Geek University

Evolua seu lado geek!

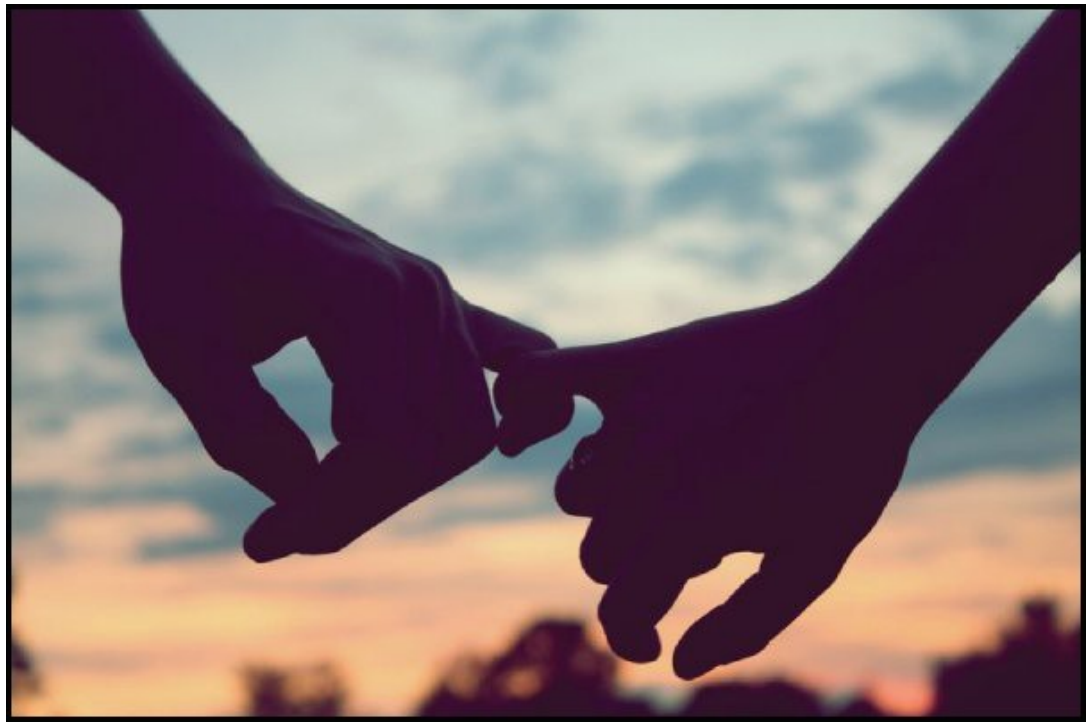
www.geekuniversity.com.br

Conhecendo as Class Based Views

Promessa

Antes de iniciar a aula, faça uma promessa:

“Após esta aula, eu nunca mais irei criar views no Django se não for usando as Class Based Views."



Conhecendo as Class Based Views

Até então criamos views aqui no Django usando Function Based Views

Mas você entendeu como isso funciona?

Conhecendo as Class Based Views

Até então criamos views aqui no Django usando Function Based Views

Mas você entendeu como isso funciona?

```
from .views import index

urlpatterns = [
    path('', index, name='index'),
]
```

```
def index(request):
    context = {
        'produtos': Produto.objects.all()
    }
    return render(request, 'index.html', context)
```

Conhecendo as Class Based Views

Até então criamos views aqui no Django usando Function Based Views

Mas você entendeu como isso funciona?

```
from .views import index

urlpatterns = [
    path('', index, name='index'),
]
```

```
def index(request):
    context = {
        'produtos': Produto.objects.all()
    }
    return render(request, 'index.html', context)
```

Uma view Django nada mais é do que uma função que:

- Recebe uma requisição (request) HTTP como entrada;
- Transforma em uma resposta (response) HTTP;

Conhecendo as Class Based Views

Até então criamos views aqui no Django usando Function Based Views

Mas você entendeu como isso funciona?

```python
from .views import index

urlpatterns = [
    path('', index, name='index'),
]
```

```python
def index(request):
    context = {
        'produtos': Produto.objects.all()
    }
    return render(request, 'index.html', context)
```

Django Function Based View

Conhecendo as Class Based Views

Então o que seria uma Class Based View?

Conhecendo as Class Based Views

Então o que seria uma Class Based View?

- Views Django baseadas em classes;
- Sendo que estas classes são executadas como funções;
- Usando de recursos chamados de mixins para adicionar funcionalidades;

Conhecendo as Class Based Views

Como se parece uma Class Based View?

Conhecendo as Class Based Views

Como se parece uma Class Based View?

```
urlspatterns = [
    path('', IndexView.as_view(), name='index'),
]
```

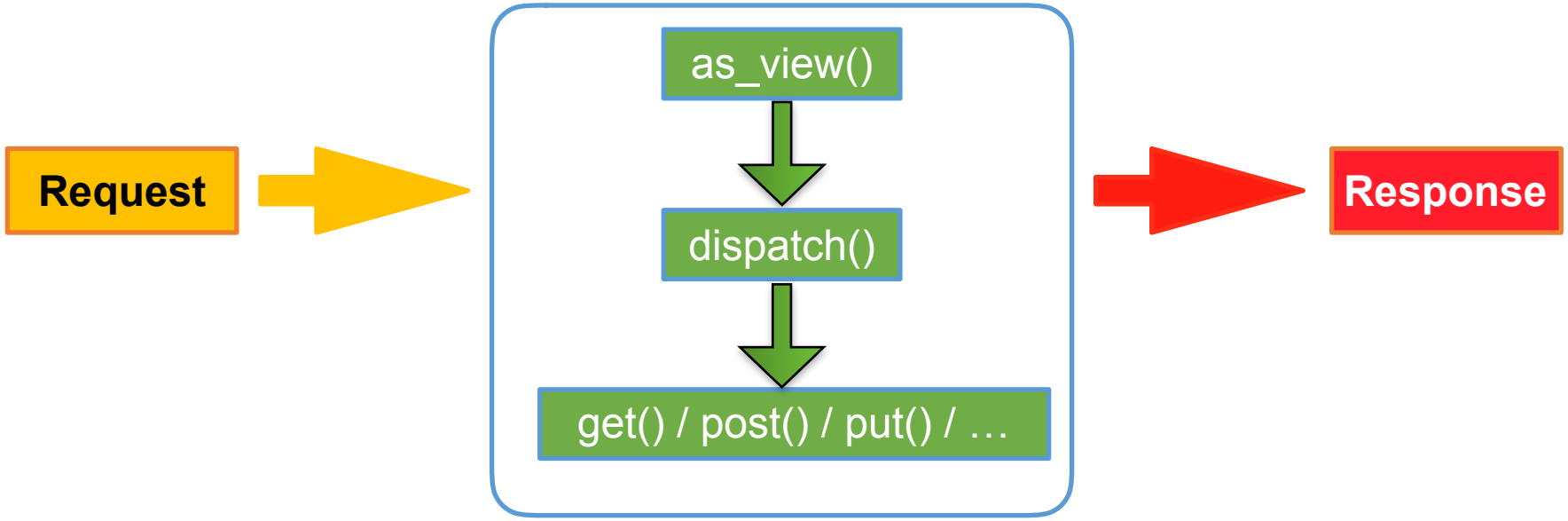
```
class IndexView(View):

    def get(self, request):
        return render(request, 'index.html')
```

Conhecendo as Class Based Views

Como funcionam as Class Based View?

Conhecendo as Class Based Views

Como funcionam as Class Based View?



Django Class Based View

Conhecendo as Class Based Views

O que são Mixins?

Conhecendo as Class Based Views

O que são Mixins?

Conhecendo as Class Based Views

O que são Mixins?

Sabores

\+

Adicionais

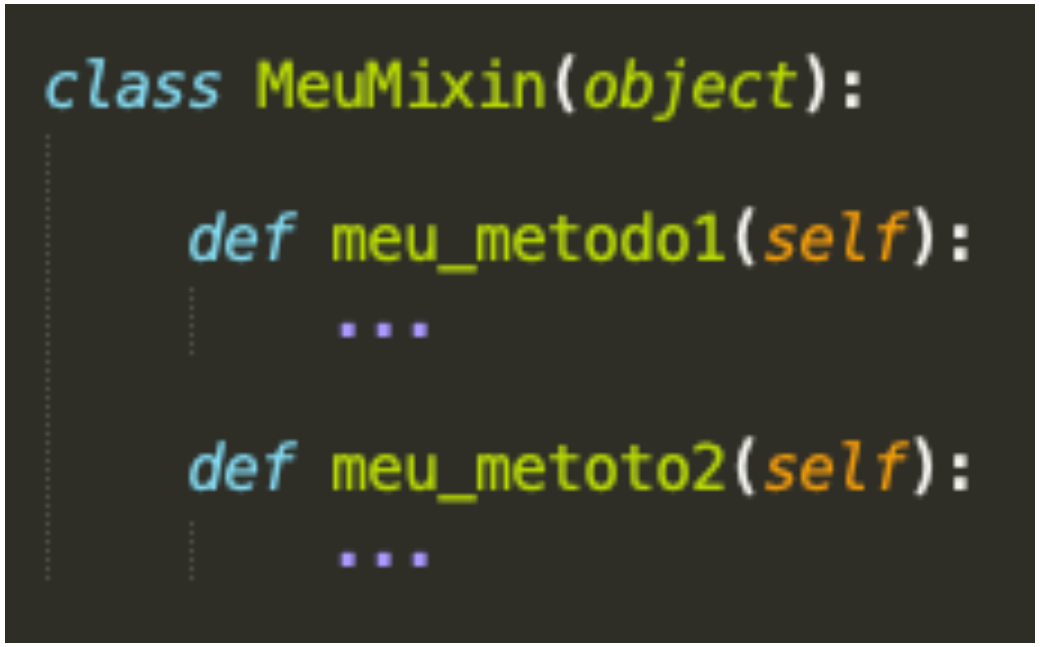
=

Mixins

Conhecendo as Class Based Views

O que são Mixins?

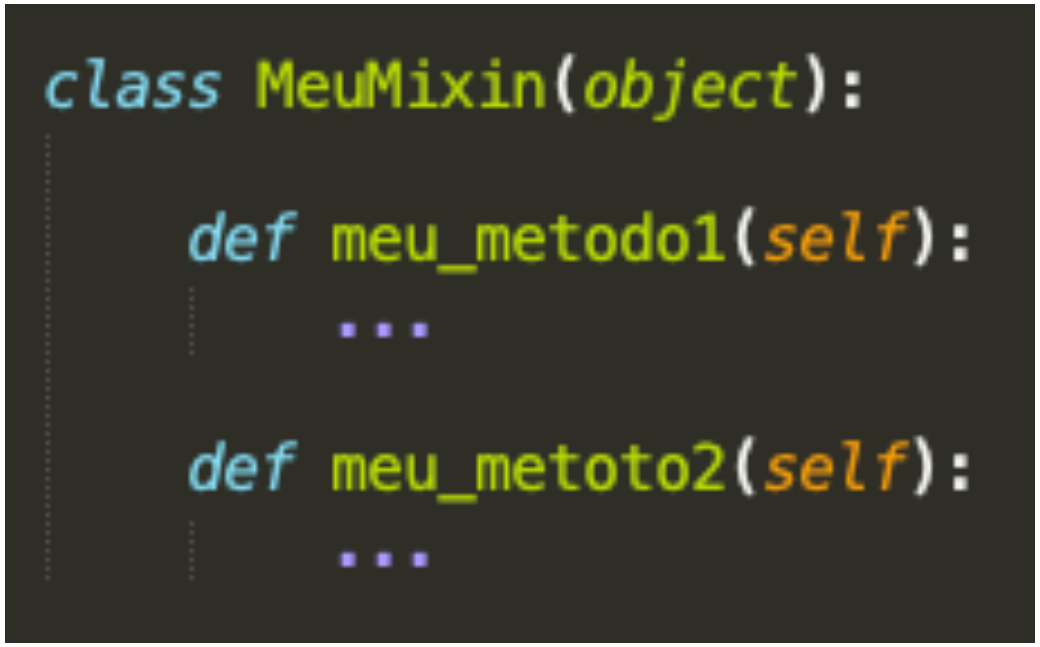
Mixins nada mais são do que classes Python que proveem alguma funcionalidade para ser herdada por outras classes mas que não são instanciadas por si so.



Conhecendo as Class Based Views

O que são Mixins?

Mixins nada mais são do que classes Python que proveem alguma funcionalidade para ser herdada por outras classes mas que não são instanciadas por si so.



Qual o objetivo?

- Adicionar funcionalidade para classes;
- Melhorar a modularidade;

Conhecendo as Class Based Views

Quando usar mixins?

Quando se quer reutilizar código/funcionalidades entre múltiplas classes.

Conhecendo as Class Based Views

Regra prática!

- As classes view base providas pelo Django sempre são posicionadas à direita.
- Mixins sempre são posicionadas à esquerda da classe view base.
- Mixins sempre devem herdar de object do Python.

Conhecendo as Class Based Views

As classes view base providas pelo Django sempre são posicionadas à direita.

```
class TemplateView(TemplateResponseMixin, ContextMixin, View):
    """
    A view that renders template. This view will also pass into the context
    any keyword argument passed bu the URLconf.
    """
    def get(self, request, *args, **kwargs):
        context = self.get_context_data(**kwargs)
        return self.render_to_response(context)
```

Conhecendo as Class Based Views

Mixins sempre são posicionadas à esquerda da classe view base.

```
class TemplateView(TemplateResponseMixin, ContextMixin, View):
    """
    A view that renders template. This view will also pass into the context
    any keyword argument passed bu the URLconf.
    """
    def get(self, request, *args, **kwargs):
        context = self.get_context_data(**kwargs)
        return self.render_to_response(context)
```

Conhecendo as Class Based Views

Mixins sempre devem herdar de object do Python.

```python
class ContextMixin(object):
    """
    A default context mixin that passes keywork arguments received by
    get_context_data as the template context.
    """

    def get_context_data(self, **kwargs):
        if 'view' not in kwargs:
            kwargs['view'] = self
        return kwargs
```

Conhecendo as Class Based Views

Class Based Views integradas do Django

Conhecendo as Class Based Views

Class Based Views integradas do Django

Temos 4 categorias principais:

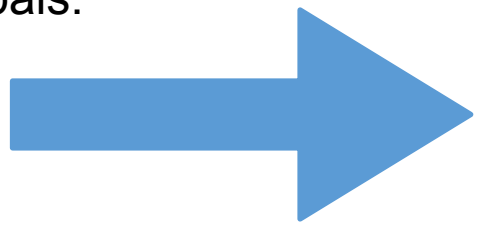
- Base
- List
- Detail
- Edit

Conhecendo as Class Based Views

Class Based Views integradas do Django

Temos 4 categorias principais:

- Base Generic Views



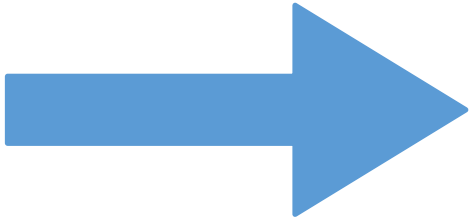
- django.views.generic.View
- django.views.generic.TemplateView
- django.views.generic.RedirectView

Conhecendo as Class Based Views

Class Based Views integradas do Django

Temos 4 categorias principais:

- List Generic Views

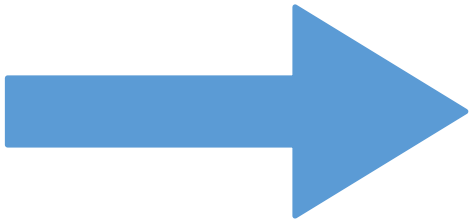


- django.views.generic.list.ListView

Conhecendo as Class Based Views

Class Based Views integradas do Django

Temos 4 categorias principais:

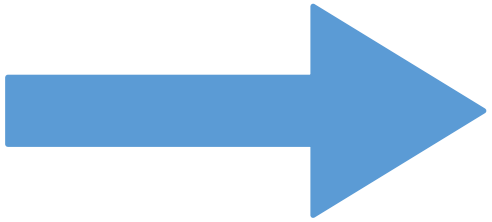
- Detail Generic Views  - django.views.generic.detail.DetailView

Conhecendo as Class Based Views

Class Based Views integradas do Django

Temos 4 categorias principais:

- Edit Generic Views



- django.views.generic.FormView
- django.views.generic.CreateView
- django.views.generic.UpdateView
- django.views.generic.DeleteView

Conhecendo as Class Based Views

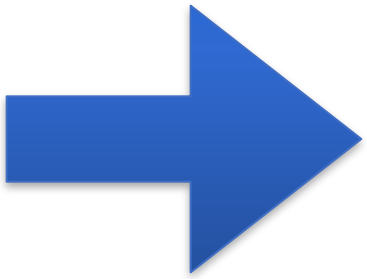
Uso de Class Based Views

Conhecendo as Class Based Views

Uso de Class Based Views

```python
from django.shortcuts import render


def index(request):
    return render(request, 'index.html')
```

Django Function Based View



```python
from django.views.generic import TemplateView


class IndexView(TemplateView):
    template_name = 'index.html'
```

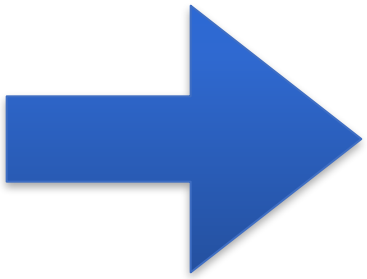
Django Class Based View

Conhecendo as Class Based Views

Uso de Class Based Views

```python
def index(request):
    context = {
        'produtos': Produto.objects.all()
    }
    return render(request, 'index.html', context)
```

Django Function Based View



```python
from django.views.generic import View
from django.shortcuts import render

from .models import Produto


class IndexView(View)

    def get(self, request):
        context = {
            'produtos': Produto.objects.all()
        }
        return render(request, 'index.html', context)
```

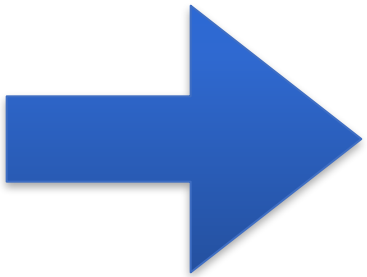
Django Class Based View

Conhecendo as Class Based Views

Uso de Class Based Views

```python
from django.shortcuts import render
from django.urls import reverse_lazy

from .models import Produto
from .forms import ProdutoForm

def produto(request):
    form = ProdutoForm(request.POST or None)
    if str(request.method) == 'POST':
        if form.is_valid():
            form.save()
            form = ProdutoForm()
            return reverse_lazy('produto')
    context = {
        'form': form
    }
    return render(request, 'produto.html', context)
```

Django Function Based View



```python
from django.views.generic import CreateView
from django.shortcuts import render

from .models import Produto
from .forms import ProdutoForm


class ProdutoView(CreateView)
    model = Produto
    form_class = ProdutoForm
    success_url = reverse_lazy('produto')
```

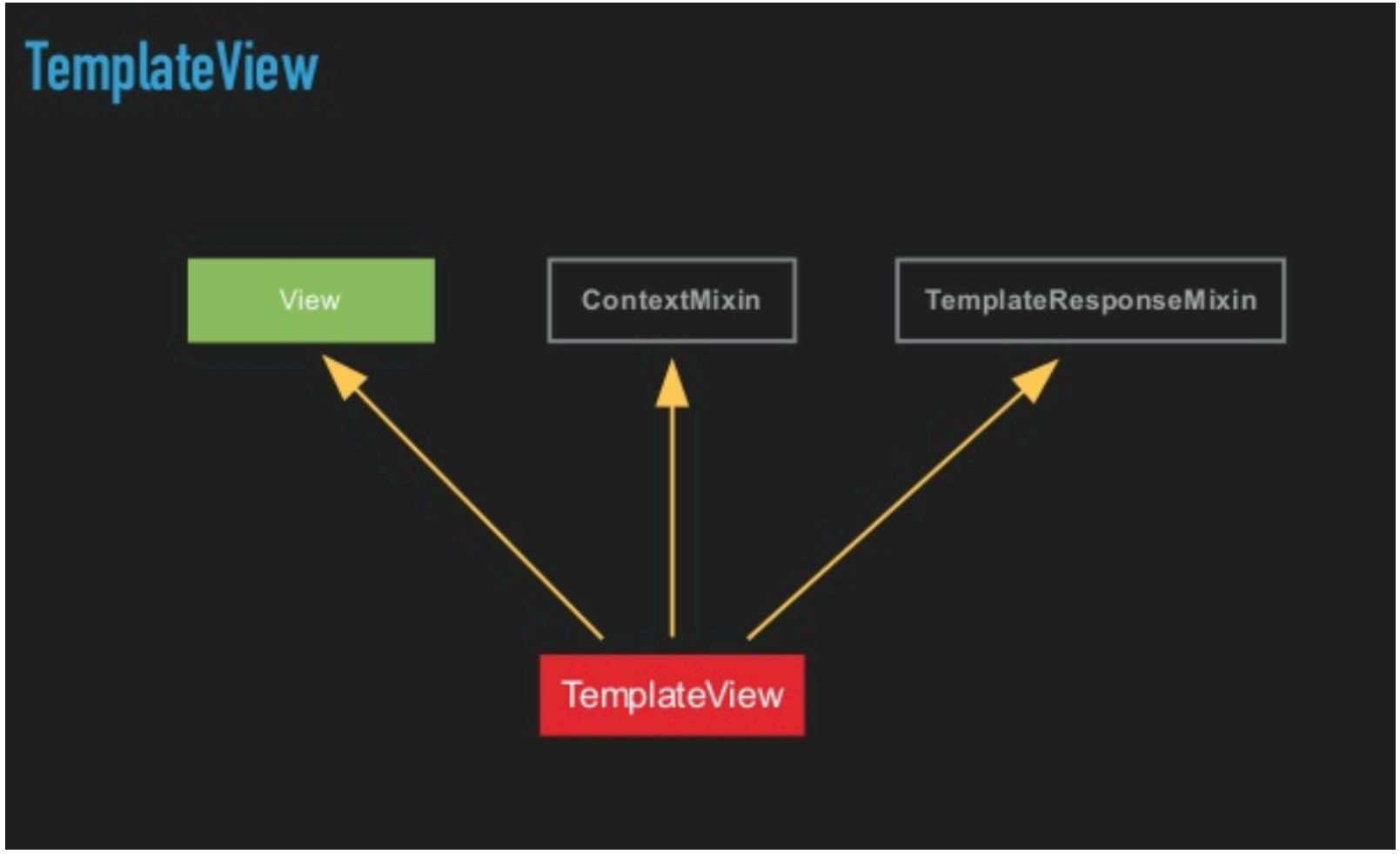
Django Class Based View

Conhecendo as Class Based Views

Class Based Views e Mixins

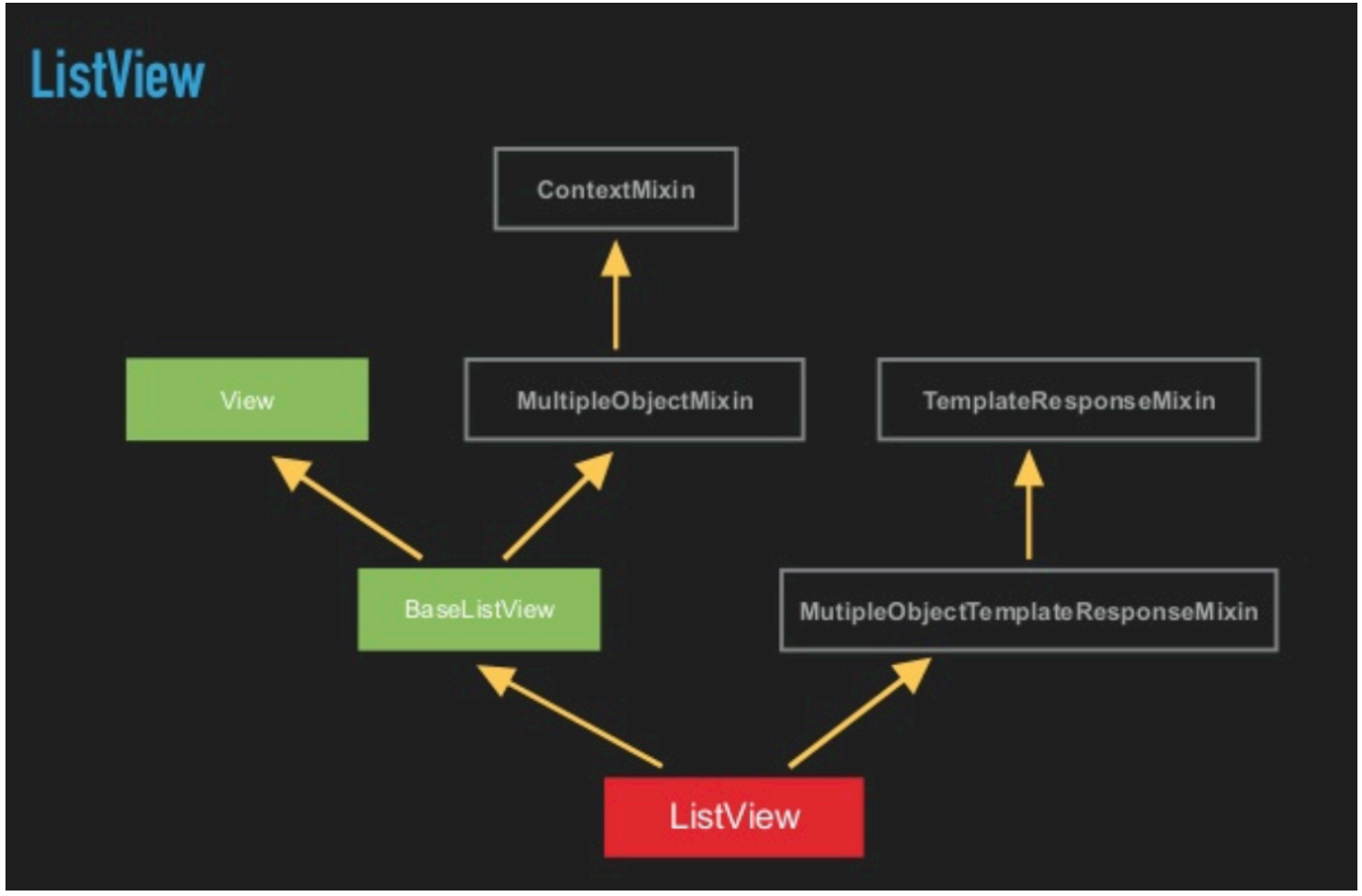
Conhecendo as Class Based Views

Class Based Views e Mixins



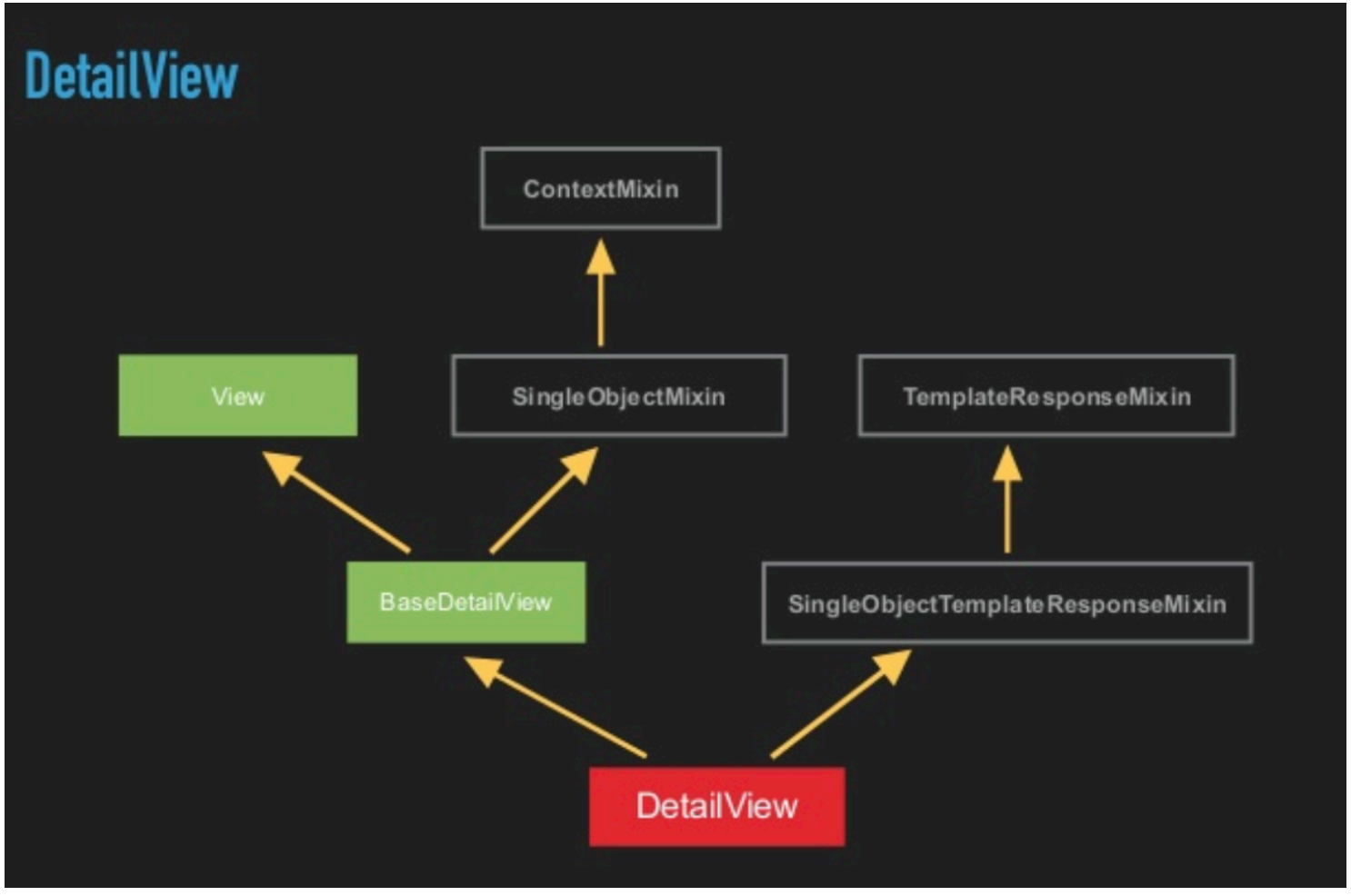
Conhecendo as Class Based Views

Class Based Views e Mixins



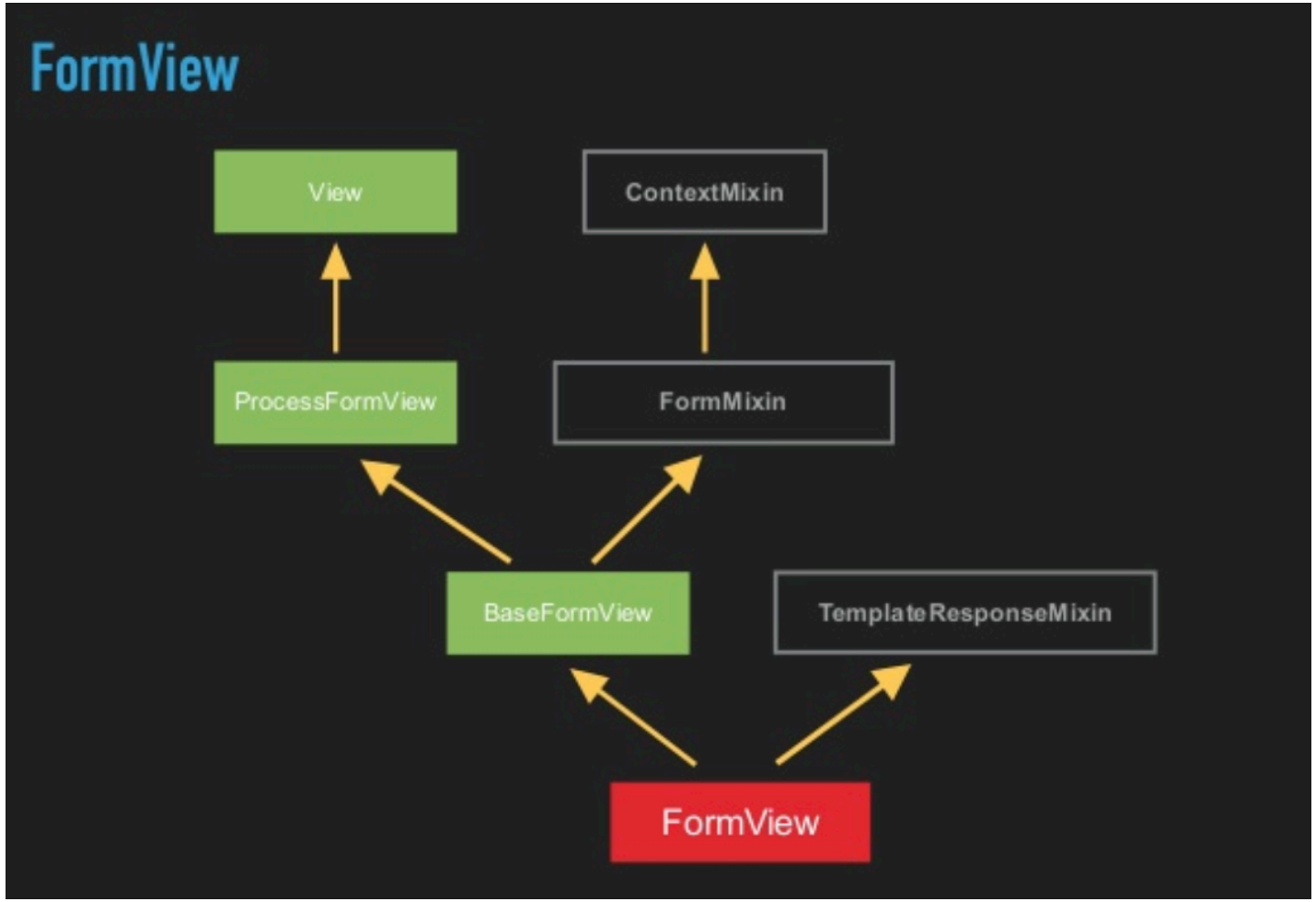
Conhecendo as Class Based Views

Class Based Views e Mixins



Conhecendo as Class Based Views

Class Based Views e Mixins



Conhecendo as Class Based Views

O lado sombrio das Class Based Views

Conhecendo as Class Based Views

O lado sombrio das Class Based Views

- É muito fácil você perder o caminho da origem dos seus métodos.
- Você pode poluir seu código com os diversos imports.
- O fluxo de controle é totalmente escondido.
- A ordem de execução dos métodos podem não ser obvias para ninguém.
- Mais difícil debugar.
- Para entender o que está acontecendo você pode precisar ler a documentação.

Lembre-se do Zen do Python:

“Explicit is better than implicit"

Conhecendo as Class Based Views

Dicas finais

- Mantenha suas views simples.
- Não repita código nas suas views.
- Mantenha seus mixins simples.

Lembre-se do Zen do Python:

“Explicit is better than implicit"

Conhecendo as Class Based Views

Como virar um expert em Class Based View?

Django Docs:
https://docs.djangoproject.com/en/2.2/ref/class-based-views/

CBV:
https://ccbv.co.uk/

Geek University

Evolua seu lado geek!

www.geekuniversity.com.br