

1. Display Hello World

```
print("Hello World")
```

2. Sum two numbers

```
a=float(input("Enter first number: ")) b=float(input("Enter second number: "))
print("Sum =", a+b)
```

3. Declare variables of different data types

```
x_int=10 x_float=3.14 x_str="MCA" x_bool=True print(type(x_int), type(x_float),
type(x_str), type(x_bool))
```

4. Declare variables and use id()

```
a=100 b="hello" c=[1,2,3] print("id(a):",id(a)) print("id(b):",id(b))
print("id(c):",id(c))
```

5. Take user input and display it

```
s=input("Enter something: ") print("You entered:", s)
```

6. Demonstrate all Python operators

```
a,b=10,3 print("Arithmetic:",a+b,a-b,a*b,a/b,a//b,a%b,a**b)
print("Comparison:",a==b,a!=b,a>b,a<b,a<=b) print("Logical:",(a>0 and b>0),(a>0 or
b<0),not(a==b)) print("Bitwise:",a&b,a|b,a^b,~a,a<<1,a>>1) a+=5
print("Assignment:",a)
```

7. Check positive/negative/zero

```
n=float(input("Enter number: ")) if n>0: print("Positive") elif n<0:
print("Negative") else: print("Zero")
```

8. Nested if for grading

```
marks=float(input("Enter marks: ")) if marks>=0 and marks<=100: if marks>=75:
grade="A" elif marks>=60: grade="B" elif marks>=50: grade="C" else: grade="F"
print("Grade:", grade) else: print("Invalid marks")
```

9. Largest of three numbers

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

10. Largest using max()

```
nums=[float(input("Enter 1: ")),float(input("Enter 2: ")),float(input("Enter 3: "))]
print("Largest is",max(nums))
```

11. Check even or odd

```
n=int(input("Enter integer: ")) if n%2==0: print("Even") else: print("Odd")
```

12. Check prime number

```
n=int(input("Enter integer: ")) if n<2: print("Not prime") else: for i in
range(2,int(n**0.5)+1): if n%i==0: print("Not prime") break else: print("Prime")
```

13. List insert, update, delete

```
lst=[1,2,3] lst.insert(1,1.5) lst[2]=99 lst.remove(3) print(lst)
```

14. Sum, max, min in list

```
lst=[int(x) for x in input("Enter numbers: ").split()]
print("Sum:",sum(lst),"Max:",max(lst),"Min:",min(lst))
```

15. Sort list ascending/descending

```
lst=[5,2,9,1] print("Ascending:",sorted(lst))
print("Descending:",sorted(lst,reverse=True))
```

16. Count even/odd in list

```
lst=[int(x) for x in input("Enter numbers: ").split()] even=sum(1 for x in lst if
x%2==0) print("Even:",even,"Odd:",len(lst)-even)
```

17. Linear search

```
lst=[int(x) for x in input("Enter list: ").split()] key=int(input("Enter key: ")) if
key in lst: print("Found at index",lst.index(key)) else: print("Not found")
```

18. Merge two lists

```
a=[1,2,3] b=[4,5] print(a+b)
```

19. Iterate list

```
lst=["apple","banana","cherry"] for item in lst: print(item)
```

20. Tuple example

```
t=(1,"hi",3.14) for i in t: print(i)
```

21. Sum tuple elements

```
t=(2,4,6,8) print("Sum:",sum(t))
```

22. Set operations

```
A={1,2,3,4} B={3,4,5} print("Union:",A|B) print("Intersection:",A&B)
print("Difference:",A-B)
```

23. String operations

```
s="HelloWorld" print(s[0:5],s+"!",s[::-1])
```

24. Palindrome check

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

25. Dictionary display

```
d={"name":"Amit","id":101} for k,v in d.items(): print(k,":",v)
```

26. Dict add, update, delete

```
d={} d["name"]="Amit" d["id"]=101 d["id"]=102 del d["name"] print(d)
```

27. Check key exists

```
d={"a":1} key=input("Key: ") print("Exists" if key in d else "Not exists")
```

28. Merge two dicts

```
d1={"a":1} d2={"b":2} print(**d1,**d2)
```

29. Dict keys, values, items

```
d={"x":10,"y":20} print(d.keys(),d.values(),d.items())
```

30. Function add two numbers

```
def add(a,b): return a+b print(add(3,4))
```

31. Average marks

```
def avg(m): return sum(m)/len(m) m=[float(x) for x in input("Marks: ").split()]
print(avg(m))
```

32. Fibonacci recursion

```
def fib(n): return n if n<=1 else fib(n-1)+fib(n-2) for i in range(int(input("n:
"))): print(fib(i),end=" ")
```

33. Perfect square check

```
import math def perfect_square(n): r=int(math.isqrt(n));return n if r*r==n else -1
print(perfect_square(16))
```

34. Simple calculator

```
def add(a,b):return a+b def sub(a,b):return a-b def mul(a,b):return a*b def
div(a,b):return a/b if b!=0 else "Div0" a=float(input("a: "))b=float(input("b:
"));op=input("op: ") print({"+":add,"-":sub,"*":mul,"/":div}.get(op,lambda
x,y:"Invalid")(a,b))
```

35. Return multiple values

```
def ops(a,b): return a+b,a-b,a*b,a/b a,b=8,2 print(ops(a,b))
```

36. Area of shapes

```
import math def circle(r):return math.pi*r*r def rect(l,w):return l*w def
tri(b,h):return 0.5*b*h print(circle(3),rect(4,5),tri(6,2))
```

37. Python keywords

```
import keyword print(keyword.kwlist)
```

38. Clear screen

```
import os os.system("cls" if os.name=="nt" else "clear")
```

39. sys module prompt (REPL only)

```
import sys sys.ps1=">>> " print("Prompt changed")
```

40. Random numbers

```
import random print(random.randint(1,100),random.random())
```

41. Current date/time

```
from datetime import datetime print(datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
```

42. User-defined module example

```
# mymath.py: def add(a,b):return a+b # main.py: import mymath print(mymath.add(2,3))
```

43. Simple package example

```
# mypkg/mod1.py def greet(): print("Hello") # main.py from mypkg import mod1  
mod1.greet()
```

44. From package import module

```
from mypkg import mod1 mod1.greet()
```

45. Read JSON file

```
import json with open("data.json") as f: print(json.load(f))
```

46. Read users.txt

```
users={} with open("users.txt") as f: for line in f: u,p=line.strip().split(",")  
users[u]=p print(users)
```

47. Write text file

```
with open("output.txt","w") as f:f.write("Hello file") print("Done")
```

48. School class objects

```
class School: def __init__(self,n,a): self.n=n;self.a=a def show(self):  
print(self.n,"-",self.a) s1=School("ABC","Delhi");s2=School("XYZ","Noida")  
s1.show();s2.show()
```

49. Exception handling file

```
try: with open("nofile.txt") as f:print(f.read()) except  
FileNotFoundError:print("File not found")
```

50. Divide by zero exception

```
try: a=float(input("a: "));b=float(input("b: "));print(a/b) except  
ZeroDivisionError:print("Cannot divide by zero")
```

51. Multiple except blocks

```
try: a=int(input("a: "));b=int(input("b: "));print(a//b) except  
ZeroDivisionError:print("Div0") except ValueError:print("Invalid")
```

52. try-except-else

```
try:v=int(input("Enter int: ")) except ValueError:print("Not int") else:print("You  
entered:",v)
```

53. try-except-finally

```
try: f=open("temp.txt");print(f.read()) except FileNotFoundError:print("No file")  
finally: try:f.close() except:pass print("Finally executed")
```

54. Raise custom exception

```
class MyError(Exception):pass def check(x): if x<0: raise MyError("Negative") return  
x try:check(-5) except MyError as e:print("Caught:",e)
```

55. Mean, median, mode

```
import statistics
nums=[float(x) for x in input("Enter numbers: ").split()]
print(statistics.mean(nums),statistics.median(nums))
try:print(statistics.mode(nums)) except:print("No unique mode")
```

56. Highest, lowest, avg marks

```
marks=[float(x) for x in input("Enter marks: ").split()]
print("High:",max(marks),"Low:",min(marks),"Avg:",sum(marks)/len(marks))
```

57. Read CSV student data

```
import csv
with open("students.csv") as f:
    for row in csv.DictReader(f):
        print(row)
```

58. Analyze sales data

```
sales=[float(x) for x in input("Sales: ").split()]
print("Total:",sum(sales),"Avg:",sum(sales)/len(sales),"Max:",max(sales))
```

59. Count char frequency

```
s=input("Enter text: ")
d={}
for c in s:
    d[c]=d.get(c,0)+1
print(d)
```

60. Count word frequency

```
p=input("Enter paragraph: ")
words=p.split()
d={}
for w in words:
    d[w.lower()]=d.get(w.lower(),0)+1
print(d)
```