Task 3

Exercise 2.1 and 2.2: Simple messages.

In [ ]:

```
message="Hello Bytwise Fellowship!"
print(message)
```

```
Hello Bytwise Fellowship!
```

In [ ]:

```
message="Hello Bytwise Fellowship!"
print(message)
message="My name is Muid."
print(message)
```

```
Hello Bytwise Fellowship!
My name is Muid.
```

Task 2.3: Personal Message.

In [ ]:

```
name="Eric"
message="Hello " + name + ", would you like to learn some Python today?"
print(message)
```

```
Hello Eric, would you like to learn some Python today?
```

Task 2.4: Name Cases.

In [ ]:

```
name="Eric"
print(name.lower())
print(name.upper())
print(name.title())
```

```
eric
ERIC
Eric
```

Task 2.5: Famous Quote.

In [ ]:

```
name="Albert Einstein"
message='"A person who never made a mistake never tried anything new."'
print(name + " once said, " + message)
```

```
Albert Einstein once said, "A person who never made a mistake never tried anything
new."
```

Task 2.6: Famous Quote 2.

In [ ]:

```
famous_person="Albert Einstein"
message='"A person who never made a mistake never tried anything new."'
print(famous_person + " once said, " + message)
```

```
Albert Einstein once said, "A person who never made a mistake never tried anything
new."
```

Task 2.7: Stripping Names.

In [ ]:

```python
name=" \tMuid \t\n"
print(name.lstrip())
print(name.rstrip())
print(name.strip())
```

```
Muid 	

        Muid
Muid
```

Task 2.8: Number Eight.

In [ ]:

```python
print(5+3)
print(2*4)
print(16/2)
print(10-2)
```

```
8
8
8.0
8
```

Task 2.9: Favorite Number.

In [ ]:

```python
favorite_number=16
message="My favorite number is " + str(favorite_number) + "."
print(message)
```

```
My favorite number is 16.
```

Task 2.10: Adding Comments.

In [ ]:

```python
#assigning favorite number to a variable
favorite_number = 16
#assigning a message to a variable
message = "My favorite number is " + str(favorite_number) + "."
#printing the message
print(message)
```

```
My favorite number is 16.
```

Task 2.11: Zen of Python.

In [ ]:

```python
import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```