Python Files

---------- **FileName:**
10PythonC.py

Content:
```
text = "P@ssw0rd"
vowels = "aeiouAEIOU"
new_text = ""

for char in text:
if char in vowels:
new_text += "*"
else:
new_text += char

print(new_text)
```

---------- **FileName:**
11PythonC.py

Content:
```
print("Hello, world")
```

---------- **FileName:**
12PythonC.py

Content:
```
Input = input("Enter a Your Name: ")

print("Welcome, " + Input)
```

---------- **FileName:**
13PythonC.py

Content:
```
a = "1337"

print(int(a) + 10)
```

---------- **FileName:**
14PythonC.py

Content:
```
a = float(input("First number: "))
b = float(input("Second number: "))
```

```
print("Sum:", a + b)
print("Difference:", a - b)
print("Product:", a * b)

if b != 0:
print("Quotient:", a / b)
else:
print("Quotient: Cannot divide by zero")
```

FileName: 15PythonC.py

Content:

```
Target = "rekcah_repus"

print(Target[::-1])
```

FileName: 16PythonC.py

Content:

```
# Even or Odd Checker

num = input("Enter a number: ")

# Validate input
if not num.strip().lstrip('-').isdigit():
print("That's not a valid integer!")
else:
num = int(num)
if num % 2 == 0:
print(f"{num} is even.")
else:
print(f"{num} is odd.")
```

FileName: 17PythonC.py

Content:

```
StoreValue = True

print(StoreValue)
```

FileName: 18PythonC.py

Content:
```
binary_string = "100101"
decimal_number = int(binary_string, 2)
print(decimal_number)
```

---------- **FileName:**
19PythonC.py

Content:
```
Target = "H4ck3r"
print(Target.capitalize())
```

---------- **FileName:**
1PythonC.py

Content:
```
print("Hello, world")
```

---------- **FileName:**
20PythonC.py

Content:
```
text = "P@ssw0rd"
vowels = "aeiouAEIOU"
new_text = ""

for char in text:
if char in vowels:
new_text += "*"
else:
new_text += char

print(new_text)
```

---------- **FileName:**
21PythonC.py

Content:
```
Array = ["Nmap", "Metasploit", "Wireshark", "Burp suite", "JohnTheRipper", "whatweb", "Nikto",
"OpenVAS", "Aircrack-ng", "Hydra"]

print(Array)
```

---------- **FileName:**
22PythonC.py

Content:
```
Array = ["Nmap", "Metasploit", "Wireshark", "Burp suite", "JohnTheRipper", "whatweb", "Nikto",
"OpenVAS", "Aircrack-ng", "Hydra"]

print(Array[:4])
```

---------- **FileName:** 23PythonC.py

Content:
```
http_status_codes = {
200: "OK",
404: "Not Found"
}

def main():
print("--- Interactive HTTP Status Code Lookup ---")
print("Enter a status code to find its meaning, or type 'quit' to exit.")

while True:
user_input = input("\nEnter a status code: ").strip().lower()

if user_input in ["q", "quit", "exit"]:
print("Goodbye!")
break

try:
code = int(user_input)
meaning = http_status_codes.get(code, "Sorry, that code is not in our dictionary.")
print(f"-> The meaning of {code} is: {meaning}")
except ValueError:
print("! Invalid input. Please enter a number for the status code.")

if __name__ == "__main__":
main()
```

---------- **FileName:** 24PythonC.py

Content:
```
target = input("Enter string: ")
HowMuch = len(target)
print(HowMuch)
```

---------- **FileName:** 25PythonC.py

Content:
```
import random
```

```
target = random.randint(1, 100000000000000000)

print(target)
```

---------- **FileName:** 26PythonC.py

Content:
```
Ports = input("Enter Port: ")
Services = input("Enter Service: ")

arrayDatastore = [Ports, Services]
print("success to datastore in array")

print("-----------------------------------------")
print("advanced options")
print("-----------------------------------------")
print(arrayDatastore)
```

---------- **FileName:** 27PythonC.py

Content:
```
def remove_duplicates(lst):
result = []
for item in lst:
if item not in result:
result.append(item)
return result

numbers = [4, 5, 9, 4, 9, 1, 5]
Result = remove_duplicates(numbers)
print(Result)
```

---------- **FileName:** 28PythonC.py

Content:
```
array = ["hello", "world", "system", "code", "Discord", "is", "The", "best"]

comma_string = ", ".join(array)

print(comma_string)
```

---------- **FileName:** 29PythonC.py

Content:
```
array = ["hello", "world", "system", "code", "Discord", "is", "The", "best",
```

```
"systemErrorCommandPrompt"]

Result = max(array, key=len)
print(Result)
```

---------- **FileName:** 2PythonC.py

Content:
```
Input = input("Enter a Your Name: ")

print("Welcome, " + Input)
```

---------- **FileName:** 30PythonC.py

Content:
```
Input = input("Enter to get the Username and password: ")
Array = ["Jack", "password123", "Alex", "#DEWA12e", "ahmed", "Pqrs1234"]

if Input == "Jack":
print("Jack")
print("password123")
if Input == "Alex":
print("Alex")
print("#DEWA12e")
if Input == "ahmed":
print("ahmed")
print("Pqrs1234")
else:
print("Error. no someone in database with this name")
```

---------- **FileName:** 31PythonC.py

Content:
```
string = input("type string to reverse it: ")

def Reverse_words():
global string
reversed_string = string[::-1]
print(reversed_string)


Reverse_words()
```

---------- **FileName:** 32PythonC.py

Content:

```
import string

def check_password_strength(password):
if len(password) < 8:
return "Your password is too short. Make it at least 8 characters."

if not any(char.isdigit() for char in password):
return "Add at least one number to make it stronger."

if not any(char in string.punctuation for char in password):
return "Throw in a special character (!, @, #, etc.) for extra security."

return "Nice! Your password looks strong."


user_password = input("Type your password: ")
print(check_password_strength(user_password))
```

FileName: 33PythonC.py

Content:

```
import random
import string

def generate_password(length):
characters = string.ascii_letters + string.digits + string.punctuation
password = ''.join(random.choice(characters) for _ in range(length))
return password

if __name__ == "__main__":
try:
password_length = int(input("Enter the desired password length: "))
if password_length <= 0:
print("Password length must be a positive number.")
else:
new_password = generate_password(password_length)
print(f"Generated Password: {new_password}")
except ValueError:
print("Invalid input. Please enter a number for the password length.")
```

FileName: 34PythonC.py

Content:

```
import hashlib

# Hashing a string
Input = input("enter string to hash it: ")
hash_object_sha256 = hashlib.md5(Input.encode())
hex_digest = hash_object_sha256.hexdigest()
```

```
print(f"Hash: {hex_digest}")

# Hashing a file iteratively
def hash_file(filepath):
hasher = hashlib.sha256()
with open(filepath, 'rb') as f:
while True:
chunk = f.read(4096) # Read in 4KB chunks
if not chunk:
break
hasher.update(chunk)
return hasher.hexdigest()

# Example usage (assuming a file named 'example.txt' exists)
# file_hash = hash_file('example.txt')
# print(f"File hash: {file_hash}")
```

FileName: 35PythonC.py

Content:

```
import os
Input = input("enter the ip to check he is online or not: ")

os.system("ping " + Input)
print("is he working know he is online")
```

FileName: 36PythonC.py

Content:

```
import random

def random_mac():
parts = []
for _ in range(6):
num = random.randint(0, 255)
parts.append(f"{num:02x}")
return ":".join(parts)

for i in range(5):
print(random_mac())
```

FileName: 37PythonC.py

Content:

```
def xor_encrypt_decrypt(data, key):

xored_data = bytearray()
```

```
key_len = len(key)
for i in range(len(data)):
xored_data.append(data[i] ^ key[i % key_len])
return bytes(xored_data)

Input = input("enter string to encrypt it: ")
plaintext = b"Hello, World!"
key = b"secret"

encrypted_data = xor_encrypt_decrypt(plaintext, key)
print(f"Encrypted: {encrypted_data}")
```

FileName: 38PythonC.py

Content:

```
import uuid

UUid = uuid.uuid1()
print(UUid)
```

FileName: 39PythonC.py

Content:

```
import socket

hostname = socket.gethostname()
ip_address = socket.gethostbyname(hostname)
print(f"Result: {hostname} {ip_address}")
```

FileName: 3PythonC.py

Content:

```
a = "1337"

print(int(a) + 10)
```

FileName: 40PythonC.py

Content:

```
Input = input("enter a String: ")
vowels = ["a", "e", "i", "o", "u"]

for char in Input:
if char.lower() in vowels:
```

```
print(f"The string contains the vowel: {char}")
```

FileName: 4PythonC.py

Content:

```
a = float(input("First number: "))
b = float(input("Second number: "))

print("Sum:", a + b)
print("Difference:", a - b)
print("Product:", a * b)

if b != 0:
print("Quotient:", a / b)
else:
print("Quotient: Cannot divide by zero")
```

FileName: 5PythonC.py

Content:

```
Target = "rekcah_repus"

print(Target[::-1])
```

FileName: 6PythonC.py

Content:

```
# Even or Odd Checker

num = input("Enter a number: ")

# Validate input
if not num.strip().lstrip('-').isdigit():
print("That's not a valid integer!")
else:
num = int(num)
if num % 2 == 0:
print(f"{num} is even.")
else:
print(f"{num} is odd.")
```

FileName: 7PythonC.py

Content:
```
StoreValue = True

print(StoreValue)
```

---------- **FileName:** 8PythonC.py

Content:
```
binary_string = "100101"
decimal_number = int(binary_string, 2)
print(decimal_number)
```

---------- **FileName:** 9PythonC.py

Content:
```
Target = "H4ck3r"
print(Target.capitalize())
```

---------- **FileName:** main.py

Content:
```
from reportlab.lib.pagesizes import A4
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Table, TableStyle
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib import colors
import os

def read_file_content(filename):
"""Safely read file content as text."""
try:
with open(filename, "r", encoding="utf-8") as f:
return f.read()
except Exception as e:
return f"Error reading file: {e}"

def add_file_section(story, styles, filename, content):
"""Add one file’s section to the PDF."""
section = f"""
```

FileName: {filename}

Content:
```
{content.replace('\n', '
')}
```

```
"""
story.append(Paragraph(section, styles["Normal"]))
story.append(Spacer(1, 20))

def make_pdf():
doc = SimpleDocTemplate("result.pdf", pagesize=A4)
styles = getSampleStyleSheet()
story = []

include_all = input("Do you want to include all files in this folder? (yes/no): ").strip().lower()

if include_all == "yes":
py_files = [f for f in os.listdir() if f.endswith(".py")]
txt_files = [f for f in os.listdir() if f.endswith(".txt")]

# ---- Add .py files sections ----
if py_files:
story.append(Paragraph("**Python Files**", styles["Heading2"]))
story.append(Spacer(1, 10))
for f in py_files:
content = read_file_content(f)
add_file_section(story, styles, f, content)

# ---- Add .txt files sections ----
if txt_files:
story.append(Paragraph("**Text Files**", styles["Heading2"]))
story.append(Spacer(1, 10))
for f in txt_files:
content = read_file_content(f)
add_file_section(story, styles, f, content)

# ---- Python Files Table ----
if py_files:
data = [["Python File Name", "Preview (first 80 chars)"]]
for f in py_files:
code = read_file_content(f)
preview = (code[:80] + "...") if len(code) > 80 else code
data.append([f, preview])
table = Table(data, colWidths=[200, 300])
table.setStyle(TableStyle([
("BACKGROUND", (0, 0), (-1, 0), colors.lightblue),
("GRID", (0, 0), (-1, -1), 1, colors.black),
("ALIGN", (0, 0), (-1, -1), "LEFT"),
]))
story.append(Paragraph("**Python Files Summary**", styles["Heading3"]))
story.append(table)
story.append(Spacer(1, 20))

# ---- Text Files Table ----
if txt_files:
data = [["Text File Name", "Preview (first 80 chars)"]]
for f in txt_files:
text = read_file_content(f)
preview = (text[:80] + "...") if len(text) > 80 else text
data.append([f, preview])
table = Table(data, colWidths=[200, 300])
```

```
table.setStyle(TableStyle([
("BACKGROUND", (0, 0), (-1, 0), colors.lightgreen),
("GRID", (0, 0), (-1, -1), 1, colors.black),
("ALIGN", (0, 0), (-1, -1), "LEFT"),
]))
story.append(Paragraph("Text Files Summary", styles["Heading3"]))
story.append(table)

print("■ Added all .py and .txt files to result.pdf")

else:
# ---- Single file mode ----
filename = input("Enter the file name (.py or .txt): ").strip()
if not os.path.exists(filename):
print("■ File not found.")
return
if not (filename.endswith(".py") or filename.endswith(".txt")):
print("■ Only .py and .txt files are allowed.")
return

content = read_file_content(filename)
add_file_section(story, styles, filename, content)
print(f"■ Added {filename} to result.pdf")

# ---- Build PDF ----
doc.build(story)
print("■ PDF created successfully: result.pdf")

if __name__ == "__main__":
make_pdf()
```

Python Files Summary

Python File Name	Preview (first 80 chars)
10PythonC.py	text = "P@ssw0rd" vowels = "aeiouAEIOU" new_text = "" for char in text: if ...
11PythonC.py	print("Hello, world")
12PythonC.py	Input = input("Enter a Your Name: ") print("Welcome, " + Input)
13PythonC.py	a = "1337" print(int(a) + 10)

14PythonC.py	a = float(input("First number: ")) b = float(input("Second number: ")) print("S...
15PythonC.py	Target = "rekcah_repus" print(Target[::-1])
16PythonC.py	# Even or Odd Checker num = input("Enter a number: ") # Validate input if not ...
17PythonC.py	StoreValue = True print(StoreValue)
18PythonC.py	binary_string = "100101" decimal_number = int(binary_string, 2) print(decimal_nu...
19PythonC.py	Target = "H4ck3r" print(Target.capitalize())
1PythonC.py	print("Hello, world")
20PythonC.py	text = "P@ssw0rd" vowels = "aeiouAEIOU" new_text = "" for char in text: if ...
21PythonC.py	Array = ["Nmap", "Metasploit", "Wireshark", "Burp suite", "JohnTheRipper", "w
22PythonC.py	Array = ["Nmap", "Metasploit", "Wireshark", "Burp suite", "JohnTheRipper", "w

23PythonC.py	http_status_codes = {
	  200: "OK",
	  404: "Not Found"
	}
	
	def main():
	  pri...

24PythonC.py	target = input("Enter string: ") HowMuch = len(target) print(HowMuch)
25PythonC.py	import random target = random.randint(1, 100000000000000000) print(target)
26PythonC.py	Ports = input("Enter Port: ") Services = input("Enter Service: ") arrayDatastor...

27PythonC.py	def remove_duplicates(lst):
	    result = []
	    for item in lst:
	        if item...

28PythonC.py	array = ["hello", "world", "system", "code", "Discord", "is", "The", "best"] co...
29PythonC.py	array = ["hello", "world", "system", "code", "Discord", "is", "The", "best", "sy...
2PythonC.py	Input = input("Enter a Your Name: ") print("Welcome, " + Input)
30PythonC.py	Input = input("Enter to get the Username and password: ") Array = ["Jack", "pass...
31PythonC.py	string = input("type string to reverse it: ") def Reverse_words(): global s...

32PythonC.py	import string
	
	def check_password_strength(password):
	    if len(password) < 8:
	...

33PythonC.py	import random import string def generate_password(length): characters = str...

34PythonC.py	import hashlib
	
	# Hashing a string
	Input = input("enter string to hash it: ")
	ha...

35PythonC.py	import os Input = input("enter the ip to check he is online or not: ") os.syste...

36PythonC.py	import random
	
	def random_mac():
	    parts = []
	    for _ in range(6):
	        n...

37PythonC.py	def xor_encrypt_decrypt(data, key):
	
	    xored_data = bytearray()
	    key_len = ...

38PythonC.py	import uuid UUid = uuid.uuid1() print(UUid)

39PythonC.py	import socket hostname = socket.gethostname() ip_address = socket.gethostbyname...
3PythonC.py	a = "1337" print(int(a) + 10)
40PythonC.py	Input = input("enter a String: ") vowels = ["a", "e", "i", "o", "u"] for char i...
4PythonC.py	a = float(input("First number: ")) b = float(input("Second number: ")) print("S...
5PythonC.py	Target = "rekcah_repus" print(Target[::-1])
6PythonC.py	# Even or Odd Checker num = input("Enter a number: ") # Validate input if not ...
7PythonC.py	StoreValue = True print(StoreValue)
8PythonC.py	binary_string = "100101" decimal_number = int(binary_string, 2) print(decimal_nu...
9PythonC.py	Target = "H4ck3r" print(Target.capitalize())
main.py	from reportlab.lib.pagesizes import A4 from reportlab.platypus import SimpleDocT...