

1.### Regular Expression

- Pattern Matching
- Patterns(re) package
- Cap Symbol is used to represent the start of re
- Dollar Symbol is used to represent the end of re
- [0-9] -->any digit matching
- ^[0-9]{2}\$ - Two digit matching
- ^[0-9]{5}\$ - Five digit Matching

2### Regular Expression for characters

- [a-z] -->Any lower case characters
- [A-Z] -->Any upper case characters
- ^[a-z]{5}\$ -->It accept 5 lower case characters
- ^[a-zA-Z]{8}\$ --> Accept 8 characters can be anything lower and upper
- ^[a-zA-Z0-9]{8}\$ -->Accept 8 characters can be anything lower.upper

3.# function to test two digit number matching

```
import re
def twodigitmatching(n):
    pattern = '^[0-9]{2}$'
    n = str(n)
    if re.match(pattern,n):
        return True
    return False
print(twodigitmatching(12)) #true
print(twodigitmatching(123)) #false
```

4..#Function to define to test username having 9 characters

Upper and lower case

```
def testusername(s):
    pattern = '^[a-zA-Z]{9}$'

    if re.match(pattern,s):
        return True
    return False
print(testusername('GiTaMHyDE')) #true
Pr
```

```
int(testusername('gitam')) #false
```

5.Regular expression to match the Indian Mobile Number

- 10 Digits
- (First digit will be [6-9]) and remaining 9 digits will be [0-9]
- Example:- 9988774455
- Re-^[6-9][0-9]{9}
- Example:- +919988774455
- Re- ^[+][9][1][6-9][0-9]{9}

```
6.def phonenumbervalidation(phone):
```

```
    pattern = '^[0][6-9][0-9]{9}$|^[0][6-9][0-9]{9}$|^[+][9][1][6-9][0-9]{9}$'
```

```
    phone = str(phone)
```

```
    if re.match(pattern,phone):
```

```
        return True
```

```
    return False
```

```
print(phonenumbervalidation('+919988776655')) #true
```

7.- ##### Regular Expression to validate the roll number

- EXAMPLE: 1521A0501

- Example: 1521A0109

- Example: 1521A0499

- ##### Regular Expression to validate the password

- Parameters: Len min of 6 characters and Max of 15 characters

- Accept Lower case, Upper case, Digits spl char(@,#,!)

8.Email Id validation using Regular Expression

- Example :- [Username@DomianName.extension](#)

Username :-

- Length will be [6-15]

- No Spls characters apart from Underscore(_)

- Should not begin and ends with Underscore(_)

- - Characters set : All digits and lower case

DomainName :-

- Length will be [3-18]

- No Spls characters

- - Character Set : All digits and lower case

```

9.def emailValidation(email):
    pattern = '^'[0-9a-z]0-9a-z_]{5,14}[@][a-z0-9]{2,18}[.][a-z]{2,4}$'
    if re.match(pattern,email):
        return True
    return False
emailValidation('ayyagaribhignya@gmail.com')

```

10. Python Turtle

- Turtle Graphics

```

11.# Step1 : Make all the turtle package to be imported
import turtle
# Turtle method creates and returns a new object
a1 = turtle.Turtle()
# forward() method moves 100 pixels
turtle.forward(250)
# We are done
turtle.done()

```

```

12.#Line draw in reverse direction
import turtle as tt
a1 = tt.Turtle()
tt.backward(100)
tt.done()

```

```

13.# Draw square
import turtle as tt
a1 = tt.Turtle
a1.forward(150)
a1.right(90)
a1.forward(150)
a1.right(90)
a1.forward(150)
a1.right(90)
a1.forward(150)
a1.right(90)
tt.done()

```

```

14.# Draw square
import turtle as tt
a1 = tt.Turtle()

```

```
a1.backward(150)
a1.left(90)
a1.backward(150)
a1.left(90)
a1.backward(150)
a1.left(90)
a1.backward(150)
a1.left(90)
tt.done()
```

```
15.# loop statement
import turtle as t
aa = t.Turtle()
for i in range(4):
    aa.backward(150)
    aa.left(90)
t.done()
```

```
16.# Star
import turtle as t
a1 = t.Turtle()
for i in range(40):
    a1.forward(50)
    a1.right(144)
t.done()
```

```
17.# Spiraling Star
import turtle as t
a1 = t.Turtle()
a1.pencolor('blue')
for i in range(40):
    a1.forward(i * 10)
    a1.right(144)
t.done()
```

18.# square spiral help of turtle

```
import turtle as t
al = t.Turtle()
al.pencolor('purple')
for i in range (250):
    al.forward(i)
    al.left(91)
t.done()
```

19.# hexagon

```
from turtle import *
colors = ['blue','green','yellow','orange','purple','red']
for x in range(360):
    pencolor(colors[x%6])
    width(x/100 + 1)
    forward(x)
    left(59)
```

20.#goto function

```
from turtle import *
goto(50,50)
goto(-50,50)
goto(100,-50)
goto(-50,-50)
```

21.# setheading(heading)

will change the current direction to the heading angle

```
from turtle import *
colors = ['blue','red','purple','orange','green','yellow']
for angle in range(0,360,15):
    pencolor(colors[angle%6])
    setheading(angle)
    forward(100)
    write(str(angle)+ 'o')
    backward(100)
```

22.#undo() function will undo the turtle lst action

```
from turtle import *
pencolor('blue')
for i in range (15):
    forward(100)
```

```
    left(90)
    forward(10)
    left(90)
    forward(100)
    right(90)
    forward(10)
    right(90)
pencolor('red')
for i in range(90):
    undo()
```

```
23.# Square
from turtle import *
pencolor('pink')
for i in range (15):
    forward(100)
    left(90)
    forward(100)
    left(90)
    pencolor('pink')
for i in range(90):
    undo()
```

```
24.# rectangle
from turtle import *
pencolor('pink')
for i in range (15):
    forward(100)
    left(90)
    forward(50)
    left(90)
    pencolor('pink')
for i in range(90):
    undo()
```

```
25.from turtle import *
colors = ('red','orange','yellow','seagreen4','orchid4','royalblue')
reset()
up()
goto(-320,-195)
width(90)
for pcolor in colors:
    color(pcolor)
    down()
    forward(750)
    up()
```

```
        backward(750)
        left(90)
        forward(100)
        right(90)
width(55)
color('white')
goto(0,-170)
down()
```

```
circle(170)
left(90)
forward(340)
up()
left(180)
forward(170)
up()
backward(170)
left(90)
down()
forward(170)
up()
goto(0,300)
```

```
26.from turtle import *
colors = ('red','orange','yellow','seagreen4','orchid4','royalblue')
reset()
up()
goto(-320,-195)
width(90)
for pcolor in colors:
    color(pcolor)
    down()
    forward(750)
    up()
    backward(750)
    left(90)
    forward(100)
    right(90)
width(55)
color('white')
goto(0,-170)
down()

left(90)
forward(300)
left(90)
```

```
forward(300)
left(90)
forward(300)
left(90)
forward(300)
left(10)
goto(0,300)
```