



ورودی : Input

خواندن رشته از ورودی و چاپ :

```
# Input: Get an entry via command line
# Arguments: (text) [optional]
entry = input("Enter a number: ") # Prompt the user to enter a number

# Print the input value
print(entry) # Display the entered value in the console
```

شرط : if

```
# Conditional structure in Python

# If statement: checks if the condition is true (1 evaluates to True)
if 1:
    print("Here is 1")
# Elif statement: checks if the previous condition was false and this condition is true (0
evaluates to False)
elif 0:
    print("Here is 2")
# Else statement: executes if all previous conditions are false
else:
    print("Here is 3")
```

شرط اصلی : if

و یا اگر این چنین بود : elif یا else

اگر این چنین نبود : else





تمرین ۱ : برنامه ای بنویسید که دو عدد از ورودی خوانده و عدد بزرگتر را در خروجی چاپ کند .

```
## Practice 1 :  
# Write a program that reads two numbers from input and prints the larger number.  
  
# Input: Get two numbers from user input  
number1 = float(input("Enter the first number: "))  
number2 = float(input("Enter the second number: "))  
  
# Compare the two numbers and print the larger one  
if number1 > number2:  
    print("The larger number is:", number1)  
elif number2 > number1:  
    print("The larger number is:", number2)  
else:  
    print("Both numbers are equal.")
```

تمرین ۲ : برنامه ای بنویسید که از ورودی یک عدد بگیرد و مشخص کند که زوج است یا فرد .

```
## Practice 2 :  
# Write a program that takes an input number and determines if it is even or odd.  
  
# Input: Get a number from user input  
number = int(input("Enter a number: "))  
  
# Check if the number is even or odd  
if number % 2 == 0:  
    print("The number", number, "is even.")  
else:  
    print("The number", number, "is odd.")
```

* در پایتون عملگر (%) باقی مانده دو عدد را بدست می آورد . *



حلقه : for (loop)

* این حلقه زمانی استفاده میشود که **تعداد تکرار مشخص باشد** و **به شرطی بستگی ندارد** *

```
# Loop with start, stop, and step values
for i in range(start, stop, step):
    # do something

# Loop with start and stop values (default step is 1)
for i in range(start, stop):
    # do something

# Loop with stop value (default start is 0 and step is 1)
for i in range(stop):
    # do something
```

stop : مقدار پایان است و نوشتنش اجبار است و پیش فرضی ندارد :

start : مقدار شروع است و به صورت پیش فرض روی صفر است :

step : گام شمارش است و به صورت پیش فرض روی یک است :

تمرین ۳ : برنامه ای بنویسید که اعداد زوج دو رقمی را چاپ کند .

```
## Practice 3 :
# Program to print all two-digit even numbers

# Loop through the range of two-digit numbers
for number in range(10, 100, 2):
    print(number)
```

تمرین ۴ : برنامه ای بنویسید که مجموع اعداد یک تا ده را محاسبه کند .

```
## Practice 4 :
# Program to calculate the sum of numbers from 1 to 10
# Initialize sum to zero
total = 0

# Loop through numbers 1 to 10 and add them to total
for num in range(1, 11):
    total += num
    # OR total = total + num
# Print the result
print("Sum of numbers from 1 to 10 is:", total)
```

+= : یعنی اضافه کن و برابر قرار بده





تمرین ۵ : برنامه ای بنویسید که مجموع اعداد ۱ تا n را محاسبه کند .

```
## Practice 5 :  
# Program to calculate the sum of numbers from 1 to n  
  
# Input the value of n  
n = 10 # Replace with any value of n you want to calculate  
  
# Initialize sum to zero  
total = 0  
  
# Loop through numbers from 1 to n and add them to total  
for i in range(1, n + 1):  
    total += i  
  
# Print the result  
print(f"The sum of numbers from 1 to {n} is:", total)
```

* وقتی از f در اول رشته **print** استفاده میشود ، میتوان یک داده را در رشته چاپ شده جایگذاری کرد مثل {n} *

تمرین ۶ : برنامه ای بنویسید که ۱۰ عدد از ورودی خوانده سپس بزرگترین عدد وارد شده را در خروجی چاپ کند .

```
## Practice 6 :  
# Program to find the maximum of 10 numbers entered by the user  
  
# Initialize max to 0 (assuming all numbers are positive)  
max = 0  
  
# Loop to read 10 numbers from the user  
for i in range(0, 10):  
    number = int(input("Enter number: "))  
    if max < number: # Check if current number is greater than max  
        max = number # Update max if current number is greater  
  
# Print the maximum number  
print(max)
```





تمرین ۷ : برنامه ای بنویسید که یک عدد از ورودی بگیرد و مقسوم علیه های آن را در خروجی چاپ کند.

```
## Practice 7 :
# Program to print the factors of a number entered by the user

# Take input from the user
number = int(input("Enter a number: "))

# Print message if the number is less than or equal to zero
if number <= 0:
    print("Please enter a positive integer.")
else:
    print(f"The factors of {number} are:")

    # Find and print all factors of the number
    for i in range(1, number + 1):
        if number % i == 0:
            print(i)
```

حلقه : while (loop)

* این حلقه زمانی استفاده میشود که تعداد تکرار نامشخص باشد و به شرطی بستگی دارد *

```
# Loop
while condition:
    # do something

# Loop
while True:
    print('infinity')

# This while loop will never execute because the condition is False
while False:
    print('Not run')
```

* این حلقه بی پایان است و یا تا زمانی که یک شرط درست باشد انجام میشود و یا از دستور break استفاده شود *





تمرین ۸ : برنامه ای بنویسید که تعداد نامحدودی از عدد را از ورودی خوانده حاصل جمع آنها را در خروجی چاپ کند. (شرط پایان : ورود عدد منفی باشد)

```
## Practice 8 :
# Program to continuously sum numbers entered by the user until a negative number is entered

total = 0
number = 0

while number >= 0:
    number = int(input("Enter a number (negative to exit): "))
    if number >= 0:
        total += number

print(f"The sum of the numbers entered is: {total}")
```

تمرین ۹ : برنامه ای بنویسید که a و b و c را از ورودی خوانده و عبارت را محاسبه کند . عبارت : $\frac{a+b}{c}$

```
## Practice 9 :
# Write a program to read a, b, and c from input and calculate the following expression. ((a+b)/c)

# Input values for variables a, b, and c
a = int(input("Enter a number: ")) # Get value for a
b = int(input("Enter b number: ")) # Get value for b
c = int(input("Enter c number: ")) # Get value for c

# Initialize variables for factorial calculations
d, e, f = 1, 1, 1

# Calculate factorial of a
for i in range(1, a + 1):
    d = d * i

# Calculate factorial of b
for i in range(1, b + 1):
    e = e * i

# Calculate factorial of c
for i in range(1, c + 1):
    f = f * i

# Calculate and print the result of (a+b)/c
result = (a + b) / c
print(f"The result of ({a} + {b}) / {c} is: {result}")
```





تابع : function (def)

انواع توابع : ۱- توابع داخلی (Built-in Functions) ۲- بازگشتی (Recursive Functions)

```
# General structure of a function
def function_name(parameter1, parameter2, ...):
    # Function body (code block)
    # Perform operations, calculations, or logic
    return result # Optional: Return statement

# Example of a built-in function
def do_something(param1, param2):
    print(f"Parameters: {param1} and {param2}")

# Example of a recursive function: calculating factorial
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```

۱- توابع داخلی (Built-in Functions): این نوع توابع فقط دارای عملیات می باشند و هیچگونه مقدار برگشتی (return value) ندارند.

۲- توابع بازگشتی (Recursive Functions): این نوع توابع پس از انجام عملیات، یک مقدار خاص را برمی گردانند (بازگشتی می باشند).





```
## Practice 10 :
# Program to calculate (!a + !b) / !c

# Function to calculate factorial
def factorial(n):
    if n == 0:
        return 1
    else:
        result = 1
        for i in range(1, n + 1):
            result *= i
        return result

# Input values for !a, !b, and !c
a = int(input("Enter a value for !a: "))
b = int(input("Enter a value for !b: "))
c = int(input("Enter a value for !c: "))

# Calculate !a, !b, and !c using factorial function
not_a = factorial(a)
not_b = factorial(b)
not_c = factorial(c)

# Calculate (!a + !b) / !c
result = (not_a + not_b) / not_c

# Print the result
print(f"The result of (!{a} + !{b}) / !{c} is: {result}")
```

* نکته : هیچ وقت از **else** در شروط مربوط به **حلقه** ها **استفاده نکنید** زیرا باعث ایجاد **نتیجه غلط** می شود *

مثلا : در اینجا کدی برای مشخص کردن اول بودن عدد ارائه شده ، بخاطر وجود شرط مثبت در حلقه ، کد بهم میریزد و جواب غلط میدهد .

(کد اشتباه است)

```
def is_prime(number):
    if number > 1:
        for i in range(2, number // 2 + 1):
            if number % i == 0:
                return False
            else:
                return True
    return True
```





تمرین ۱۱ : برنامه ای بنویسید که ۱۰ را از ورودی خوانده ، تعداد اعداد اول را یافته و در خروجی چاپ کند .

```
## Practice 11 :  
# Prime Number Counter  
  
def is_prime(number):  
    if number > 1:  
        for i in range(2, number // 2 + 1):  
            if number % i == 0:  
                return False  
        return True  
  
count = 0 # Initialize counter for prime numbers  
for i in range(10): # Loop 10 times  
    number = int(input("Enter a number: ")) # Input a number from user  
    if is_prime(number): # Check if the number is prime  
        count += 1 # Increment count if prime  
  
print(count) # Output the count of prime numbers entered
```

تمرین ۱۲ : برنامه ای بنویسید که اعداد اول بین ۱۰ تا ۱۰۰۰ را چاپ کند .

```
### Practice 12 :  
# Prime Number 10 to 1000  
  
def is_prime(number):  
    if number > 1:  
        for i in range(2, number // 2 + 1):  
            if number % i == 0:  
                return False  
        return True  
  
# Print prime numbers between 10 and 1000  
print("Prime numbers between 10 and 1000:")  
for num in range(10, 1001):  
    if is_prime(num):  
        print(num)
```





تمرین ۱۳ : برنامه ای بنویسید که عدد از ۱ تا ورودی دریافتی را با فاکتوریل جمع کند .

$$1! + 2! + 3! + \dots + n! = result$$

Practice 13 :

Program to calculate and print the sum of factorials from 1 to num

Function to calculate factorial

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        result = 1
```

```
        for i in range(1, n + 1):
```

```
            result *= i
```

```
        return result
```

```
total = 0
```

```
number = int(input("Enter a number: ")) # Input the number for calculation
```

```
for i in range(1, number + 1):
```

```
    total += factorial(i)
```

```
print(f"The sum of factorials from 1 to {number} is: {total}")
```

