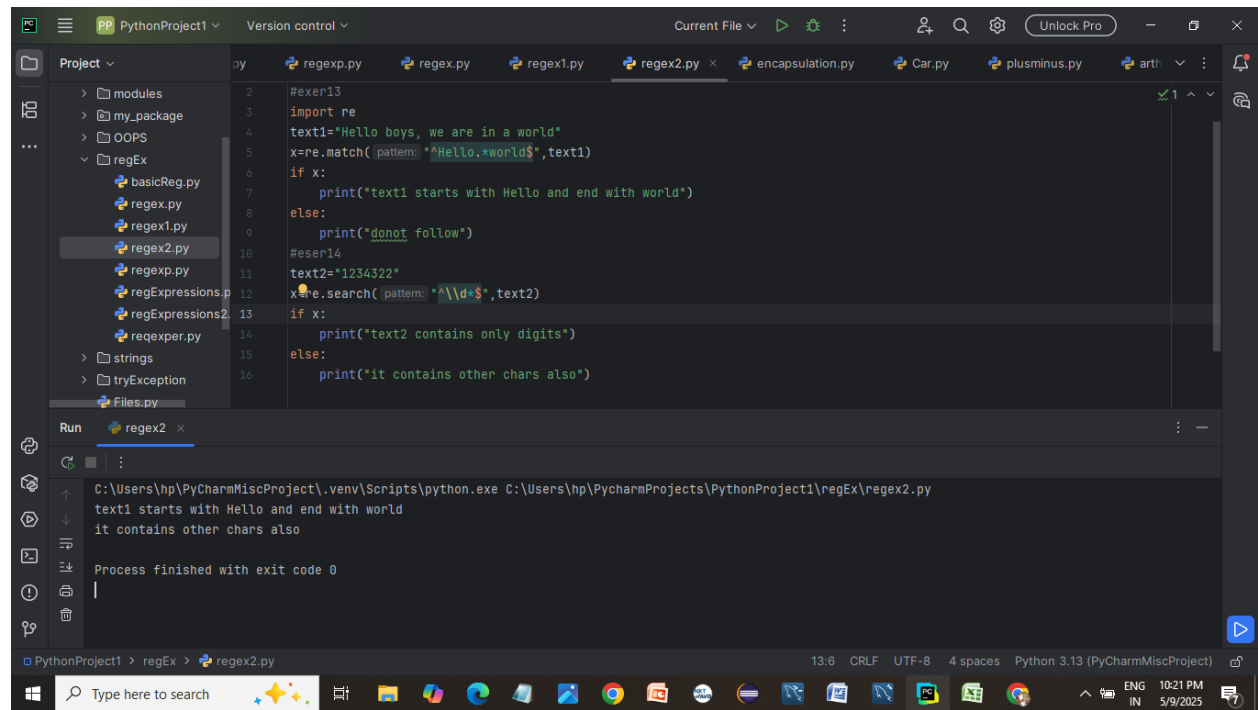
# ➤ Use `re.match()` with `^` (start) and `\$` (end) anchors

# Exercise 14: Check if a string contains only digits (using `\\d` and `^` for start, `\$`

### Code)

```
#exer13
import re
text1="Hello boys, we are in a world"
x=re.match("^Hello.*world$",text1)
if x:
    print("text1 starts with Hello and end with world")
else:
    print("donot follow")
#eser14
text2="1234322"
x=re.search("^\\d*$",text2)
if x:
    print("text2 contains only digits")
else:
    print("it contains other chars also")
```

### Output )

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and settings, along with a 'Unlock Pro' button. The left sidebar displays the project structure, including folders like 'modules', 'my\_package', 'OOPS', and 'regEx', and files like 'basicReg.py', 'regex.py', 'regex1.py', 'regex2.py', 'regExp.py', 'regExpressions.py', 'regExpressions2.py', 'reqexper.py', 'strings', 'tryException', and 'Files.py'. The main editor window shows the code from the previous block, with line numbers 2 through 16. The 'Run' button is highlighted. Below the editor, the 'Run' console shows the output: 'text1 starts with Hello and end with world' and 'it contains other chars also'. The status bar at the bottom indicates the file path 'PythonProject1 > regEx > regex2.py', encoding 'UTF-8', and the Python version 'Python 3.13 (PyCharmMiscProject)'. The Windows taskbar at the very bottom shows the search bar and various application icons, with the system clock displaying '10:21 PM 5/9/2025'.