QUIZ

Complete the code to create a Flask application object.

```python
from flask import Flask
app = Flask( __name__ )
```
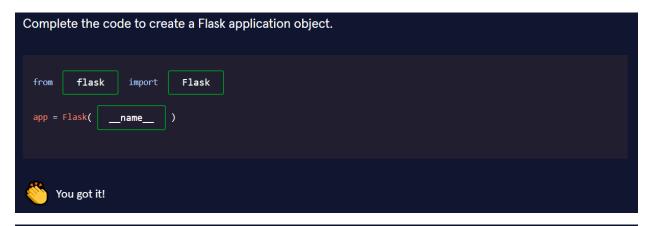
👏 You got it!

Complete the code to have the webpage display a greeting when the user visits port **5000** of the **localhost** during development.

```python
@app.route ( '/' )
def home():
    return 'Welcome!'
```

👏 You got it!

What will be the result of the following code?

```python
@app.route('/home')
@app.route('/about')
def greeting():
    return 'Hello, World!'
```

Hello, World! will be displayed at both URL paths (**/home** and **/about**).

👏 Correct! Multiple URL paths can be bound to a single view function.

Which of these is NOT true about Flask?

Flask can be used to create static web pages.

Flask can be run with any WSGI-compatible server.

Flask comes with a database abstraction layer and form validation.

👏 Well done! Flask is a microframework that comes with minimal built-in features. Instead, the developer has the freedom to choose which additional functionalities to add.

Flask is a Python framework for developing web applications.

Which of the following code will display `305.65K` on the webpage at the `/convert/celsius_to_kelvin/32.5` URL?

```python
@app.route('/convert/celsius_to_kelvin/<celsius>')
def celsius_to_kelvin(celsius):
    return f'{celsius + 273.15}K'
```

```python
@app.route('/convert/celsius_to_kelvin/<int:celsius>')
def celsius_to_kelvin(celsius):
    return f'{celsius + 273.15}K'
```

```python
@app.route('/convert/celsius_to_kelvin/<float:celsius>')
def celsius_to_kelvin(celsius):
    return f'{celsius + 273.15}K'
```

👏 You got it!

What will be displayed on the webpage at the / URL?

```python
@app.route('/')
def home():
    return '<h1>My First Flask App</h1>'
```

`'<h1>My First Flask App</h1>'`

An error will be thrown.

`<h1>My First Flask App</h1>`

`My First Flask App`

👏 Correct! The HTML will be rendered and `My First Flask App` will be displayed as a level-1 heading.

Which of the following code will display `Posts tagged with X` when the user visits `/tagged/X`, where `X` could be any tag.

```python
@app.route('/tagged/tag_name)
def tagged(tag_name):
    return f'Posts tagged with {tag_name}'
```

```python
@app.route('/tagged/<tag_name>)
def tagged():
    return f'Posts tagged with {tag_name}'
```

```python
@app.route('/tagged/{tag_name}')
def tagged(tag_name):
    return f'Posts tagged with {tag_name}'
```

```python
@app.route('/tagged/<tag_name>')
def tagged(tag_name):
    return f'Posts tagged with {tag_name}'
```

👏 You got it!