1. Python Output

```
# Python is a case sensitive language
print('Hello World')
```

```
Hello World
```

```
print('salman khan')
```

```
salman khan
```

```
print(salman khan)
```

```
  File "<ipython-input-3-0713073d8d88>", line 1
    print(salman khan)
                 ^
SyntaxError: invalid syntax
```

```
print(7)
```

```
7
```

```
print(7.7)
```

```
7.7
```

```
print(True)
```

```
True
```

```
print('Hello',1,4.5,True)
```

```
Hello 1 4.5 True
```

```
print('Hello',1,4.5,True,sep='/')
```

```
Hello/1/4.5/True
```

```
print('hello')
print('world')
```

```
hello
world
```

```
print('hello',end='-')
print('world')
```

```
hello-world
```

2. Data Types

```
# Integer
print(8)
# 1*10^308
print(1e309)
```

```
8
inf
```

```
# Decimal/Float
print(8.55)
print(1.7e309)
```

```
8.55
inf
```

```
# Boolean
print(True)
print(False)
```

```
True
False
```

```
# Text/String
print('Hello World')
```

```
Hello World
```

```
# complex
print(5+6j)
```

```
(5+6j)
```

```
# List-> C-> Array
print([1,2,3,4,5])
```

```
[1, 2, 3, 4, 5]
```

```
# Tuple
print((1,2,3,4,5))
```

```
(1, 2, 3, 4, 5)
```

```
# Sets
print({1,2,3,4,5})
```

```
{1, 2, 3, 4, 5}
```

```
# Dictionary
print({'name':'Nitish','gender':'Male','weight':70})
```

```
{'name': 'Nitish', 'gender': 'Male', 'weight': 70}
```

```
# type
type([1,2,3])
```

```
list
```

3. Variables

```
# Static Vs Dynamic Typing
# Static Vs Dynamic Binding
# stylish declaration techniques
```

```
# C/C++
name = 'nitish'
print(name)

a = 5
b = 6

print(a + b)
```

```
nitish
11
```

```
# Dynamic Typing
a = 5
# Static Typing
int a = 5
```

```
# Dynamic Binding
a = 5
print(a)
a = 'nitish'
print(a)

# Static Binding
int a = 5
```

```
5
nitish
```

```
a = 1
b = 2
c = 3
print(a,b,c)
```

```
1 2 3
```

```
a,b,c = 1,2,3
print(a,b,c)
```

```
1 2 3
```

```
a=b=c= 5
print(a,b,c)
```

```
5 5 5
```

Comments

```
# this is a comment
# second line
a = 4
b = 6 # like this
# second comment
print(a+b)
```

```
10
```

4. Keywords & Identifiers

```
# Keywords

# Identifiers
# You can't start with a digit
name1 = 'Nitish'
print(name1)
# You can use special chars -> _
_ = 'ntiish'
print(_)
# identiers can not be keyword
```

```
Nitish
ntiish
```

Temp Heading

5. User Input

```
# Static Vs Dynamic
input('Enter Email')
```

```
Enter Emailnitish@gmail.com
'nitish@gmail.com'
```

```
# take input from users and store them in a variable
fnum = int(input('enter first number'))
snum = int(input('enter second number'))
#print(type(fnum),type(snum))
# add the 2 variables
result = fnum + snum
# print the result
print(result)
print(type(fnum))
```

```
enter first number56
enter second number67
123
<class 'int'>
```

6. Type Conversion

```
# Implicit Vs Explicit
print(5+5.6)
print(type(5),type(5.6))

print(4 + '4')
```

```
10.6
<class 'int'> <class 'float'>
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-57-72e5c45cdb6f> in <module>
      3 print(type(5),type(5.6))
      4
----> 5 print(4 + '4')

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
# Explicit
# str -> int
#int(4+5j)

# int to str
str(5)

# float
float(4)
```

```
4.0
```

7. Literals

```
a = 0b1010 #Binary Literals
b = 100 #Decimal Literal
c = 0o310 #Octal Literal
d = 0x12c #Hexadecimal Literal

#Float Literal
float_1 = 10.5
float_2 = 1.5e2 # 1.5 * 10^2
float_3 = 1.5e-3 # 1.5 * 10^-3

#Complex Literal
x = 3.14j

print(a, b, c, d)
print(float_1, float_2,float_3)
print(x, x.imag, x.real)
```

```
# binary
x = 3.14j
print(x.imag)
```

```
3.14
```

```
string = 'This is Python'
strings = "This is Python"
char = "C"
multiline_str = """This is a multiline string with more than one line code."""
unicode = u"\U0001f600\U0001F606\U0001F923"
raw_str = r"raw \n string"

print(string)
print(strings)
print(char)
print(multiline_str)
print(unicode)
print(raw_str)
```

```
This is Python
This is Python
C
This is a multiline string with more than one line code.
😀😆🤣
raw \n string
```

```
a = True + 4
b = False + 10

print("a:", a)
print("b:", b)
```

```
a: 5
b: 10
```

```
k = None
a = 5
b = 6
print('Program exe')
```

```
Program exe
```

8. Operators

```
# Arithmetic
# Relational
# Logical
# Bitwise
# Assignment
# Membership
```

9. If-Else

Start coding or generate with AI.