Here are the notes for the Py Day 15.ipynb file.

**Day 15: Functions**

A function is a reusable block of code that performs a specific task. Functions help organize code, make it more readable, and avoid repetition.

**Pre-defined Functions in Python**

Python has many built-in functions that are ready to use.

**Example 1: min(), max(), sum(), count()**

* **min(iterable)**: Returns the smallest item in an iterable (like a list).
* **max(iterable)**: Returns the largest item in an iterable.
* **sum(iterable)**: Returns the sum of all items in an iterable.
* **.count(value)**: A *method* (a function belonging to an object, like a list) that returns how many times a specific value appears in the list.

**Example Code:**

Python

a = [1,2,3,5,4,6]

min(a) # Output: 1

max(a) # Output: 6

sum(a) # Output: 21

a.count(1) # Output: 1

**Example 2: pow()**

* **pow(base, exponent)**: Returns the base raised to the power of the exponent. (e.g., pow(3, 2) is 3²).

**Example Code:**

Python

x = 3

y = 2

pow(x,y) # Output: 9

**User-Defined Functions**

You can create your own functions using the def keyword.

**Syntax:**

Python

def function\_name(parameter1, parameter2):

# Code block to execute

# (Optional) return value

**Example 1: Basic Function**

**Definition:** This defines a simple function named Great that takes no **parameters** (arguments) and just prints a string when **called**.

**Example Code:**

Python

# Creating a funtion

def Great():

print("Good Morning")

# Calling Great Function

Great()

**Output:**

Plaintext

Good Morning

**Example 2: Function with a Parameter**

**Definition:** This function Great now takes one parameter n. When the function is called, the value passed to it (the **argument**, "Omkar") is assigned to n and used inside the function.

**Example Code:**

Python

def Great(n):

print(f"Hello {n}")

Great("Omkar")

**Output:**

Plaintext

Hello Omkar

**Example 3: Function for Addition**

**Definition:** This function addi takes two parameters, a and b, adds them together, and prints the result.

**Example Code:**

Python

def addi(a,b):

add = a + b

print(add)

addi(458,655)

**Output:**

Plaintext

1113

**Example 4: Function with input()**

**Definition:** This function my\_self takes no parameters but uses the input() function *inside* it to ask the user for their name and age, then prints a formatted string.

**Example Code:**

Python

def my\_self():

a = str(input("Enter Name: "))

b = int(input("Enter Age: "))

print(f"Hello my name is {a} and my age is {b}.")

my\_self()

**Output:**

Plaintext

Enter Name: fs

Enter Age: 21

Hello my name is fs and my age is 21.

**Example 5: Area of a Triangle**

**Definition:** This function calculates the area of a triangle (0.5 \* base \* height) using the two parameters b and h.

**Example Code:**

Python

def area\_of\_triangle(b,h):

a = 0.5 \* b \* h

print(a)

area\_of\_triangle(10,12)

**Output:**

Plaintext

60.0

**Example 6: Pythagoras Theorem (Incorrect)**

**Definition:** This function attempts to calculate the Pythagoras theorem (a² + b² = c²).

**Note: The formula in the file is incorrect.** a\*2 is multiplication, not "to the power of 2". The correct formula for the square is a \*\* 2 or pow(a, 2).

**Example Code (as in file):**

Python

def pythogoras(a,b):

c2 = a\*2 + b\*2 # Incorrect: Should be a\*\*2 + b\*\*2

print(c2)

pythogoras(3,4)

**Output:**

Plaintext

14 # (3\*2 + 4\*2 = 6 + 8 = 14)

**Example 7: Pythagoras Theorem with input() (Incorrect)**

**Definition:** This is the same function as Example 6, but it takes user input. It still contains the **same incorrect formula**.

**Example Code (as in file):**

Python

def pythogoras():

a = int(input("Enter a: "))

b = int(input("Enter b: "))

c2 = a\*2 + b\*2 # Incorrect: Should be a\*\*2 + b\*\*2

print(c2)

pythogoras()

**Output:**

Plaintext

Enter a: 4

Enter b: 5

18 # (4\*2 + 5\*2 = 8 + 10 = 18)

**Example 8: Volume of a Sphere (Incorrect)**

**Definition:** This function attempts to calculate the volume of a sphere (4/3 \* π \* r³).

**Note: This function has two errors:**

1. "Speare" is misspelled (it's "sphere").
2. The formula for the volume of a sphere does not use height (h). The h parameter is accepted by the function but never used.

**Example Code (as in file):**

Python

def volume\_of\_speare(r,h):

v = (4/3) \* 3.14 \* (r \*\* 3) # 'h' is unused

print(v)

volume\_of\_speare(3,4)

**Output:**

Plaintext

113.03999999999999

**Example 9: Prime Number Check**

**Definition:** This function prime(n) checks if a number n is prime. It iterates from 2 up to n-1.

* If n % i == 0 (meaning n is divisible by i), it prints "not prime" and breaks from the loop. A counter c is incremented.
* If the loop finishes *without* finding any divisors (c remains 0), the number is prime.

**Example Code:**

Python

def prime(n):

c = 0

for i in range(2,n):

if n % i == 0:

print(f"{n} is not prime")

c += 1

break

if c == 0:

print(f"{n} is prime")

prime(2)

Output:

(Note: The for loop range(2, 2) is empty, so it correctly goes to the if c == 0 check.)

Plaintext

2 is prime

**Example 10: Percentage Grading (if-elif-else)**

**Definition:** This function percentage takes a user's percentage and uses an if-elif-else chain to determine and print the corresponding grade.

**Example Code:**

Python

def percentage():

p = float(input("Enter Percentage: "))

if p <= 100 and p >= 90:

print("Grade A+")

elif p < 90 and p >= 80:

print("Grade A")

# ... (other elif blocks) ...

elif p < 40 and p >= 33:

print("Grade D")

else:

print("Fail")

percentage()

**Output:**

Plaintext

Enter Percentage: 15

Fail

**Example 11: Percentage Grading (Nested if-else)**

**Definition:** This function percentage\_2 performs the same task as Example 10 but uses a deeply **nested if-else** structure instead of elif. The logic is more complex to read but achieves a similar (though not identical) result.

**Example Code:**

Python

def percentage\_2():

p = float(input("Enter Percentage: "))

if p <= 100:

if p <= 80:

if p <= 70:

if p <= 60:

if p <= 50:

if p <= 35:

if p <= 0:

print("Invalid")

else:

print("Fail")

else:

print("D")

else:

print("C")

else:

print("B")

else:

print("A")

else:

print("A+")

else:

print("Invalid Percentage")

percentage\_2()

**Output:**

Plaintext

Enter Percentage: 59.17

C