■ Python Notebook

1. Arithmetic Operators

Operators: *, /, //, %, **, +, -

Precedence (highest → lowest):

() — parentheses

** — exponentiation (evaluates right-to-left)

*, /, //, % — multiplication/division group

+, - — addition/subtraction

✔ / → floating-point division
✔ // → floor division (drops remainder)

■ Always use parentheses () to make order explicit.

2. Comparison Operators

Operators: ==, !=, <, >, <=, >=

Each returns True or False.

✔ Comparisons can be chained:

```
if 1 < x < 5:
    print("x is between 1 and 5")
```

3. Logical Operators

Operators: and, or, not, is, is not, in, not in

✔ Use in / not in to test membership:

```
if "a" in "cat":
    print("Found it!")
```

4. Loops

a. for loop

```
for idx in range(start, stop, step):
    # loop body
```

start → starting value (default = 0)

stop → end value (exclusive)

step → increment (default = 1)

Examples:

```
for i in range(5):       # 0 → 4
    print(i)

for i in range(1, 5):    # 1 → 4
    print(i)

for i in range(5, -1, -1):  # 5 → 0
    print(i)
```

b. while loop

```
cnt = 0
while cnt < 3:
    print(cnt)
    cnt += 1   # must update or loop never ends
```

5. Loop Control Statements

pass → placeholder (does nothing)

continue → skip rest of this loop cycle

break → exit loop completely

6. Branch Control (if / elif / else)

Rules:

Must have 1 if.

May have 0 or more elif.

At most 1 else.

Indentation matters!

```
# simple if
if x == 5:
    print(5)

# if with block
if x == 5:
    print(x)
    x += 1
print(x)   # runs no matter what

# if / else
if x == 5:
    print(x)
else:
    print("x is not 5")

# if / elif / else
if x == 5:
    print(x)
elif x:    # better than "elif x == True"
    print(True)
else:
    print("not 5 and not True")
```

7. Try / Except

Used to catch runtime errors so program doesn’t crash.

```
txt = "Input an integer: "
try:
    x = int(input(txt))
except ValueError:
    print("That was not an integer!")
else:
    print("You entered:", x)
finally:
    print("Done.")
```

8. Example Putting It All Together

```
txt = "Please input an integer: "
while True:
    try:
        counter = int(input(txt))
    except ValueError:
        txt = "You must enter an integer: "
        continue
    else:
        if counter <= 0:
            txt = "You must enter a positive number: "
            continue

        # valid positive integer
        for i in range(counter):
            if i == 2:
                continue
            print(f"The square of {i} is {i*i}")
        break
```

✔ This version avoids infinite loops and ensures errors are handled cleanly.