22 Module Assessments in Python

This document provides 22 assessments covering the math, datetime, random, re, json, collections, and itertools modules in Python, ranging from easy to hard, including their solutions and expected results.

Math Module Assessments

1. Calculate Factorial

Write a function to calculate the factorial of a number using the math module.

```
import math

def calculate_factorial(n):
    return math.factorial(n)

# Test
print(calculate_factorial(5))  # Output: 120
```

2. Calculate GCD

Write a function to calculate the greatest common divisor (GCD) of two numbers using the math module.

```
import math

def calculate_gcd(a, b):
    return math.gcd(a, b)

# Test
print(calculate_gcd(48, 18))  # Output: 6
```

Datetime Module Assessments

3. Get Current Date and Time

Write a function to get the current date and time using the datetime module.

```
from datetime import datetime

def get_current_datetime():
    return datetime.now()

# Test
print(get_current_datetime())  # Output: current date and time
```

4. Calculate Age

Write a function to calculate age given a birthdate using the datetime module.

```
from datetime import datetime

def calculate_age(birthdate):
    today = datetime.now()
    age = today.year - birthdate.year - ((today.month, today.day) < (birthdate.month, birthd
    return age

# Test
print(calculate_age(datetime(1990, 7, 22)))  # Output: age in years
```

Random Module Assessments

5. Generate Random Integer

Write a function to generate a random integer between two numbers using the random module.

```
import random

def generate_random_integer(a, b):
    return random.randint(a, b)

# Test
print(generate_random_integer(1, 10))  # Output: random integer between 1 and 10
```

6. Shuffle a List

Write a function to shuffle a list using the random module.

```
import random

def shuffle_list(lst):
    random.shuffle(lst)
    return lst

# Test
print(shuffle_list([1, 2, 3, 4, 5]))  # Output: shuffled list
```

Re Module Assessments

7. Find All Matches

Write a function to find all matches of a pattern in a string using the re module.

```
import re
```

```python
def find_all_matches(pattern, string):
    return re.findall(pattern, string)

# Test
print(find_all_matches(r'\b\w+\b', 'This is a test'))  # Output: ['This', 'is', 'a', 'test']
```

8. Replace Pattern

Write a function to replace all occurrences of a pattern in a string with a replacement using the re module.

```python
import re

def replace_pattern(pattern, repl, string):
    return re.sub(pattern, repl, string)

# Test
print(replace_pattern(r'\b\w+\b', 'word', 'This is a test'))  # Output: 'word word word word'
```

Json Module Assessments

9. Convert Dictionary to JSON

Write a function to convert a dictionary to a JSON string using the json module.

```python
import json

def dict_to_json(d):
    return json.dumps(d)

# Test
print(dict_to_json({'a': 1, 'b': 2}))  # Output: '{"a": 1, "b": 2}'
```

10. Convert JSON to Dictionary

Write a function to convert a JSON string to a dictionary using the json module.

```python
import json

def json_to_dict(json_str):
    return json.loads(json_str)

# Test
print(json_to_dict('{"a": 1, "b": 2}'))  # Output: {'a': 1, 'b': 2}
```

Collections Module Assessments

11. Count Elements

Write a function to count the occurrences of elements in a list using collections.Counter.

```
from collections import Counter

def count_elements(lst):
    return Counter(lst)

# Test
print(count_elements([1, 2, 2, 3, 3, 3]))  # Output: Counter({3: 3, 2: 2, 1: 1})
```

12. Group by Key

Write a function to group elements by a key using collections.defaultdict.

```
from collections import defaultdict

def group_by_key(lst, key_func):
    grouped = defaultdict(list)
    for item in lst:
        grouped[key_func(item)].append(item)
    return dict(grouped)

# Test
print(group_by_key(['apple', 'banana', 'cherry', 'date'], lambda x: x[0]))
# Output: {'a': ['apple'], 'b': ['banana'], 'c': ['cherry'], 'd': ['date']}
```

Itertools Module Assessments

13. Get Cartesian Product

Write a function to get the Cartesian product of two lists using itertools.product.

```
from itertools import product

def cartesian_product(lst1, lst2):
    return list(product(lst1, lst2))

# Test
print(cartesian_product([1, 2], ['a', 'b']))  # Output: [(1, 'a'), (1, 'b'), (2, 'a'), (2,
```

14. Group by Key

Write a function to group elements by a key using itertools.groupby.

```python
from itertools import groupby

def group_by_key(lst, key_func):
    grouped = {k: list(g) for k, g in groupby(sorted(lst, key=key_func), key_func)}
    return grouped

# Test
print(group_by_key(['apple', 'banana', 'cherry', 'date'], lambda x: x[0]))
# Output: {'a': ['apple'], 'b': ['banana'], 'c': ['cherry'], 'd': ['date']}
```

Advanced Assessments

15. Calculate Combination

Write a function to calculate the number of combinations (nCr) using the math module.

```python
import math

def calculate_combination(n, r):
    return math.comb(n, r)

# Test
print(calculate_combination(5, 2))  # Output: 10
```

16. Get Date Difference

Write a function to get the difference between two dates in days using the datetime module.

```python
from datetime import datetime

def date_difference(date1, date2):
    return abs((date2 - date1).days)

# Test
print(date_difference(datetime(2024, 1, 1), datetime(2024, 1, 10)))  # Output: 9
```

17. Generate Random Choice

Write a function to randomly select an element from a list using the random module.

```python
import random

def random_choice(lst):
    return random.choice(lst)
```

```
# Test
print(random_choice([1, 2, 3, 4, 5]))  # Output: random element from the list
```

18. Validate Email

Write a function to validate an email address using the re module.

```
import re

def validate_email(email):
    pattern = r'^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$'
    return re.match(pattern, email) is not None

# Test
print(validate_email('test@example.com'))  # Output: True
print(validate_email('invalid-email'))  # Output: False
```

19. Pretty Print JSON

Write a function to pretty print a JSON string using the json module.

```
import json

def pretty_print_json(json_str):
    return json.dumps(json.loads(json_str), indent=4)

# Test
print(pretty_print_json('{"a": 1, "b": 2}'))  # Output: pretty-printed JSON string
```

20. Count Unique Elements

Write a function to count the number of unique elements in a list using collections.Counter.

```
from collections import Counter

def count_unique_elements(lst):
    return len(Counter(lst))

# Test
print(count_unique_elements([1, 2, 2, 3, 3, 3]))  # Output: 3
```

21. Generate Permutations

Write a function to generate all permutations of a list using itertools.permutations.

```
from itertools import permutations

def generate_permutations(lst):
```

```
    return list(permutations(lst))

# Test
print(generate_permutations([1, 2, 3]))  # Output: all permutations of the list
```

22. Convert Date to String

Write a function to convert a datetime object to a formatted string using the datetime module.

```
from datetime import datetime

def date_to_string(date, format):
    return date.strftime(format)

# Test
print(date_to_string(datetime(2024, 1, 1), "%Y-%m-%d"))  # Output: '2024-01-01'
```

Stay Updated

Be sure to this repository to stay updated with new examples and enhancements!

License

This project is protected under the MIT License.

Contact

Panagiotis Moschos - pan.moschos86@gmail.com

Note: This is a Python script and requires a Python interpreter to run.

Happy Coding

Made with by Panagiotis Moschos (https://github.com/pmoschos)