

[]: WEEK-1

[]: Write a python program to check whether a number is even or odd.

[]: Algorithm :

- 1) Start
- 2) Read a number n
- 3) If $n \% 2 == 0$, print Even
- 4) Else print Odd
- 5) Stop

```
[1]: n = int(input("Enter a number: "))

if n % 2 == 0:
    print("Even")
else:
    print("Odd")
```

Enter a number: 6
Even

[]: write a python program to check whether a given is positive, negative, or zero.

[]: Algorithm :

- 1) Read number n
- 2) If $n > 0 \rightarrow$ Positive
- 3) If $n < 0 \rightarrow$ Negative
- 4) Else \rightarrow Zero

```
[2]: n = int(input("Enter a number: "))

if n > 0:
    print("Positive")
elif n < 0:
    print("Negative")
else:
    print("Zero")
```

Enter a number: -9
Negative

[]: write a python program to find the largest among three numbers.

[]: Algorithm ;

- 1) Read a, b, c
- 2) Compare all three
- 3) Print largest

```
[3]: a = int(input("Enter a: "))
b = int(input("Enter b: "))
c = int(input("Enter c: "))

if a >= b and a >= c:
    print("Largest:", a)
elif b >= a and b >= c:
    print("Largest:", b)
else:
    print("Largest:", c)
```

Enter a: 5
Enter b: 8
Enter c: 1
Largest: 8

[]: write a python program to check whether a given is a prime number.

[]: Algorithm ;

- 1) Start
- 2) Read number n
- 3) If $n \leq 1$, Not Prime
- 4) Check divisibility from 2 to n-1
- 5) If divisible \rightarrow Not Prime
- 6) Else \rightarrow Prime

```
[4]: n = int(input("Enter number: "))
if n <= 1:
    print("Not Prime")
else:
    for i in range(2, n):
        if n % i == 0:
            print("Not Prime")
            break
    else:
        print("Prime")
```

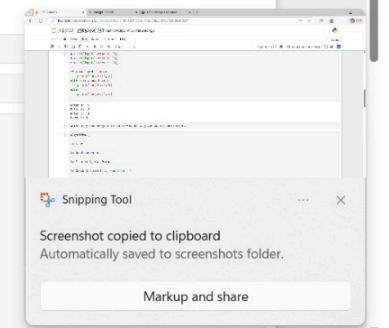
Enter number: 7
Prime

[]: WEEK-2

[]: Write a python program to find the factorial of a number.

[]: Algorithm :

- 1) Start
- 2) Read n
- 5) Initialize fact = 1
- 6) Multiply numbers from 1 to n
- 7) Print factorial



```
[5]: n = int(input("Enter number: "))
fact = 1
for i in range(1, n+1):
    fact *= i
print("Factorial:", fact)
```

Enter number: 5
Factorial: 120

[]: write a python program to check whether a number is a palindrome.

[]: Algorithm :

- 1) Start
- 2) Read number as string
- 3) Reverse the string
- 4) Compare original and reversed
- 5) Print result
- 6) Stop

```
[6]: n = input("Enter number: ")
if n == n[::-1]:
    print("Palindrome")
else:
    print("Not Palindrome")
```

Enter number: 121
Palindrome

[]: Write a python program to check whether a given string is a palindrome.

[]: Algorithm :

- 1) Start
- 2) Read string
- 3) Reverse string
- 4) Compare
- 5) Print result
- 6) Stop

[7]: s = input("Enter string: ")

+ % ⌂ ⌂ ▶ ■ ⌂ Code

JupyterLab ⓘ Python [conda env:base] * ○ ⌂

```
[7]: s = input("Enter string: ")
if s == s[::-1]:
    print("Palindrome")
else:
    print("Not Palindrome")
```

Enter string: stoic
Not Palindrome

[]: WEEK-3

[]: Write a python program to **print** the fibonacci series up to N terms.

[]: Algorithm :

- 1) Start
- 2) Read n
- 3) Initialize a=0, b=1
- 4) Print Fibonacci series up to n terms
- 5) Stop

[8]: n = int(input("Enter terms: "))

+ % ⌂ ⌂ ▶ ■ ⌂ Code

JupyterLab ⓘ Python [conda env:base] * ○ ⌂

```
[8]: n = int(input("Enter terms: "))
a, b = 0, 1
for i in range(n):
    print(a, end=" ")
    a, b = b, a + b
```

Enter terms: 123

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 352
4578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073 4807526976 77
78742049 12586269025 20365011074 32951280099 53316291173 86267571272 139583862445 225851433717 365435296162 591286729879 956722026041 1548008755920 25047
30781961 4052739537881 6557478319842 10610209857723 17167680177565 27777890035288 44945570212853 72723460248141 117669030460994 190392490709135 308061521
170129 498454011879264 806515533049393 1384969544928657 211148507798056 3416454622986707 5527939700884757 8944394323791464 14472334024676221 23416728348
467685 37889062373143908 61305790721611591 99194853894755497 160500643816367088 259695496911122585 420196140727489673 679891637638612258 1100087778366101
931 1779979416004714189 2880067194370816120 4660046610375530399 7540113804746346429 12200160415121876738 19740274219868223167 31940434634990099905 516807
08854858323072 83621143489848422977 135301852344708746049 218922995834555169025 354224848179261915075 573147844013817084161 927372692193078999176 1500520
536206896083277 2427893228399975082453 3928413764606871165730 6356306993006846248183 10284720757613717413913 16641027750620563662096 26925748508234281076
009 4356677625854844738105 70492524767089125814114 114059301825943970552219 184551825793033096366333 298611126818977066918552 483162952612010163284885 7
81774079430987230203437 1264937830249297933488322 2046711111473984623691759 3311648143516982017180081 5358359254998966460871840 8670007398507948658051921
14028366653498915298923761
```

[]:

[]:

Anaconda Toolbox

File Edit View Run Kernel Settings Help

+ % ⌂ ⌂ ▶ ■ ⌂ Code

JupyterLab ⓘ Python [conda env:base] * ○ ⌂

Anaconda

Your Toolbox for Python Projects

Anaconda Cloud

Create a New Project

Create a New Notebook

My Projects

Code Snippets

Manage Code Snippets

Environments

[]: Write a python program to find the **sum** of digits of a number.

[]: Algorithm :

- 1) Start
- 2) Read number
- 3) Extract digits using modulus
- 4) Add digits
- 5) Print sum
- 6) Stop

```
[9]: n = int(input("Enter number: "))
s = 0
while n > 0:
    s += n % 10
    n /= 10
print("Sum:", s)
```

Enter number: 345
Sum: 12

Anaconda Toolbox

File Edit View Run Kernel Settings Help

Anaconda
Your Toolbox for Python Projects

Anaconda Cloud

Create a New Project
Create a New Notebook
My Projects

Code Snippets

Manage Code Snippets

Environments

Create new Environment

[]: Write a python program to count vowels **and** consonants **in** a string.

[]: Algorithm :

- 1) Start
- 2) Read string
- 3) Check each character
- 4) Count vowels **and** consonants
- 5) Print result
- 6) Stop

[10]:
s = input("Enter string: ").lower()
v = c = 0
for ch in s:
 if ch.isalpha():
 if ch in "aeiou":
 v += 1
 else:
 c += 1
print("Vowels:", v)
print("Consonants:", c)

Anaconda Toolbox

File Edit View Run Kernel Settings Help

Anaconda
Your Toolbox for Python Projects

Anaconda Cloud

Create a New Project
Create a New Notebook
My Projects

Code Snippets

Manage Code Snippets

Environments

Create new Environment

Enter string: Jupiter
Vowels: 3
Consonants: 4

[]: WEEK-4

[]: Write a python program to reverse a string without using built-in functions.

Algorithm :

- 1) Start
- 2) Read string
- 3) Reverse using loop
- 4) Print result
- 5) Stop

[11]:
s = input("Enter string: ")
rev = ""
for ch in s:
 rev = ch + rev
print("Reversed:", rev)

Enter string: aba

Anaconda Toolbox

File Edit View Run Kernel Settings Help

Anaconda
Your Toolbox for Python Projects

Anaconda Cloud

Create a New Project
Create a New Notebook
My Projects

Code Snippets

Manage Code Snippets

Environments

Create new Environment

Anaconda AI Assistant
localhost:8888/projects

[]: Algorithm :

- 1) Start
- 2) Read two numbers
- 3) Read operator
- 4) Perform operation
- 5) Print result
- 6) Stop

[13]:
a = int(input("Enter a: "))
b = int(input("Enter b: "))
op = input("Enter operator: ")

if op == '+':
 print(a + b)
elif op == '-':
 print(a - b)
elif op == '*':
 print(a * b)
elif op == '/':
 print(a / b)
else:
 print("Invalid")

Enter a: 2
Enter b: 4
Enter operator: +

Anaconda Toolbox Trusted

Anaconda

Your Toolbox for Python Projects

Anaconda Cloud

Create a New Project

Create a New Notebook

My Projects

Code Snippets

Manage Code Snippets

Environments

Create new Environment

Enter string: aba
Reversed: aba

[]: Write a python program to count the occurrence of each character **in** a string.

[]: Algorithm :

- 1) Start
- 2) Read string
- 3) Count each character
- 4) Print frequency
- 5) Stop

[12]:

```
s = input("Enter string: ")  
for ch in set(s):  
    print(ch, ":", s.count(ch))
```

Enter string: cbc
b : 1
c : 2

[]: Write a python program to create a simple calculator using conditional statements.

Anaconda

Your Toolbox for Python Projects

Anaconda Cloud

Create a New Project

Create a New Notebook

My Projects

Code Snippets

Manage Code Snippets

Environments

Create new Environment

[]: Algorithm :

- 1) Start
- 2) Read two numbers
- 3) Read operator
- 4) Perform operation
- 5) Print result
- 6) Stop

[13]:

```
a = int(input("Enter a: "))  
b = int(input("Enter b: "))  
op = input("Enter operator: ")  
  
if op == '+':  
    print(a + b)  
elif op == '-':  
    print(a - b)  
elif op == '*':  
    print(a * b)  
elif op == '/':  
    print(a / b)  
else:  
    print("Invalid")
```

6 JupyterLab Python [conda env:base]

Anaconda

Your Toolbox for Python Projects

Anaconda Cloud

Create a New Project

Create a New Notebook

My Projects

Code Snippets

Manage Code Snippets

Environments

Create new Environment

Anaconda AI Assistant

[]: WEEK-5

[]: Write a python program to implement a menu-driven calculator using a loop (repeat until the user exists).

[]: Algorithm :

- 1) Start
- 2) Display menu
- 3) Read user choice
- 4) Perform selected operation
- 5) Repeat until exit option **is** chosen
- 6) Stop

[14]:

```
while True:  
    print("1.Add 2.Sub 3.Mul 4.Div 5.Exit")  
    ch = int(input("choice: "))  
    if ch == 5:  
        break  
    a = int(input("a: "))  
    b = int(input("b: "))  
  
    if ch == 1: print(a + b)  
    elif ch == 2: print(a - b)  
    elif ch == 3: print(a * b)
```

write a python program to generate a multiplication table for a given number (loop until the user stops).

[]: Algorithm:

- 1) Start
- 2) Read number
- 3) Use loop `from 1 to 10`
- 4) Multiply `and print result`
- 5) Stop

[15]: `n = int(input("Enter number: "))
for i in range(1, 11):
 print(n, "x", i, "=", n*i)`

```
Enter number: 7  
7 x 1 = 7  
7 x 2 = 14  
7 x 3 = 21  
7 x 4 = 28  
7 x 5 = 35  
7 x 6 = 42  
7 x 7 = 49
```

A screenshot of a web browser window titled "Untitled4". The address bar shows "localhost:8888/notebooks/Untitled4.ipynb?". The page content displays the multiplication table for 7, starting from 7 x 1 = 7 up to 7 x 10 = 70.

```
7 x 6 = 42  
7 x 7 = 49  
7 x 8 = 56  
7 x 9 = 63  
7 x 10 = 70
```

[]: Write a python program to `print` different patterns using loop concepts (e.g. star patterns, number patterns)

[]: Algorithm:

- 1) Start
- 2) Read number of rows
- 3) Use nested loops
- 4) Print pattern line by line
- 5) Stop

[16]: `for i in range(1, 6):
 print("*" * i)`

```
*
```



```
**
```



```
***
```



```
****
```



```
*****
```

Search - Microsoft Bing | Code examples execution | Home | Untitled4 | +

localhost:8888/notebooks/Untitled4.ipynb | Error Chat

FUNCTION-BASED QUESTIONS

```
[ ]: week-6
```

```
[ ]: Write a python function that takes a users name and prints a greeting message.
```

```
[17]: def greet(name):
        print("Hello", name)

greet("Devil")
Hello Devil
```

```
[ ]: Write a python functions that accepts two numbers and returns their sum.
```

```
[18]: def add(a, b):
        return a + b

print(add(10, 5))
15
```

```
[ ]: WEEK-7
```

```
[ ]: Write a python recursion
```

Search - Microsoft Bing | Home | Untitled5 | +

localhost:8889/notebooks/Untitled5.ipynb | Error Chat

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python [conda env:base]*

```
[ ]: WEEK-7
```

```
[ ]: Write a python recursive function to find the factorial of a number.
```

```
[ ]: Algorithm :
1) Start
2) Define recursive function
3) If n == 0, return 1
4) Else return n * factorial(n-1)
5) Call function
6) Stop
```

```
[19]: def fact(n):
        if n == 0:
            return 1
        return n * fact(n-1)

print(fact(5))
120
```

Search - Microsoft Bing Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb? File Edit View Run Kernel Settings Help JupyterLab Python [conda env:base] * Trusted

```
[ ]: Write a python lambda function to check whether a number is even.
```

```
[ ]: Algorithm:  
1) Start  
  
2) Define lambda function  
  
3) Check modulus condition  
  
4) Print result  
  
5) Stop
```

```
[20]: even = lambda x: x % 2 == 0  
print(even(6))  
True
```

```
[ ]: Write a python program to calculate factorial using recursion with input validation.
```

```
[ ]: Algorithm:  
1) Start  
  
2) Read input number  
  
3) Validate input  
  
4) If valid, calculate factorial using recursion
```

Search - Microsoft Bing Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb? File Edit View Run Kernel Settings Help JupyterLab Python [conda env:base] * Trusted

```
5) Print result  
6) Stop
```

```
[21]: def fact(n):  
    if n < 0:  
        return "Invalid"  
    if n == 0:  
        return 1  
    return n * fact(n-1)  
  
print(fact(5))  
120
```

```
[ ]: Project/Advanced Questions
```

```
[ ]: WEEK-8
```

```
[ ]: Write a python program to create a library book management system using functions.
```

```
[ ]: Algorithm :  
1) Start  
2) Initialize empty book list  
3) Create functions to add and display books
```

Search - Microsoft Bing Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb?

```
2) Initialize empty book list

3) Create functions to add and display books

4) Call functions as required

5) Stop
```

```
[22]: books = []

def add_book(book):
    books.append(book)

def show():
    print("Books:", books)

add_book("Python")
add_book("Java")
show()
```

```
Books: ['Python', 'Java']
```

```
[ ]: WEEK-9
```

```
[ ]: Write a python project to build a calculator using modular programming ( separate module for operation ).
```

Search - Microsoft Bing Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb?

```
[ ]: calculator_app/
|
+-- main.py
+-- modules/
|   +-- operations.py
+-- tests/
|   +-- test_operations.py
+-- docs/
|   +-- flowchart.png
|   +-- step_screenshot1.png
|   +-- step_screenshot2.png
```

```
[23]: def add(a,b): return a+b
def sub(a,b): return a-b

print(add(10,5))
print(sub(10,5))

15
5
```

```
[ ]: WEEK-10
```

```
[ ]: Write a python programming that applies modular programming principles and defines multiple reusable functions.
```

Search - Microsoft Bing Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb?

```
[ ]: week10_project/
|
├── main.py
└── utils/
    ├── __init__.py
    ├── input_validation.py
    ├── calculations.py
    └── display.py
└── tests/
    ├── test_input_validation.py
    ├── test_calculations.py
    └── test_display.py
└── docs/
    ├── flowchart.png
    └── steps.png
```

```
[24]: def square(n): return n*n
      def cube(n): return n*n*n

      print(square(4))
      print(cube(3))

16
27
```

Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb?

JupyterLab Python [conda env:base] *

```
[ ]: week11_project/
|
├── main.py
└── modules/
    ├── __init__.py
    ├── input_validation.py
    ├── operations.py
    └── logger.py
└── tests/
    ├── test_input_validation.py
    ├── test_operations.py
    └── test_logger.py
└── docs/
    ├── test_cases.txt
    └── debugging_notes.txt
```

```
[25]: def is_even(n):
      if not isinstance(n, int):
          return "Invalid"
      return n % 2 == 0

      print(is_even(4)) # Test 1
      print(is_even(7)) # Test 2
      print(is_even("a")) # Test 3

True
False
Invalid
```

```
[ ]: WEEK-12
```

Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb? JupyterLab Python [conda env:base]

```
[ ]: Write a python project for a user registration system with input validation, testing, and debugging documentation.

[ ]: week12_project/
|
|   app.py
|   core/
|     __init__.py
|     validation.py
|     user_manager.py
|     logger.py
|   database/
|     users.txt
|   tests/
|     test_validation.py
|     test_user_manager.py
|     test_logger.py
|   docs/
|     test_cases.txt
|     input_rules.txt
|     debugging_notes.txt

[26]: users = []

def register(username, password):
    if len(password) < 6:
        print("Weak password")
    else:
        users.append(username)
        print("Registered")
```

Home Untitled5

localhost:8889/notebooks/Untitled5.ipynb? JupyterLab Python [conda env:base]

```
[26]: users = []

def register(username, password):
    if len(password) < 6:
        print("Weak password")
    else:
        users.append(username)
        print("Registered")

register("admin", "123456")
Registered
```