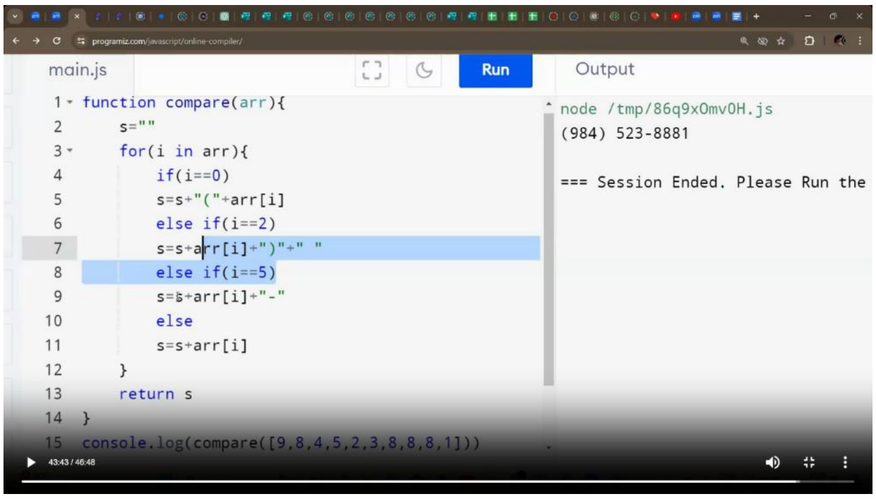
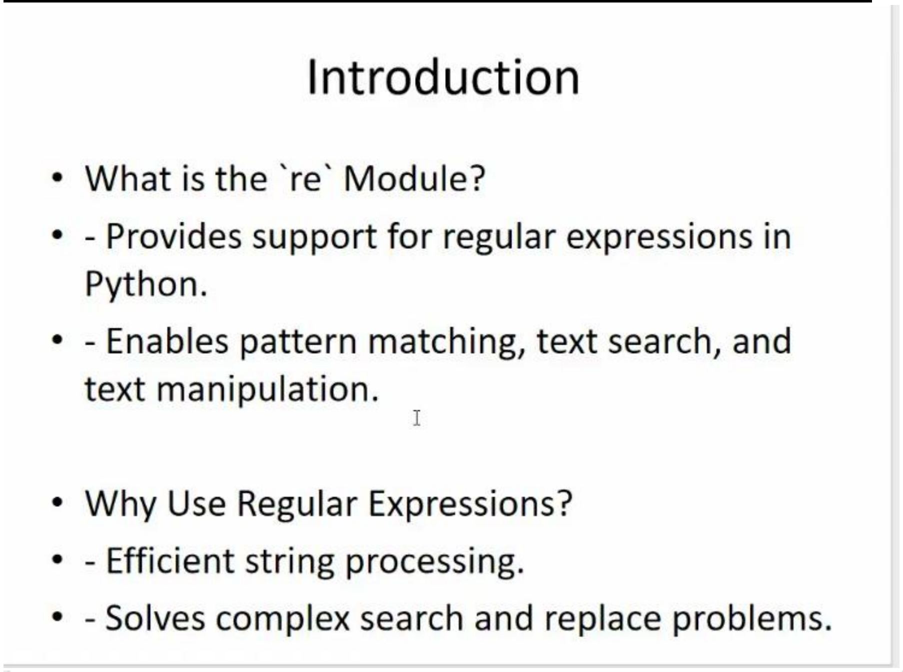


```javascript
function compare(arr){
    s=""
    for(i in arr){
        if(i==0)
        s=s+"("+arr[i]
        else if(i==2)
        s=s+arr[i]+")"+" "
        else if(i==5)
        s=s+arr[i]+"-"
        else
        s=s+arr[i]
    }
    return s
}
console.log(compare([9,8,4,5,2,3,8,8,8,1]))
```

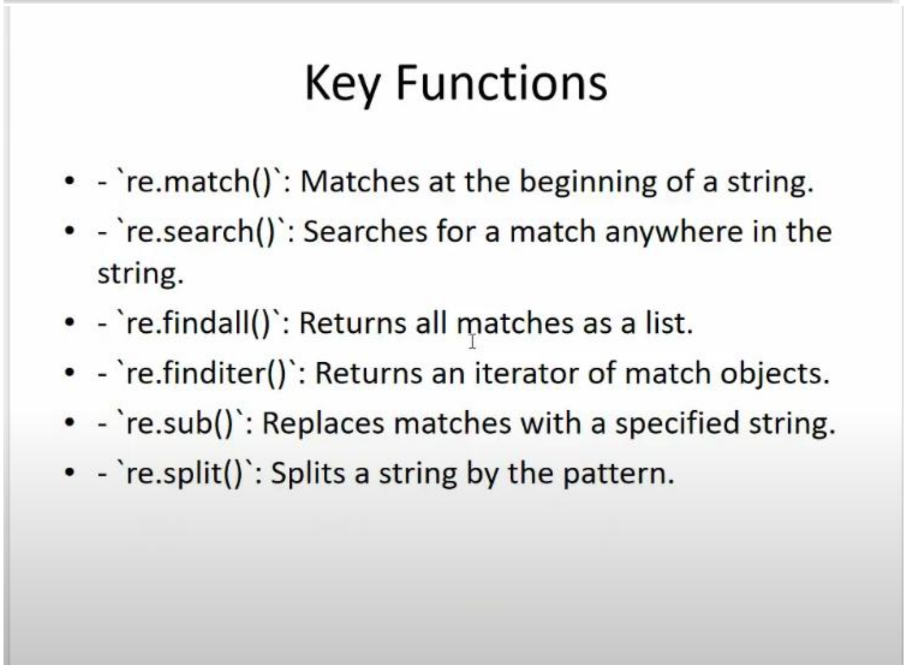
```
node /tmp/86q9xOmv0H.js
(984) 523-8881

=== Session Ended. Please Run the
```



Introduction

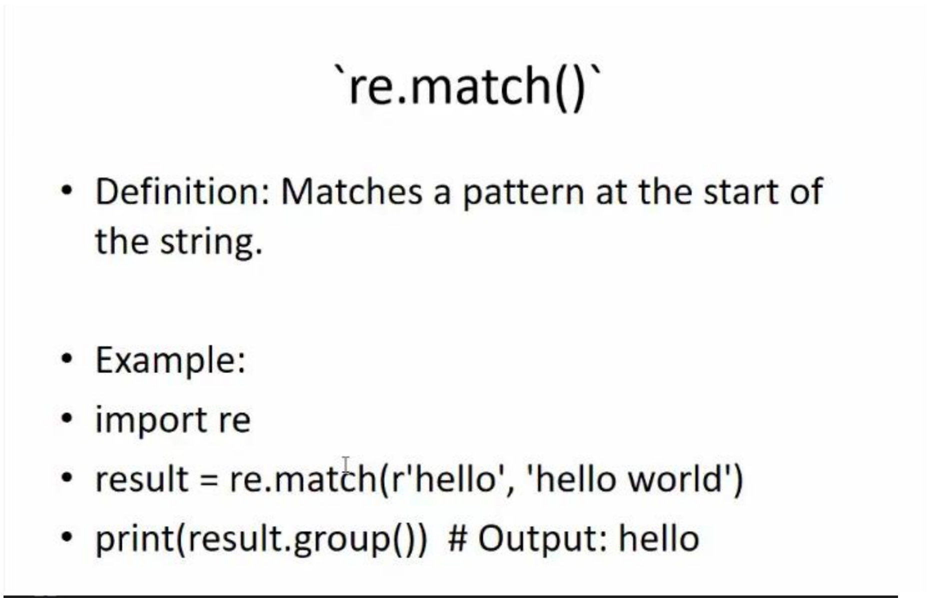
- What is the `re` Module?
- - Provides support for regular expressions in Python.
- - Enables pattern matching, text search, and text manipulation.

- Why Use Regular Expressions?
- - Efficient string processing.
- - Solves complex search and replace problems.



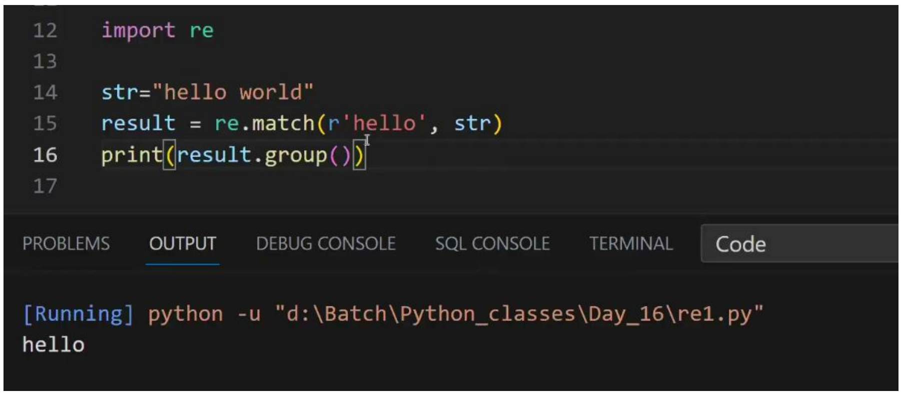
Key Functions

- - `re.match()`: Matches at the beginning of a string.
- - `re.search()`: Searches for a match anywhere in the string.
- - `re.findall()`: Returns all matches as a list.
- - `re.finditer()`: Returns an iterator of match objects.
- - `re.sub()`: Replaces matches with a specified string.
- - `re.split()`: Splits a string by the pattern.



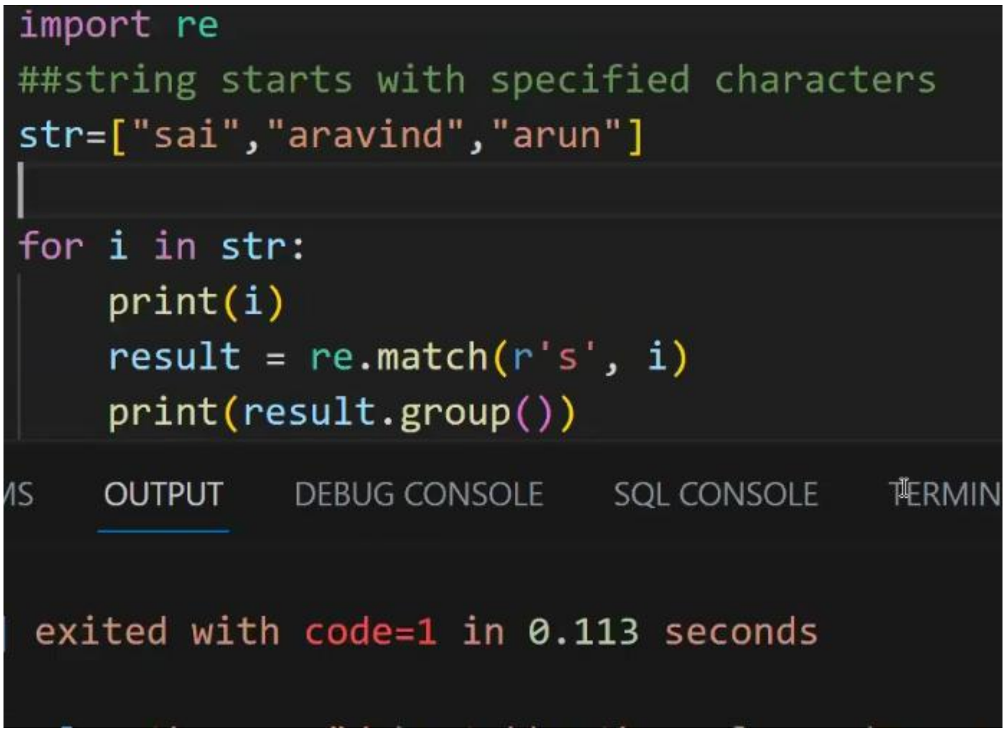
`re.match()`

- Definition: Matches a pattern at the start of the string.
- Example:
- import re
- result = re.match(r'hello', 'hello world')
- print(result.group()) # Output: hello



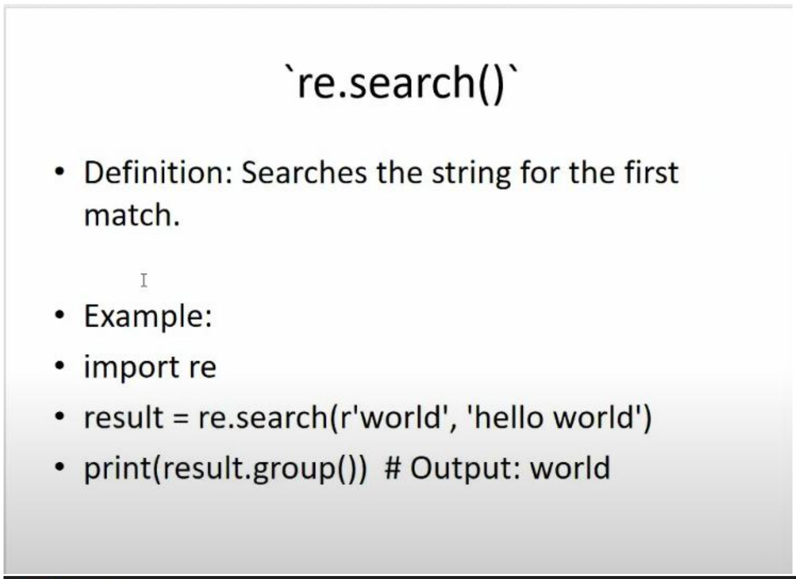
```python
import re

str="hello world"
result = re.match(r'hello', str)
print(result.group())
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
hello
```



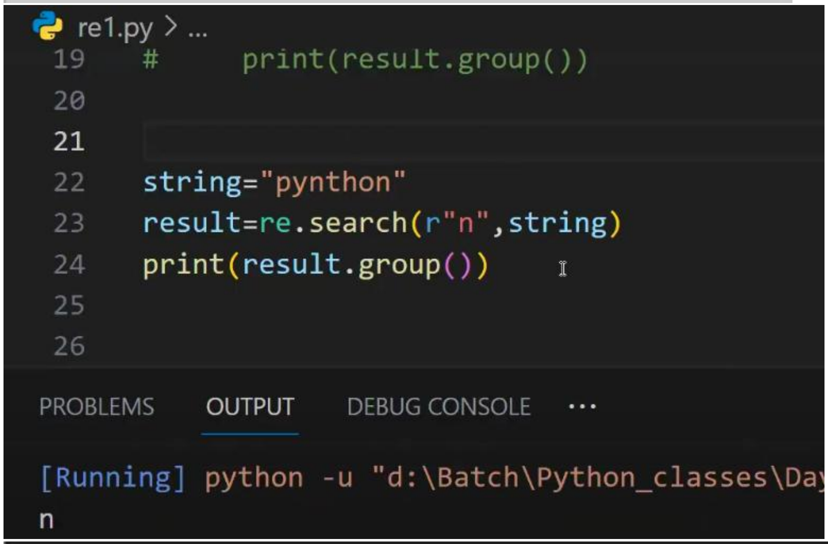
```python
import re
##string starts with specified characters
str=["sai","aravind","arun"]

for i in str:
    print(i)
    result = re.match(r's', i)
    print(result.group())
```

```
exited with code=1 in 0.113 seconds
```



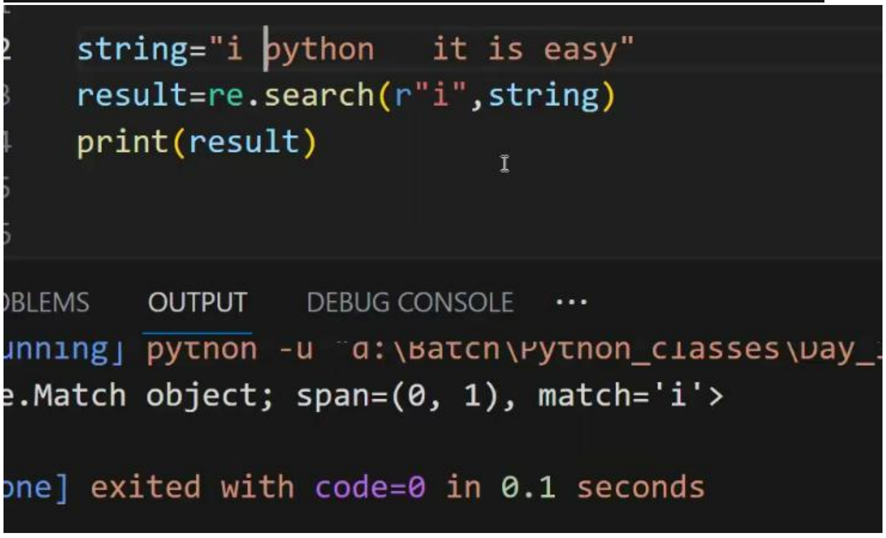
`re.search()`

- Definition: Searches the string for the first match.
- Example:
- import re
- result = re.search(r'world', 'hello world')
- print(result.group()) # Output: world



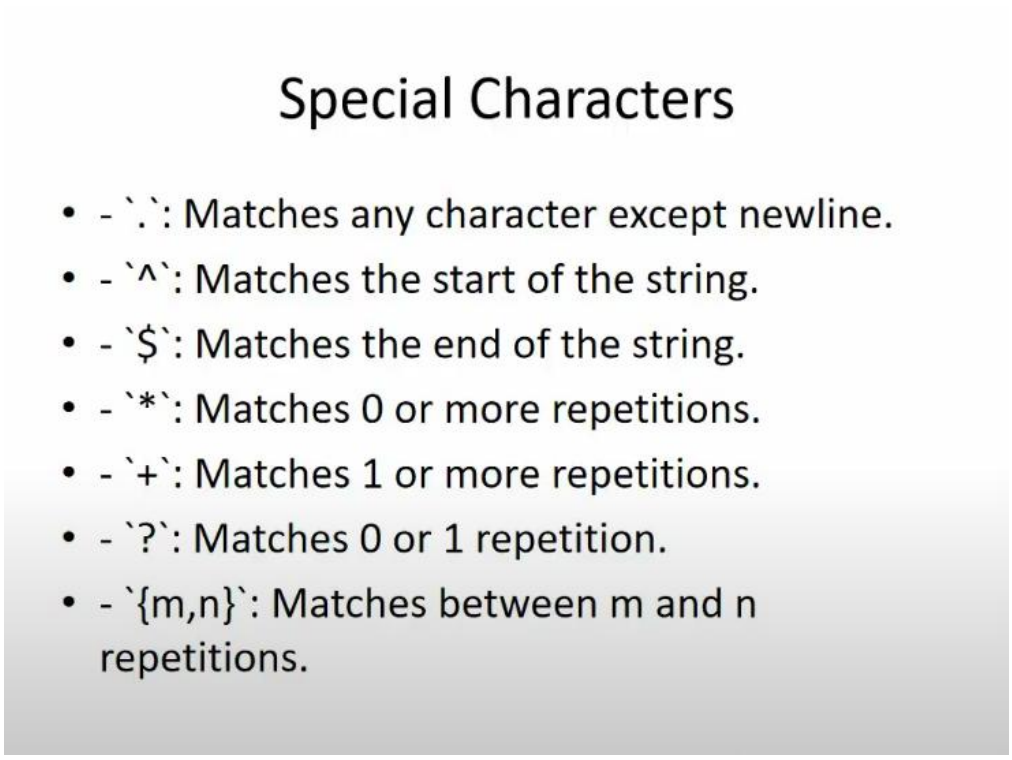
```
#     print(result.group())


string="pynthon"
result=re.search(r"n",string)
print(result.group())
```

```
[Running] python -u "d:\Batch\Python_classes\Day
n
```



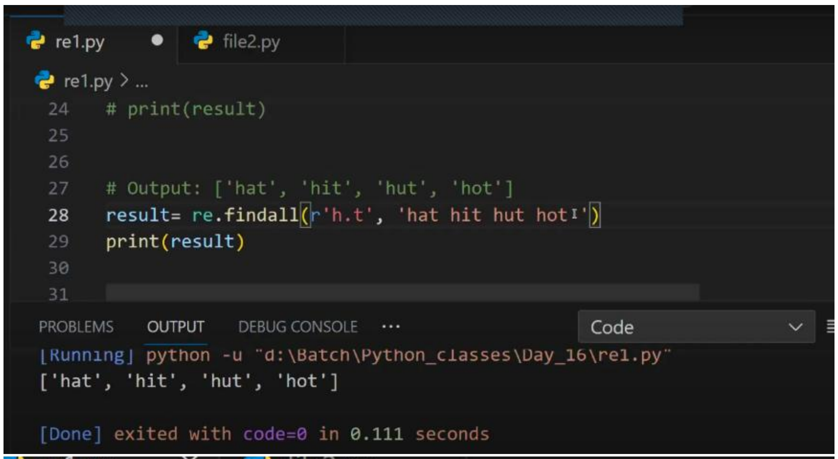
```
string="i python   it is easy"
result=re.search(r"i",string)
print(result)
```

```
unning] python -u "d:\Batch\Python_classes\Day_
e.Match object; span=(0, 1), match='i'>

one] exited with code=0 in 0.1 seconds
```



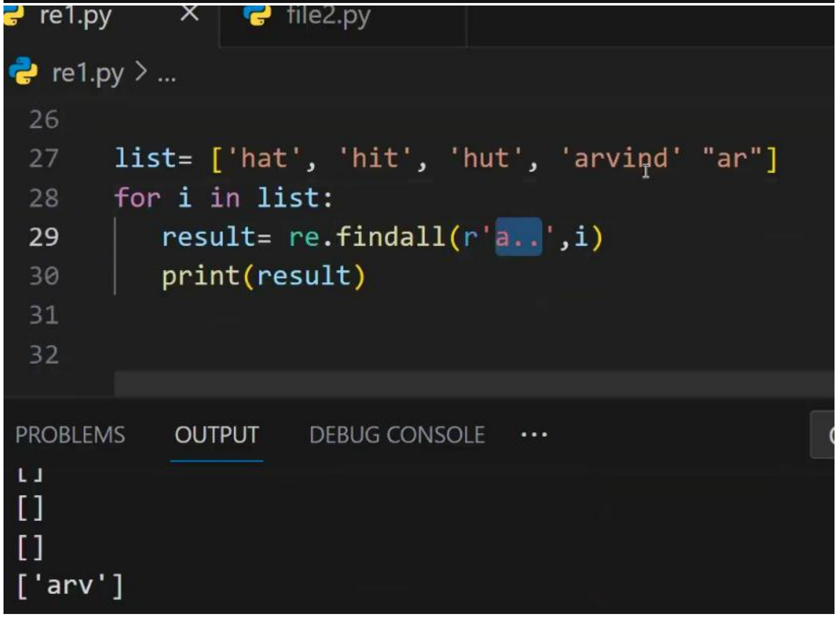
Special Characters

- - `.`: Matches any character except newline.
- - `^`: Matches the start of the string.
- - `$`: Matches the end of the string.
- - `*`: Matches 0 or more repetitions.
- - `+`: Matches 1 or more repetitions.
- - `?`: Matches 0 or 1 repetition.
- - `{m,n}`: Matches between m and n repetitions.



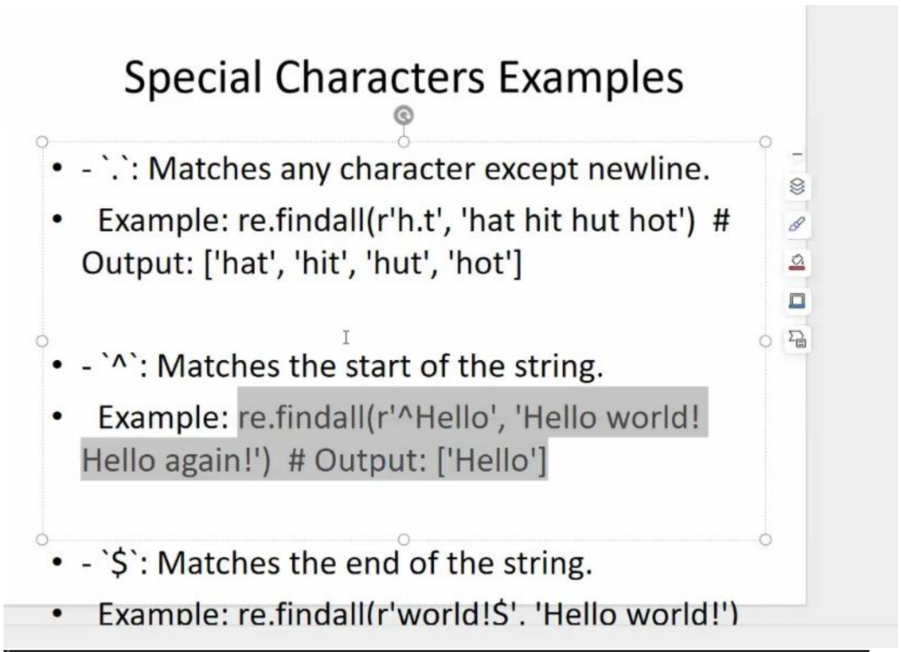
```python
# print(result)


# Output: ['hat', 'hit', 'hut', 'hot']
result= re.findall(r'h.t', 'hat hit hut hot')
print(result)
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
['hat', 'hit', 'hut', 'hot']

[Done] exited with code=0 in 0.111 seconds
```



```python

list= ['hat', 'hit', 'hut', 'arvind' "ar"]
for i in list:
    result= re.findall(r'a..',i)
    print(result)

```

```
[]
[]
['arv']
```

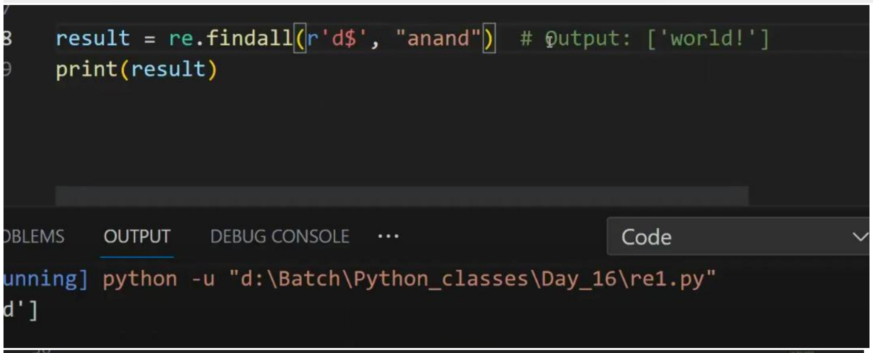


Special Characters Examples

- - `.`: Matches any character except newline.
- Example: re.findall(r'h.t', 'hat hit hut hot') # Output: ['hat', 'hit', 'hut', 'hot']

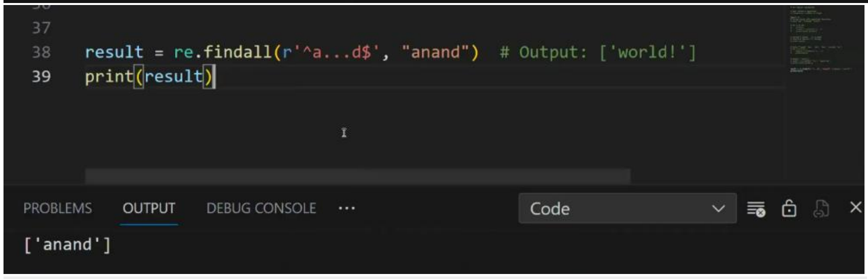
- - `^`: Matches the start of the string.
- Example: re.findall(r'^Hello', 'Hello world! Hello again!') # Output: ['Hello']

- - `$`: Matches the end of the string.
- Example: re.findall(r'world!$', 'Hello world!')



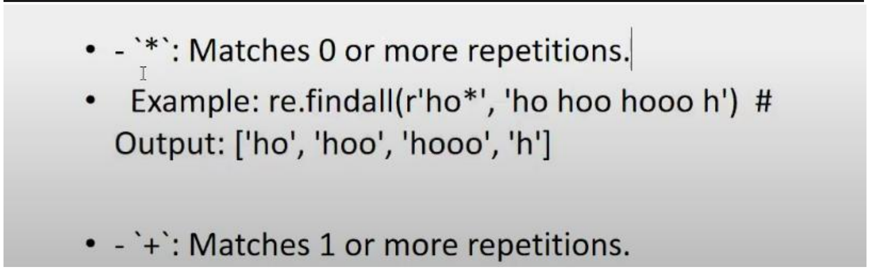
```
result = re.findall(r'd$', "anand")  # Output: ['world!']
print(result)
```

```
unning] python -u "d:\Batch\Python_classes\Day_16\re1.py"
d']
```



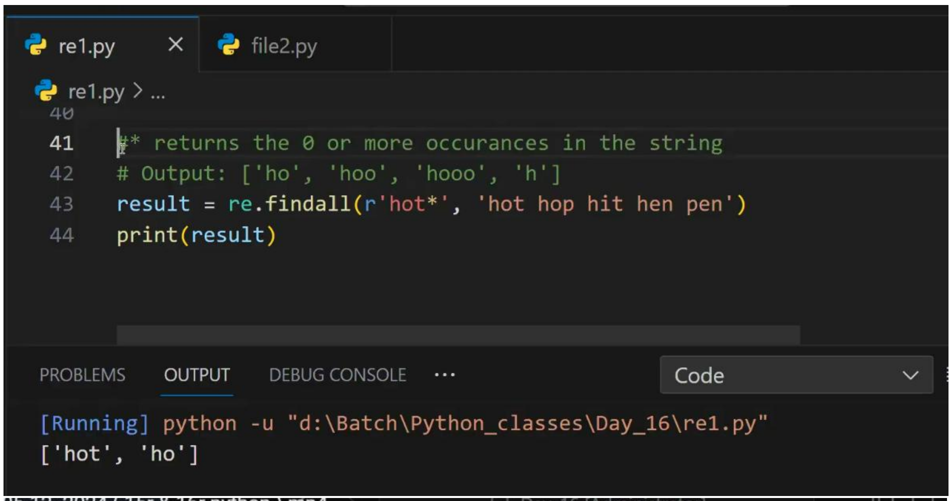
```
result = re.findall(r'^a...d$', "anand")  # Output: ['world!']
print(result)
```

```
['anand']
```



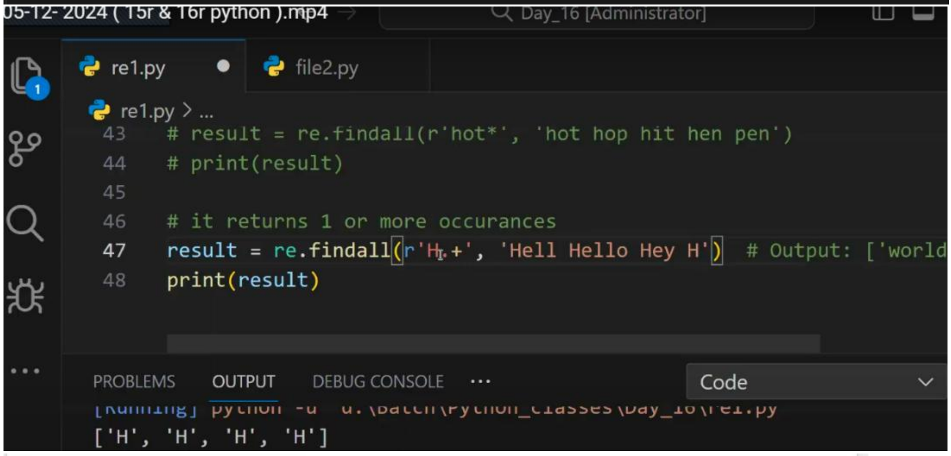
- - `*`: Matches 0 or more repetitions.
- Example: re.findall(r'ho*', 'ho hoo hooo h') # Output: ['ho', 'hoo', 'hooo', 'h']

- - `+`: Matches 1 or more repetitions.



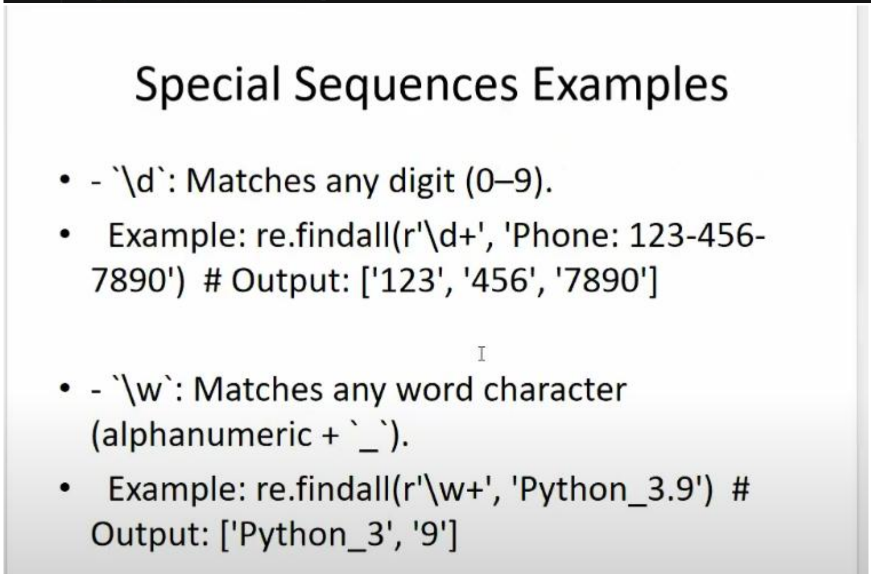
```python
#* returns the 0 or more occurances in the string
# Output: ['ho', 'hoo', 'hooo', 'h']
result = re.findall(r'hot*', 'hot hop hit hen pen')
print(result)
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
['hot', 'ho']
```



```python
# result = re.findall(r'hot*', 'hot hop hit hen pen')
# print(result)

# it returns 1 or more occurances
result = re.findall(r'H+', 'Hell Hello Hey H')  # Output: ['world
print(result)
```

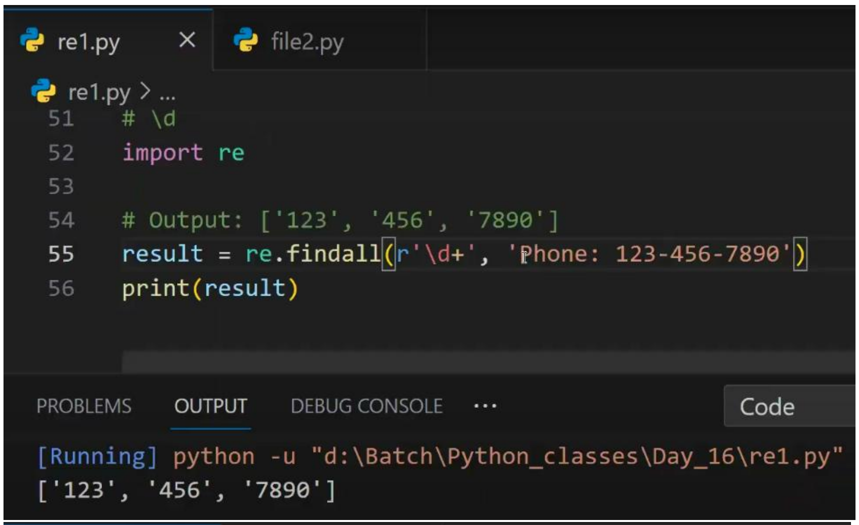
```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
['H', 'H', 'H', 'H']
```



Special Sequences Examples

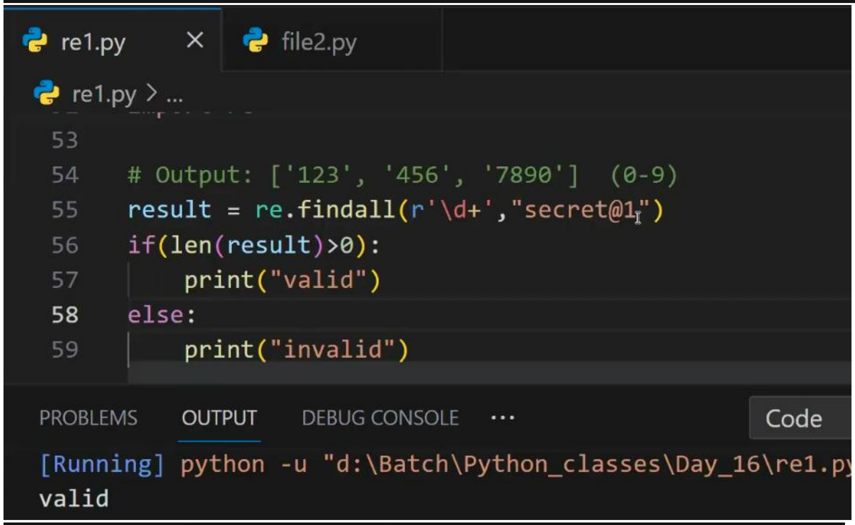
- - `\d`: Matches any digit (0–9).
- Example: re.findall(r'\d+', 'Phone: 123-456-7890') # Output: ['123', '456', '7890']

- - `\w`: Matches any word character (alphanumeric + `_`).
- Example: re.findall(r'\w+', 'Python_3.9') # Output: ['Python_3', '9']



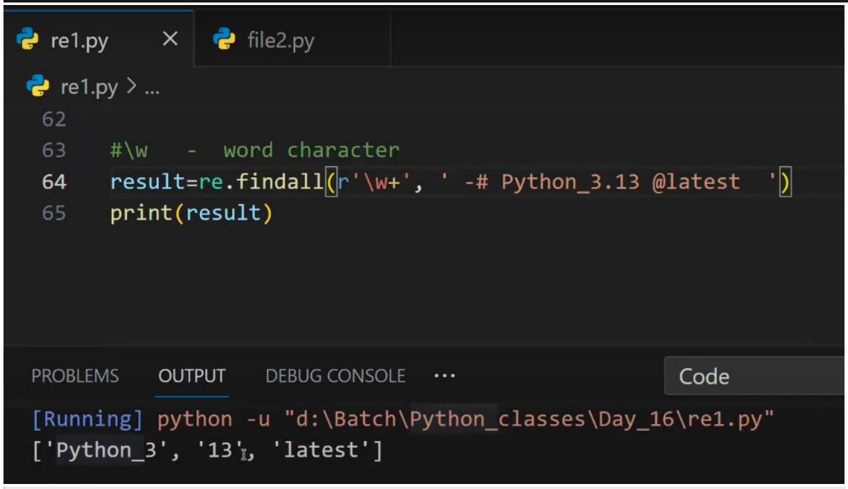
```python
# \d
import re

# Output: ['123', '456', '7890']
result = re.findall(r'\d+', 'Phone: 123-456-7890')
print(result)
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
['123', '456', '7890']
```



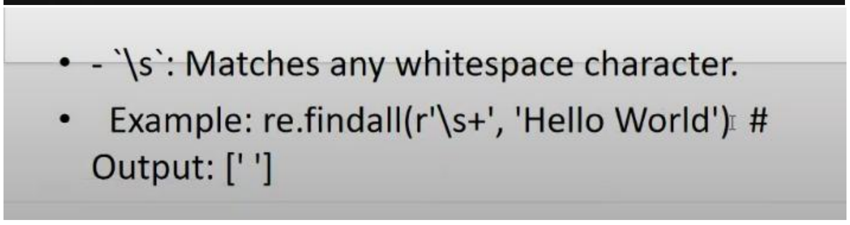
```python

# Output: ['123', '456', '7890']  (0-9)
result = re.findall(r'\d+',"secret@1")
if(len(result)>0):
    print("valid")
else:
    print("invalid")
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py
valid
```

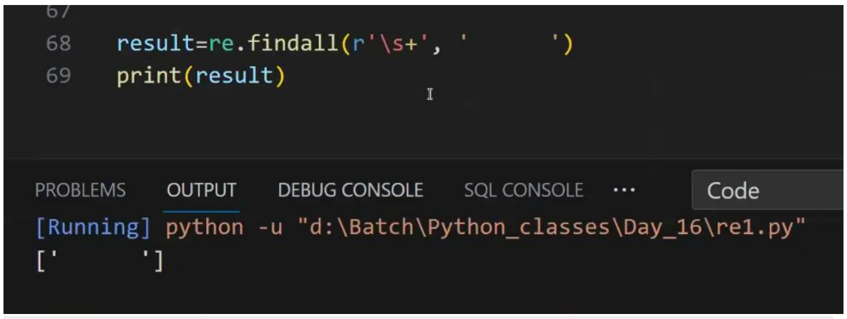


```python

#\w   -  word character
result=re.findall(r'\w+', ' -# Python_3.13 @latest  ')
print(result)
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
['Python_3', '13', 'latest']
```

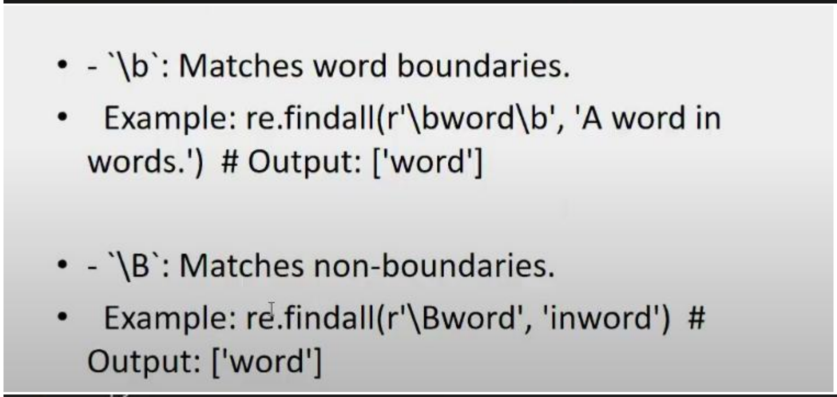


- - `\s`: Matches any whitespace character.
- Example: re.findall(r'\s+', 'Hello World') # Output: [' ']



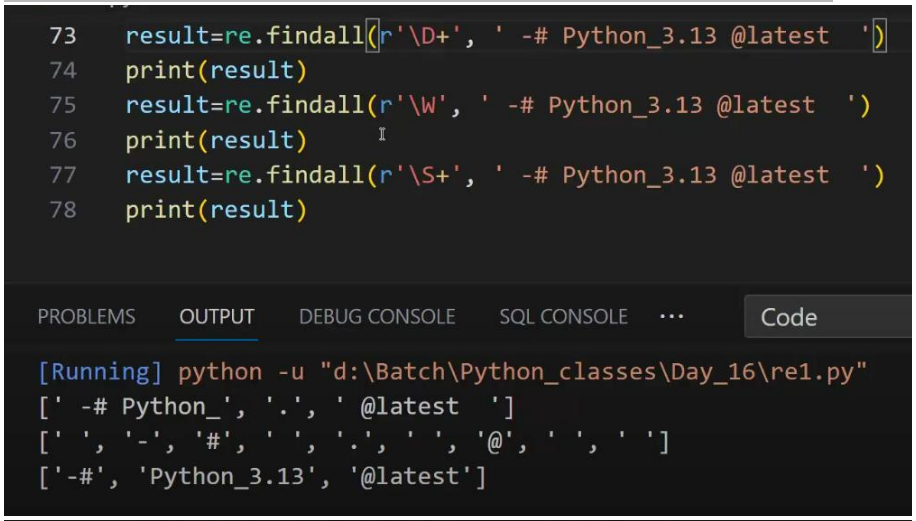
```python
result=re.findall(r'\s+', '       ')
print(result)
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
['       ']
```



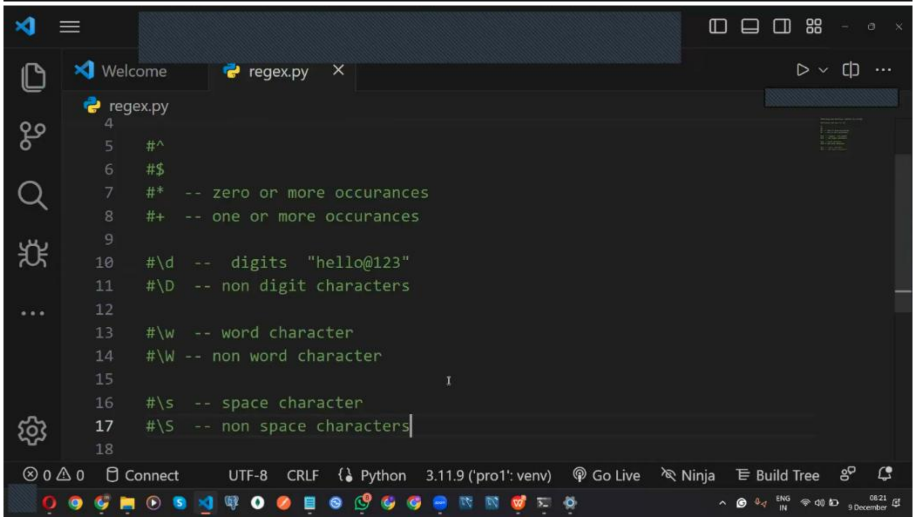
- - `\b`: Matches word boundaries.
- Example: re.findall(r'\bword\b', 'A word in words.') # Output: ['word']

- - `\B`: Matches non-boundaries.
- Example: re.findall(r'\Bword', 'inword') # Output: ['word']

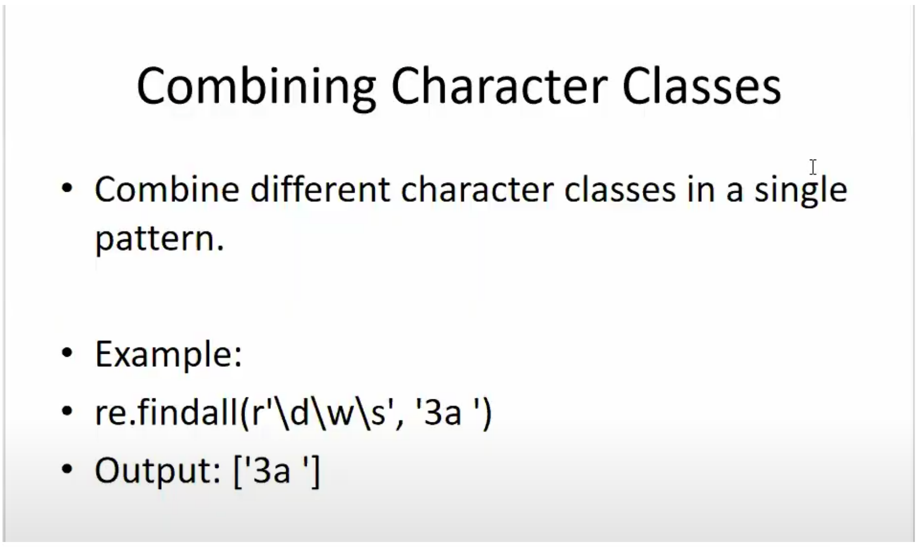


```python
result=re.findall(r'\D+', ' -# Python_3.13 @latest  ')
print(result)
result=re.findall(r'\W', ' -# Python_3.13 @latest  ')
print(result)
result=re.findall(r'\S+', ' -# Python_3.13 @latest  ')
print(result)
```

```
[Running] python -u "d:\Batch\Python_classes\Day_16\re1.py"
[' -# Python_', '.', ' @latest  ']
[' ', '-', '#', ' ', '.', ' ', '@', ' ', ' ']
['-#', 'Python_3.13', '@latest']
```



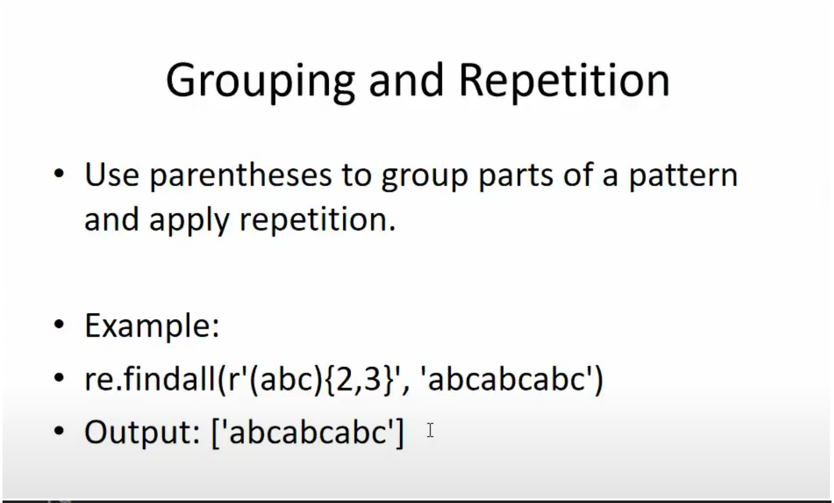
```python

#^
#$
#*  -- zero or more occurances
#+  -- one or more occurances

#\d  --  digits  "hello@123"
#\D  -- non digit characters

#\w  -- word character
#\W -- non word character

#\s  -- space character
#\S  -- non space characters

```



Combining Character Classes

- Combine different character classes in a single pattern.

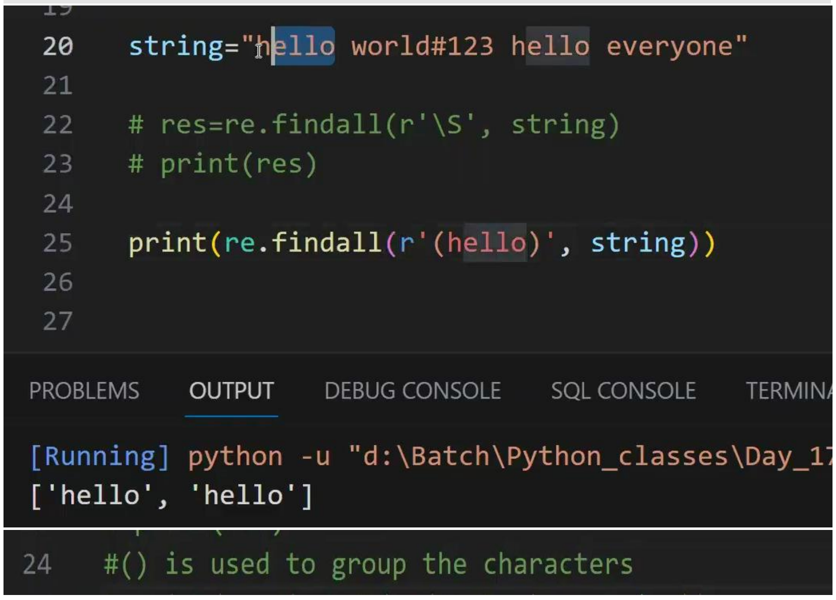
- Example:
- re.findall(r'\d\w\s', '3a ')
- Output: ['3a ']



Grouping and Repetition

- Use parentheses to group parts of a pattern and apply repetition.

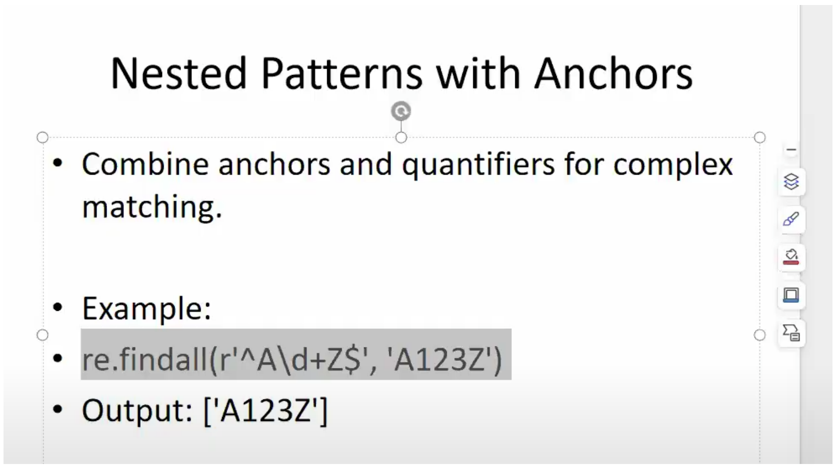
- Example:
- re.findall(r'(abc){2,3}', 'abcabcabc')
- Output: ['abcabcabc']



```
string="hello world#123 hello everyone"

# res=re.findall(r'\S', string)
# print(res)

print(re.findall(r'(hello)', string))
```

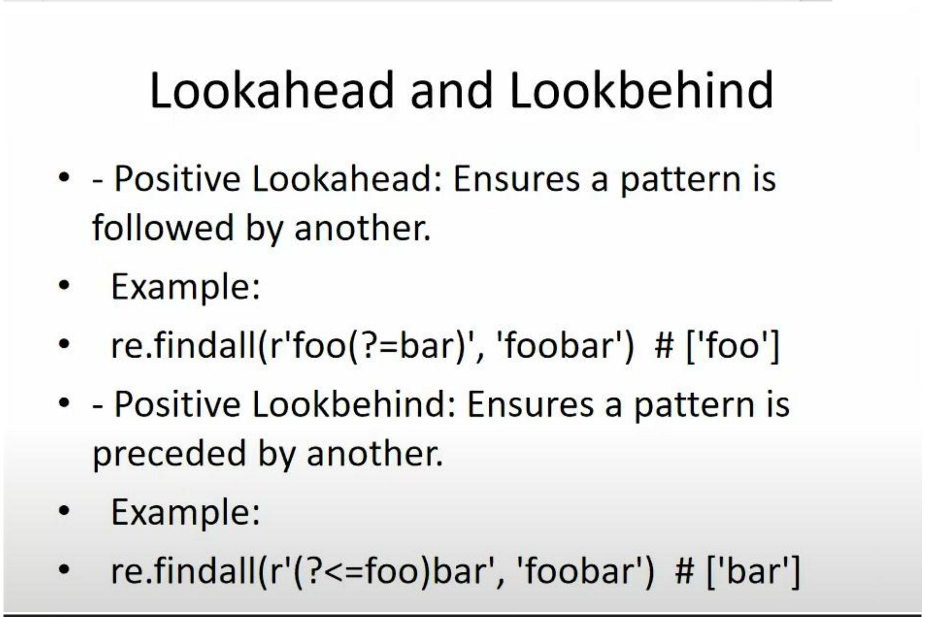
```
[Running] python -u "d:\Batch\Python_classes\Day_1
['hello', 'hello']
```

```
#() is used to group the characters
```



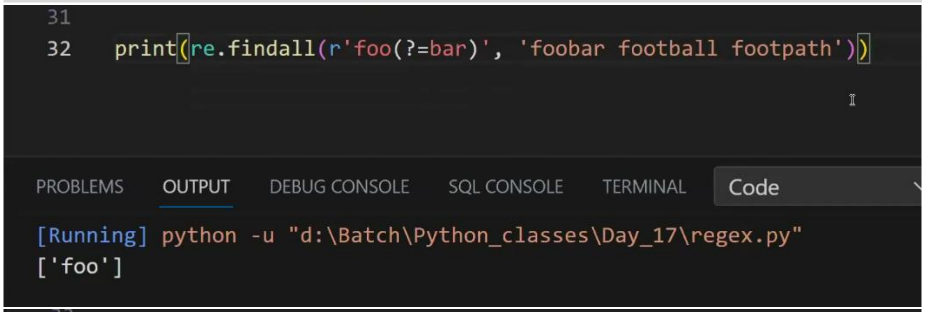
Nested Patterns with Anchors

- Combine anchors and quantifiers for complex matching.
- Example:
- re.findall(r'^A\d+Z$', 'A123Z')
- Output: ['A123Z']



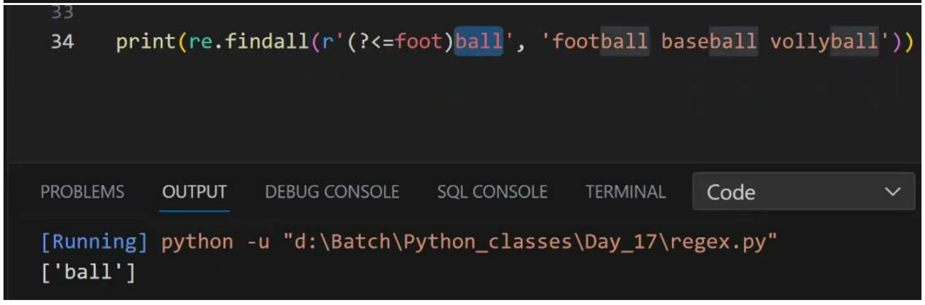
Lookahead and Lookbehind

- - Positive Lookahead: Ensures a pattern is followed by another.
- Example:
- re.findall(r'foo(?=bar)', 'foobar') # ['foo']
- - Positive Lookbehind: Ensures a pattern is preceded by another.
- Example:
- re.findall(r'(?<=foo)bar', 'foobar') # ['bar']



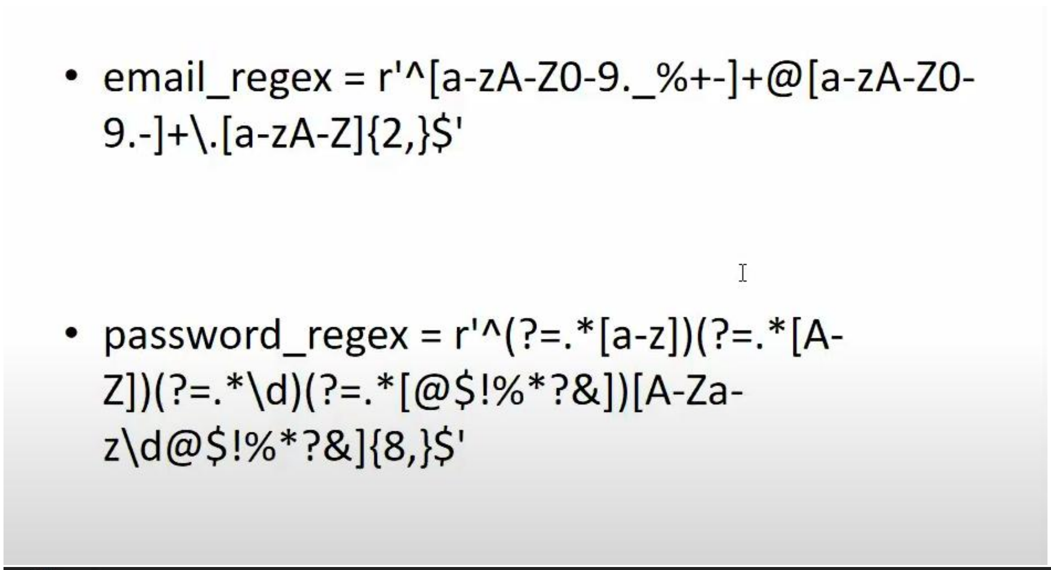
```
print(re.findall(r'foo(?=bar)', 'foobar football footpath'))
```

```
[Running] python -u "d:\Batch\Python_classes\Day_17\regex.py"
['foo']
```



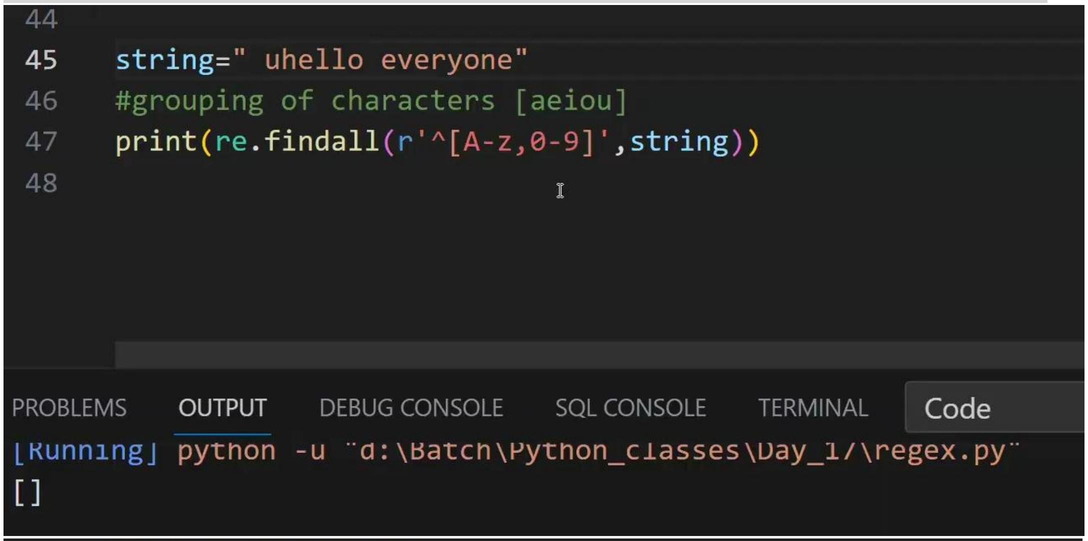
```
print(re.findall(r'(?<=foot)ball', 'football baseball vollyball'))
```

```
[Running] python -u "d:\Batch\Python_classes\Day_17\regex.py"
['ball']
```



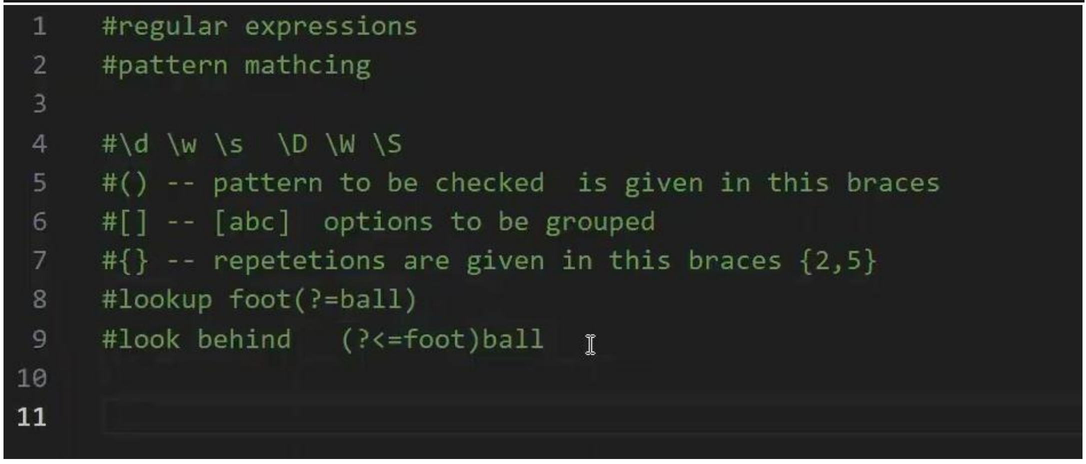
- email_regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'

- password_regex = r'^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$'



```python

string=" uhello everyone"
#grouping of characters [aeiou]
print(re.findall(r'^[A-z,0-9]',string))

```

```
[Running] python -u "d:\Batch\Python_classes\Day_17\regex.py"
[]
```



```python
#regular expressions
#pattern mathcing

#\d \w \s  \D \W \S
#() -- pattern to be checked  is given in this braces
#[] -- [abc]  options to be grouped
#{} -- repetetions are given in this braces {2,5}
#lookup foot(?=ball)
#look behind   (?<=foot)ball


```