1. To add a URL pattern with regex, you use the `re_path()` function instead of the `path()` function.

1 / 1 point

- ⦿ True
- ○ False

✓ **Correct**
Correct! If the path converter needs a complex matching pattern, you should use the `re_path()` function. Here, "re" stands for regex.

2. Which of the following sentences about the `path()` function is correct? Select all that apply.

1 / 1 point

☐ The URL string parameter of the `path()` function captures query parameters from the URL.

☑ The `path()` function is used to define a URL pattern.

✓ **Correct**
Correct! It adds a URL mapped to a view to the urlpatterns list.

☐ The `path()` function returns the path of the Django app.

☑ The `path()` function is defined in the `django.urls` module.

✓ **Correct**
Correct! The path and include functions, used to build the URL patterns list, are defined in the `django.urls` module.

3. Complete the sentence. The path converters capture _____ from the URL.

1 / 1 point

- ◉ Path parameters
- ○ Body parameters.
- ○ Query parameters
- ○ URL parameters

  ✓ **Correct**
  Correct! The converters of the format `<type:variable>` mentioned in the URL string argument to the `path()` function hold the parameters included in the URL.

4. The `request.user` attribute contains the information of the current user.

1 / 1 point

- ◉ True
- ○ False

  ✓ **Correct**
  Correct! The view function can access the information about the current user – such as the username and whether it is authenticated - with the `request.user` attribute.

5. Complete the following sentence. The HTTP status code starting with 5 implies that:

1 / 1 point

- ○ There is a client-side error.
- ○ The action has been successfully completed.
- ○ The request has been received and is under process.
- ◉ The server has encountered an error.

  ✓ **Correct**
  Correct! For a server-side error, the status code starts with 5.

6. What are the important features of a class-based view? Select all that apply.

☑ The `as_view()` method maps a URL to a class-based view.

✓ **Correct**
Correct! This method connects a view class with a URL string pattern.

☑ A class-based view implements different methods for each HTTP method.

✓ **Correct**
Correct! The user-defined view class overrides the `get()` and `post()` methods to define processing logic for corresponding request methods.

☑ A class-based view subclasses the `django.view.View` base class.

✓ **Correct**
Correct! All view classes inherit the `django.view.View` class.

☑ Class-based views are reusable.

✓ **Correct**
Correct! Python's principle of multiple inheritances makes Django's class-based views reusable.

7. The `Http404` response is a convenient alternative for an `HttpResponse`.

◉ True.

○ False

✓ **Correct**
Correct! It is a subclass of `HttpResponse` to have a consistent 404 error page across different pages in the application.

8. Complete the following sentence. The URL name is _____.

0 / 1 point

Select all that apply.

☑ passed as the `name` parameter in the `path()` function.

> ⊘ **Correct**
> Correct! The `path()` has an optional `name` parameter in addition to the URL pattern string and the view function.

☐ an optional parameter passed inside the `path()` function.

☑ used to define URL namespace.

> ⊘ **Correct**
> Correct! You can obtain the URL with syntax like `reverse(namespace:view)`.

☑ used by the `reverse()` function to fetch the URL mapped with the view function.

> ⊘ **Correct**
> Correct! The `reverse()` function is defined in the `django.urls` module obtains the URL mapped with the view function.

> You didn't select all the correct answers

9. Can you define views in the `views.py` file in the projects folder?

1 / 1 point

◉ Yes

○ No

> ⊘ **Correct**
> Correct! You can define views in the `views.py` file in the projects folder. This is used when you want to override the default error views.

10. Complete the following sentence. To override the default error view, _____.

1 / 1 point

Select all that apply.

☑ specify the appropriate handler in the project's `URLConf`.

> ⊘ **Correct**
> Correct! There are predefined handlers for customizing error views, such as `Handler404 for page_not_found()`.

☑ define the custom view in the project folder.

> ⊘ **Correct**
> Correct! The handler refers to the view function defined in the `view.py` file under the project folder.

☐ there's no need to override the default error views.

☐ you should define the custom error handler view in the app's `views.py` file.