url Tag

{% url %} - It returns an absolute path reference (a URL without the domain name) matching a given view and optional parameters. Any special characters in the resulting path will be encoded using iri_to_uri().

Syntax:-

{% url 'urlname' %}

{% url 'urlname' as var %}

{% url 'urlname' arg1=value1 arg2=value2 %}

{% url 'urlname' value1 value2 %}

path ( )

path(route, view, kwargs=None, name=None) - It returns an element for inclusion in urlpatterns.

Where,

- The route argument should be a string or gettext_lazy() that contains a URL pattern. The string may contain angle brackets e.g. <username> to capture part of the URL and send it as a keyword argument to the view. The angle brackets may include a converter specification like the int part of <int:id> which limits the characters matched and may also change the type of the variable passed to the view. For example, <int:id> matches a string of decimal digits and converts the value to an int.
- The view argument is a view function or the result of as_view() for class-based views. It can also be an django.urls.**include()**.
- The kwargs argument allows you to pass additional arguments to the view function or method. It should be a dictionary.
- name is used to perform URL reversing.

path ( )

urls.py

```
urlpatterns = [
      path(route, view, kwargs=None, name=None)
     ]
```

urls.py

```
urlpatterns = [
     path(learndj/', views.learn_Django, {'check': 'OK'}, name='learn_django'),
     ]
```

```
urlpatterns = [
    path('about/', views.about),
]
```

```
<a href="/about">About</a>
<a href="{{ab}}">About</a>
```

```
def about(request):
 return render(request, 'core/about.html', {'ab':'/about'})
```

```
<a href="{% url 'aboutus' %}">About</a>
```

```
{% url 'aboutus' as abc %}
<a href="{{abc}}">About</a>
urlpatterns = [ path('about/', views.about, name='aboutus'),]
```