

#DAY

4

DAY 3 OF 200 DAY'S PYTHON CHALLENGE



INPUT FUNCTION

The `input()` function allows user input.

The `input()` function in Python is used to get input from the user. It takes a prompt as an argument and displays it to the user. The user can then type in their response and press Enter. The `input()` function then returns the user's response as a string.



Day 4.py

```
marks = input('Enter a input')
type(marks)
-- str
marks = int(input('Enter a number'))
type(marks) # Type conversion
-- int
```

CONDITIONAL STATEMENT

Conditional statements in Python allow you to control the flow of your program based on certain conditions. There are three types of conditional statements in Python:

- if statement: The if statement is used to execute a block of code if a certain condition is true. The syntax for an if statement is:



Day 4.py

```
marks = int(input('Enter a marks'))  
if marks >= 80:  
    print("I'm very happy!")  
print( marks)
```

CONDITIONAL STATEMENT

- **else statement:** The else statement is used to execute a block of code if the condition in the if statement is false. The syntax for an else statement is:



Day 4.py

```
marks = int(input('Enter a marks'))  
if marks >= 80:  
    print("I'm very happy!")  
else:  
    print("I'm not happy!")
```

CONDITIONAL STATEMENT

- **elif statement:** The elif statement is used to check for multiple conditions. The syntax for an elif statement is:



Day 4.py

```
marks = int(input('Enter a marks\n: \n'))
if marks >= 80:
    print("I'm very happy!")
elif marks >= 60:
    print("I'm happy!")
elif marks >= 50:
    print("I'm good!")
else:
    print("I'm not happy!")
```

LOOP STATEMENT

Loop statements can be used to perform a variety of tasks, such as:

- Iterating over a list of items
- Repeating a task until a certain condition is met
- Processing data in a batch



Day 4.py

```
l = [1,2,3,4,5]
for i in l :
    print(i , type(i))

l1 = ["sudh" , "kumar" , "krish" , "naik"]
for i in l1 :
    print(i)
```

RANGE FUNCTION

The `range()` function in Python is a built-in function that creates a sequence of numbers. It can be used to loop through a block of code a specified number of times. The `range()` function takes three arguments:

- stop: (required): The end of the range, not inclusive.
- start: (optional): The start of the range, defaults to 0.
- step: (optional): The increment between the numbers in the range, defaults to 1.



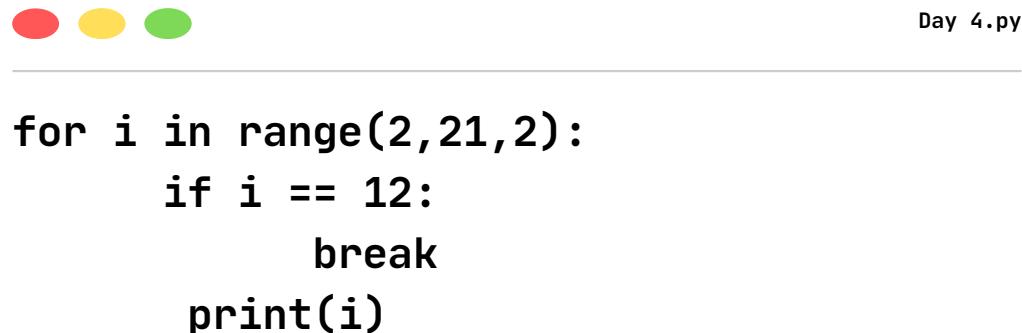
Day 4.py

```
l = [1,2,3,4,5]
for i in l :
    print(i , type(i))
```

```
l1 = ["sudh" , "kumar" , "krish" , "naik"]
for i in l1 :
    print(i)
```

BREAK KEYWORD

break allows you to exit a loop when an external condition is met.

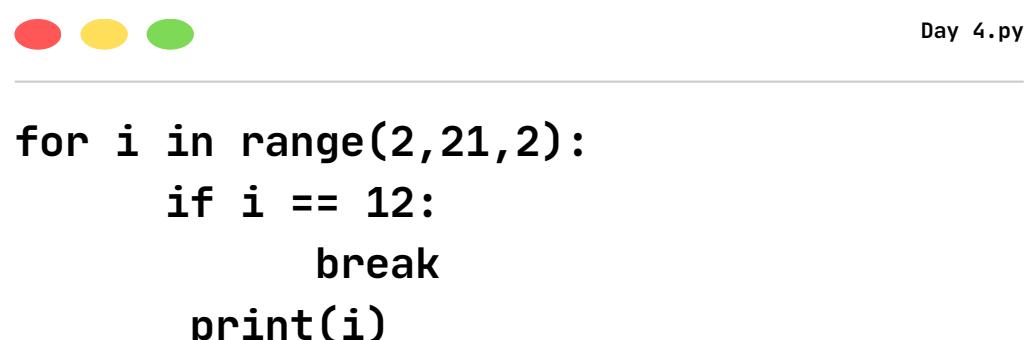


Day 4.py

```
for i in range(2,21,2):
    if i == 12:
        break
    print(i)
```

COUNTINUE KEYWORD

use the keyword continue to end the current iteration of a for loop.

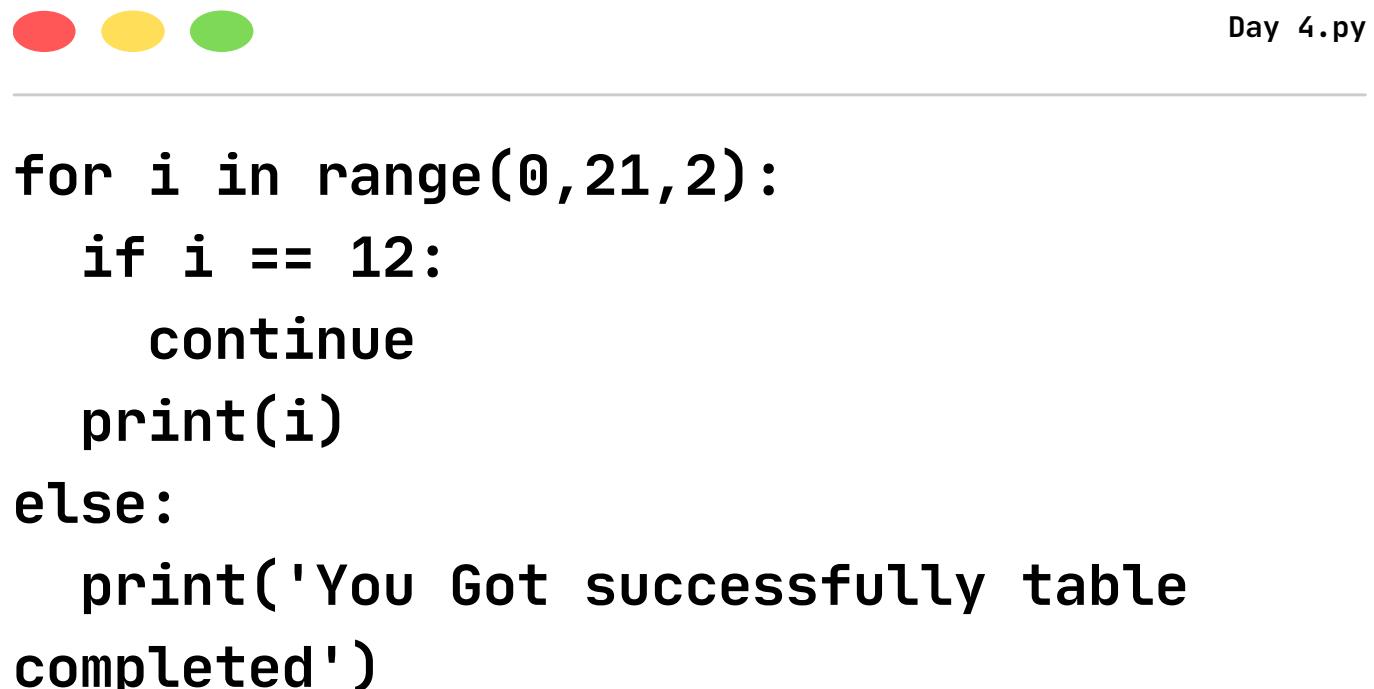


Day 4.py

```
for i in range(2,21,2):
    if i == 12:
        break
    print(i)
```

FOR-ELSE STATEMENT

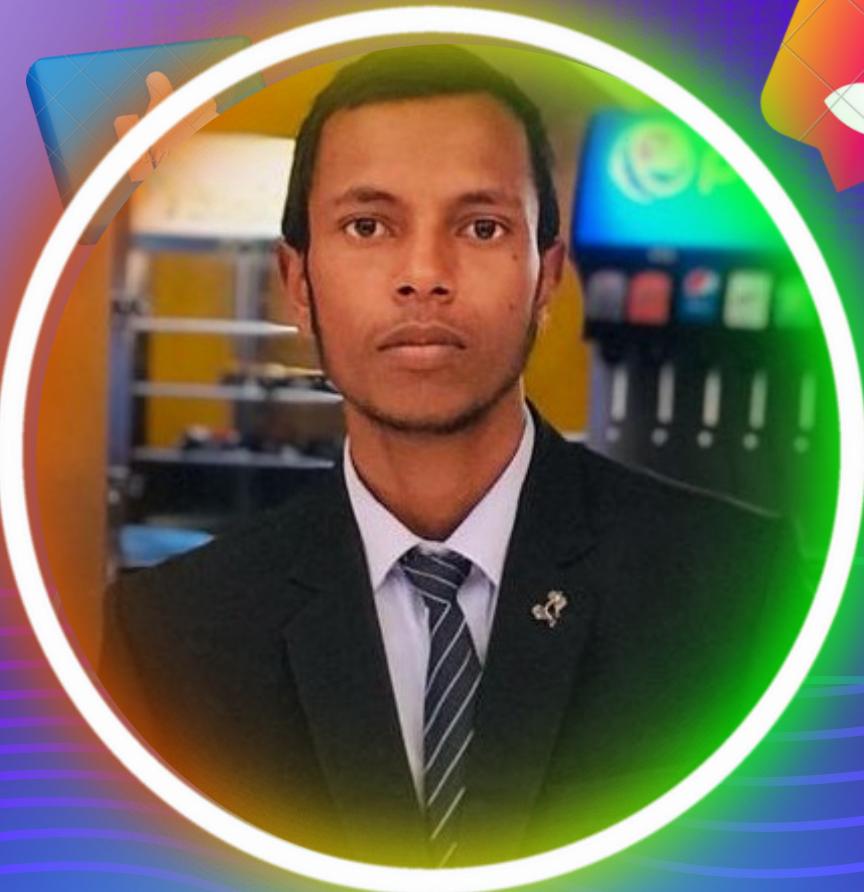
The else keyword in a for loop specifies a block of code to be executed when the loop is finished:



Day 4.py

```
for i in range(0,21,2):
    if i == 12:
        continue
    print(i)
else:
    print('You Got successfully table completed')
```

FOLLOW FOR MORE



FOLLOW ME

