**Day 3 Session 1: Numbers Learning Guide**

This guide covers setting up your Jupyter notebook environment and learning basic mathematical operations in Python.

**Setup Instructions**

**1. Open Anaconda Prompt**

Launch the Anaconda Prompt from your system.

**2. Navigate to Your Project Directory**

Execute the following commands in sequence:

E:

cd E:\CompletePython3Bootcamp\Projects\GitDemo

**3. Create PythonTutorials Directory**

mkdir PythonTutorials

**4. Navigate to PythonTutorials Directory**

cd PythonTutorials

**5. Activate Your Conda Environment**

conda activate PythonBootcampEnvironment

**6. Launch Jupyter Notebook**

jupyter notebook

**7. Create New Notebook**

1. Click **"New"** button in the top right corner
2. Select **"Python 3"** from the dropdown menu
3. A new untitled notebook will open

**8. Rename Your Notebook**

1. Click on **"Untitled"** at the top of the notebook
2. Rename it to Session1\_Numbers
3. The file will be saved as Session1\_Numbers.ipynb in the PythonTutorials directory

**Python Mathematical Operations Practice**

Copy and execute the following code blocks in your Jupyter notebook:

**Addition Operation**

# add two numbers

# use '+' operator

2 + 4

Output: 6

**Subtraction Operation**

# subtract two numbers

# use '-' operator

8 - 4

Output: 4

**Multiplication Operation**

# multiply two numbers

# use '\*' operator

4 \* 6

Output: 24

**Division Operation**

# divide two numbers

# use '/' operator

# '/' returns the float

8 / 2

Output: 4.0

# divide two numbers

# use '/' operator

# '/' returns the float

6 / 4

Output: 1.5

# divide two numbers

# use '/' operator

# '/' returns the float

8 / 3

Output: 2.6666666666666665

**Multiple Operations (BODMAS Rule)**

BODMAS stands for: **B**racket, **O**rder, **D**ivision, **M**ultiplication, **A**ddition, **S**ubtraction

# for multiple operation at a time BODMAS rule is used

8 + 9 - 10

Output: 7

# for multiple operation at a time BODMAS rule is used

8 + 2 \* 3

Output: 14

# for multiple operation at a time BODMAS rule is used

(8 + 2) \* 3

Output: 30

**Power Operations**

# for to-the-power - such as 'x' to-the-power 'y', we can use any of the two ways below:

# 1) x \* x \* x \* .... y times x

# 2) x \*\* y

# for an example, 2 to-the-power 3

2 \* 2 \* 2

Output: 8

# for to-the-power - such as 'x' to-the-power 'y', we can use any of the two ways below:

# 1) x \* x \* x \* .... y times x

# 2) x \*\* y

# for an example, 2 to-the-power 3

2 \*\* 3

Output: 8

# for to-the-power - such as 'x' to-the-power 'y', we can use any of the two ways below:

# 1) x \* x \* x \* .... y times x

# 2) x \*\* y

# for an example, 4 to-the-power 6

4 \* 4 \* 4 \* 4 \* 4 \* 4

Output: 4096

# for to-the-power - such as 'x' to-the-power 'y', we can use any of the two ways below:

# 1) x \* x \* x \* .... y times x

# 2) x \*\* y

# for an example, 4 to-the-power 6

4 \*\* 6

Output: 4096

**Floor Division Operation**

# for a division, if we want only quotient part, we will use floor division '//' operator

# '//' returns only the integer quotient

5 // 2

Output: 2

**Modulus Operation**

# for a division, if we want only remainder part, we will use percentile or modulus '%' operator

5 % 2

Output: 1

**Determining Even or Odd Numbers**

# to determine any-number is even or odd

# use percentile or modulus operator '%'

# any-number % 2

# the remainder will be '0' for even number

# the remainder will be '1' for odd number

# '45' is the odd number as the remainder is '1'

45 % 2

Output: 1

# to determine any-number is even or odd

# use percentile or modulus operator '%'

# any-number % 2

# the remainder will be '0' for even number

# the remainder will be '1' for odd number

# '34' is the even number as the remainder is '0'

34 % 2

Output: 0

**Mathematical Operators Summary**

| **Operator** | **Name** | **Example** | **Result** |
| --- | --- | --- | --- |
| + | Addition | 5 + 3 | 8 |
| - | Subtraction | 5 - 3 | 2 |
| \* | Multiplication | 5 \* 3 | 15 |
| / | Division | 6 / 3 | 2.0 |
| \*\* | Power (Exponentiation) | 2 \*\* 3 | 8 |
| // | Floor Division | 7 // 2 | 3 |
| % | Modulus (Remainder) | 7 % 2 | 1 |

**Key Learning Points**

* Python follows the BODMAS rule for order of operations
* Division (/) always returns a float number
* Floor division (//) returns only the integer part of division
* Modulus (%) returns only the remainder of division
* Power operations can be done using \*\* operator
* Modulus operator is useful for determining even/odd numbers

**BODMAS Rule Table**

BODMAS defines the order of operations in mathematical expressions:

| **Order** | **Operation** | **Symbol** | **Description** | **Example** |
| --- | --- | --- | --- | --- |
| 1 | **B**rackets | ( ) | Operations inside brackets are performed first | (2 + 3) \* 4 = 20 |
| 2 | **O**rders | \*\* | Exponentiation (powers/indices) | 2 \*\* 3 \* 4 = 32 |
| 3 | **D**ivision | /, // | Division operations (left to right) | 12 / 3 / 2 = 2.0 |
| 4 | **M**ultiplication | \* | Multiplication operations (left to right) | 2 \* 3 \* 4 = 24 |
| 5 | **A**ddition | + | Addition operations (left to right) | 5 + 3 + 2 = 10 |
| 6 | **S**ubtraction | - | Subtraction operations (left to right) | 10 - 3 - 2 = 5 |

**BODMAS Examples**

# Without brackets: 8 + 2 \* 3 = 8 + 6 = 14

8 + 2 \* 3 # Result: 14

# With brackets: (8 + 2) \* 3 = 10 \* 3 = 30

(8 + 2) \* 3 # Result: 30

# Complex example: 2 + 3 \* 4 \*\* 2 - 1

# Step 1: 4 \*\* 2 = 16 (Orders first)

# Step 2: 3 \* 16 = 48 (Multiplication next)

# Step 3: 2 + 48 = 50 (Addition)

# Step 4: 50 - 1 = 49 (Subtraction last)

2 + 3 \* 4 \*\* 2 - 1 # Result: 49

**Remember:** Operations of the same priority (like multiplication and division) are performed from left to right.