

Data Engineering 101

Python Basics



Shwetank Singh
GritSetGrow - GSGLearn.com

List Comprehension

A concise way to create lists based on existing lists

```
squares = [x**2 for x in range(10)]
```

Creates a list of squares for numbers 0 to 9



Shwetank Singh
GritSetGrow - GSGLearn.com



Dictionary Comprehension

Create dictionaries using an expression

```
word_lengths = {word: len(word) for word in ['cat',  
'dog', 'elephant']}
```

Creates a dictionary with words as keys and their lengths as values



Shwetank Singh
GritSetGrow - GSGLearn.com



Lambda Functions

Small anonymous functions

```
multiply = lambda x, y: x * y
```

Creates a function that multiplies two numbers



Shwetank Singh
GritSetGrow - GSGLearn.com



Map Function

Apply a function to all items in an iterable

```
numbers = list(map(lambda x: x * 2, [1, 2, 3, 4]))
```

Multiplies each number in the list by 2



Shwetank Singh
GritSetGrow - GSGLearn.com



Filter Function

Create a list of elements for which a function returns True

```
even_numbers = list(filter(lambda x: x % 2 == 0, [1, 2, 3, 4, 5, 6]))
```

Creates a list of even numbers



Shwetank Singh
GritSetGrow - GSGLearn.com



Zip Function

Combine two lists into a list of tuples

```
combined = list(zip([1, 2, 3], ['a', 'b', 'c']))
```

Creates a list of tuples: [(1, 'a'), (2, 'b'), (3, 'c')]



Shwetank Singh
GritSetGrow - GSGLearn.com



Enumerate Function

Loop over a list and have an automatic counter

```
for index, value in enumerate(['a', 'b', 'c']):  
    print(index, value)
```

Prints the index and value of each item in the list



Shwetank Singh
GritSetGrow - GSGLearn.com



List Slicing

Extract a portion of a list

```
my_list = [0, 1, 2, 3, 4, 5]; print(my_list[1:4])
```

Prints [1, 2, 3] (elements from index 1 to 3)



Shwetank Singh
GritSetGrow - GSGLearn.com



String Formatting

Format strings with variables

```
name = "Alice"; age = 30; print(f"{name} is {age}  
years old")
```

Prints "Alice is 30 years old"



Shwetank Singh
GritSetGrow - GSGLearn.com



Try-Except Block

Handle exceptions in code

```
try:  
    result = 10 / 0;  
except ZeroDivisionError:  
    print("Cannot divide by zero")
```

Catches the ZeroDivisionError and prints an error message



Shwetank Singh
GritSetGrow - GSGLearn.com

List Unpacking

Assign multiple variables from a list

`a, b, c = [1, 2, 3]`

Assigns 1 to a, 2 to b, and 3 to c



Shwetank Singh
GritSetGrow - GSGLearn.com



Ternary Operator

Conditional expression in one line

```
x = 10; result = "Even" if x % 2 == 0 else "Odd"
```

Assigns "Even" to result if x is even, otherwise "Odd"



Shwetank Singh
GritSetGrow - GSGLearn.com



Set Comprehension

Create sets using an expression

```
unique_letters = {char for char in 'mississippi'}
```

Creates a set of unique letters from the word
'mississippi'



Shwetank Singh
GritSetGrow - GSGLearn.com

Generator Expression

Create a generator object

```
gen = (x**2 for x in range(10))
```

Creates a generator that yields squares of numbers 0 to 9



Shwetank Singh
GritSetGrow - GSGLearn.com



Context Manager

Manage resources using 'with' statement

with open('file.txt', 'r') as f: content = f.read()

Opens a file, reads its content, and automatically closes it



Shwetank Singh
GritSetGrow - GSGLearn.com



Decorators

Modify or enhance functions

```
def uppercase_decorator(func):  
    return lambda: func().upper()
```

Creates a decorator that converts the result of a function to uppercase



Shwetank Singh
GritSetGrow - GSGLearn.com



Type Hinting

Add type annotations to functions

```
def greet(name: str) -> str: return f"Hello, {name}!"
```

Specifies that the function takes a string and returns a string



Shwetank Singh
GritSetGrow - GSGLearn.com



Default Dictionary

Dictionary subclass that provides default values

```
from collections import defaultdict;  
d = defaultdict(int)
```

Creates a dictionary that will have a default value of 0 for any new key



Shwetank Singh
GritSetGrow - GSGLearn.com



Named Tuple

Create tuple subclasses with named fields

```
from collections import namedtuple;  
Point = namedtuple('Point', ['x', 'y'])
```

Creates a named tuple 'Point' with 'x' and 'y' fields



Shwetank Singh
GritSetGrow - GSGLearn.com



Partial Functions

Create a new function with some parameters pre-set

```
from functools import partial;  
add_five = partial(lambda x, y: x + y, 5)
```

Creates a new function that always adds 5 to its argument



Shwetank Singh
GritSetGrow - GSGLearn.com



Memoization

Cache function results for optimization

```
from functools import lru_cache;  
@lru_cache(maxsize=None)  
def fib(n): return n if n < 2 else fib(n-1) + fib(n-2)
```

Caches results of the Fibonacci function for faster subsequent calls



Shwetank Singh
GritSetGrow - GSGLearn.com

Counter

Count hashable objects

```
from collections import Counter;  
c = Counter('mississippi')
```

Counts the occurrences of each character in
'mississippi'



Shwetank Singh
GritSetGrow - GSGLearn.com



Chain Map

Combine multiple dictionaries

```
from collections import ChainMap;  
combined = ChainMap({'a': 1}, {'b': 2})
```

Creates a view of multiple dictionaries as a single dictionary



Shwetank Singh
GritSetGrow - GSGLearn.com



Itertools Cycle

Iterate over a sequence indefinitely

```
from itertools import cycle;  
colors = cycle(['red', 'green', 'blue'])
```

Creates an iterator that cycles through the colors indefinitely



Shwetank Singh
GritSetGrow - GSGLearn.com



Any and All Functions

Check conditions for any or all items

```
all_positive = all(x > 0 for x in [1, 2, 3, 4]);  
any_even = any(x % 2 == 0 for x in [1, 2, 3, 4])
```

Checks if all numbers are positive and if any number is even



Shwetank Singh
GritSetGrow - GSGLearn.com



Sorting with Key

Sort a list using a key function

```
sorted(['apple', 'banana', 'cherry'], key=len)
```

Sorts the list of fruits based on their length



Shwetank Singh
GritSetGrow - GSGLearn.com



Flatten a List of Lists

Combine nested lists into a single list

```
flat_list = [item for sublist in [[1, 2], [3, 4]] for item in  
sublist]
```

Creates a flat list [1, 2, 3, 4] from the nested lists



Shwetank Singh
GritSetGrow - GSGLearn.com



Merging Dictionaries

Combine two or more dictionaries

`{**dict1, **dict2}`

Merges dict1 and dict2, with dict2's values overwriting dict1's if there are conflicts



Shwetank Singh
GritSetGrow - GSGLearn.com



Swapping Variables

Exchange values without a temporary variable

$a, b = b, a$

Swaps the values of variables a and b



Shwetank Singh
GritSetGrow - GSGLearn.com



Reversing a String

Reverse the characters in a string

```
reversed_string = "Hello"[::-1]
```

Creates a new string with the characters of
"Hello" in reverse order



Shwetank Singh
GritSetGrow - GSGLearn.com

THANK
YOU