```
# 1. Create a tuple 'fruits' containing '("apple", "banana", "cherry")'. Access and print the second element of the tuple.
fruits = ("apple", "banana", "cherry")
secondfruit = fruits[1]
print("Second fruit:", secondfruit)
```

```
Second fruit: banana
```

```
# 2. Given the tuple 'numbers = (1, 2, 3, 4, 5)', use slicing to print the subtuple '(2, 3, 4)'.
numbers = (1, 2, 3, 4, 5)
subtuple = numbers[1:4]
print("Subtuple from numbers:", subtuple)
```

```
Subtuple from numbers: (2, 3, 4)
```

```
# 3. Create a tuple 'colors' with the elements '("red", "green", "blue")'. Use the '.index()' method to find the index of "green" and pr
colors = ("red", "green", "blue")
indexcolor = colors.index("green")
print("Index of 'green' in colors:", indexcolor)
```

```
Index of 'green' in colors: 1
```

```
# 4. Given the tuple 'values = (1, 2, 3, 1, 2, 1)', use the '.count()' method to count how many times 1 appears in the tuple and print th
values = (1, 2, 3, 1, 2, 1)
count_one = values.count(1)
print("Number of 1 in values", count_one)
```

```
Number of 1 in values 3
```

```
# 5. Create a tuple 'person' containing '("John", 25, "Engineer")'. Unpack the tuple into three variables 'name', 'age', and 'professior
person = ("John", 25, "Engineer")
name, age, profession = person
print("Name:", name)
print("Age:", age)
print("Profession:", profession)
```

```
Name: John
Age: 25
Profession: Engineer
```

```
# 6. Given the tuple 'numbers = (1, 2, 3, 4, 5)', convert it to a list, append the number 6, and then convert it back to a tuple. Print t
numbers = (1, 2, 3, 4, 5)
numbers_list = list(numbers)
numbers_list.append(6)
final_tuple = tuple(numbers_list)
print("Final tuple after appending 6:", final_tuple)
```

```
Final tuple after appending 6: (1, 2, 3, 4, 5, 6)
```

```
# 7. Create a tuple 'dimensions' containing '(1920, 1080)'. Use tuple unpacking to assign the values to variables 'width' and 'height', t
dimensions = (1920, 1080)
width, height = dimensions
print("Width:", width)
print("Height:", height)
```

```
Width: 1920
Height: 1080
```

```
# 8. Given the tuple 'data = (10, 20, 30, 40, 50)', create a new tuple 'even_data' that contains only the even numbers from 'data' using
data = (10, 20, 30, 40, 50)
even_data = tuple(x for x in data if x % 2 == 0)
print("Even numbers from data:", even_data)
```

```
Even numbers from data: (10, 20, 30, 40, 50)
```

```
# 9. Create a tuple 'letters' with the elements '('a', 'b', 'c', 'd')'. Convert the tuple to a string by joining the elements with a comm
letters = ('a', 'b', 'c', 'd')
letters_string = ",".join(letters)
print("Joined letters string:", letters_string)
```

```
Joined letters string: a,b,c,d
```

```
# 10. Given the tuple 'records = ((1, "Alice"), (2, "Bob"), (3, "Charlie"))', iterate over the tuple and print each ID and name on a nev
records = ((1, "Alice"), (2, "Bob"), (3, "Charlie"))
for record in records:
    id, name = record
    print(f"ID: {id}, Name: {name}")
```

```
ID: 1, Name: Alice
ID: 2, Name: Bob
ID: 3, Name: Charlie
```

Start coding or generate with AI.