**Python e Django**

**Observe o seguinte trecho de código:**

**urlpatterns = [**

**path('admin/', admin.site.urls),**

**path('dragonball/', views.dragonball),**

**path('yuyuhakusho/', views.yuyuhakusho),**

**path('naruto/', views.naruto),**

**path('onepiece/', views.onepiece),**

**path('onepunchman/', views.onepunchman)**

**]**

**Considere que ao acessar pelo navegador o endereço "127.0.0.1:8000" apresente um erro escrito Page Not Found, qual inclusão na lista urlpatterns resolveria essa situação?**

**R:** path('', views.animes)

**Qual a finalidade do comando pip freeze?**

**R:** É um comando de dependência do pip install que ajuda na verificação de versões já instaladas.

**Visualize o seguinte trecho de código:**

**def chamar(request, nome, telefone):**

**frase = '<b>{}</b>'.format(nome)**

**return \_\_\_\_\_\_\_\_\_\_\_**

**Com base no trecho acima, para que as tags HTML <b>, capaz de tornar o texto em negrito, funcionem corretamente é necessário qual código?**

**R:** return HttpResponse('{}'.format(frase))

**Na aula o professor seguiu uma ordem de configuração específica para começar a trabalhar com Django, avalie as opções abaixo:**

**I - pip install django**

**II - django-admin startproject meu\_projeto**

**III - django-admin startapp meu\_app**

**IV - admin-django startproject meu\_projeto**

**V - admin-django startapp meu\_app**

**Qual foi a ordem correta?**

**R:** I, II e III.

**Leia as sentenças e assinale a alternativa correta. I – Sistemas web são mais rápidos e trabalhosos. II – São hospedados em servidores especializados, por isso, são considerados seguros. III – Dependendo das funcionalidades, o sistema web pode detectar erros automaticamente.**

**R:** Apenas I está errada.

**Visualize o seguinte trecho de código:**

**def identidade(request, nome, rg, cpf):**

**Para essa função ser executada por uma chamada no navegador, qual o código correto para essa finalidade?**

**R:** path('iden/<nome>/<rg>/<cpf>/', views.identidade)

**De acordo com as definições do professor, quais das alternativas abaixo não é uma característica das aplicações web?**

**R:** Promovem grande economia, pois os gastos com infraestrutura são baixos.

**Na aula foram apresentadas as relações entre os diversos arquivos Python da estrutura do Django. Sobre isso responda:**

**R:** O arquivo views.py possui funções que são disparadas conforme as chamadas configuradas no arquivo urls.py na lista urlpatterns.

**A respeito da framework Django, é incorreto afirmar:**

**R:** Django é uma ferramenta paga e open source.

**Visualize o seguinte trecho de código:**

**urlpatterns = [**

**path('admin/', admin.site.urls),**

**path('admin/<string:filme>/<horario>/<int:poltrona>/', views.reservar)**

**]**

**Em relação ao código acima é correto responder:**

**R:** O parâmetro filme e horário serão tratados como strings, enquanto que o parâmetro poltrona será tratado como int.

--------------------------------------------------------------------------------------------

**Introdução a Banco de Dados**

**Para criação de uma tabela com nome personalizado, é necessário a inserção do seguinte trecho de código dentro da classe da tabela:**

**class Meta:**

**db\_table = 'bicicletas'**

**A seguir, ao executar o comando "python manage.py makemigrations core" e gerar o arquivo 0001\_initial.py, será possível observar, por exemplo, o seguinte trecho de código dentro desse novo arquivo:**

**R:** options = {'db\_table' : 'bicicletas',}

**Ao acessar o endereço "127.0.0.1:8000/admin/" e visualizar no navegador uma tela de erro do tipo "no such table: django\_session", qual o comando foi usado pelo professor para resolver esse esse?**

**R:** python manage.py migrate

**Observe o seguinte trecho de código:**

**class TabelaPeriodica(models.Model):**

**nome = models.CharField(max\_length=100)**

**simbolo = models.CharField(max\_length=2)**

**peso = models.CharField(max\_length=100)**

**data\_descoberta = models.DateTimeField()**

**comentario = models.TextField(blank=True, null=True)**

**Considere que o aplicativo foi criado com o nome "core" e após executar o comando "python manage.py makemigrations core" e "python manage.py sqlmigrate core 0001", o sistema iria:**

**I - Criar uma tabela com o nome core\_evento**

**II - Criar uma tabela com o nome core\_TabelaPeriodica**

**III - Criar o campo id automaticamente**

**IV - Todos os campos aceitam valores nulos.**

**V - Nenhum campo aceita valores vazios.**

**Estão corretas as afirmativas:**

**R:** Apenas II, III e V.

**Observe o seguinte comando executado no terminal:**

**python manage.py sqlmigrate core 0001**

**O retorno desse comando apresentou:**

**CREATE TABLE "vertebrados" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "grupo" varchar(100) NOT NULL, "sistema\_digestorio" text NULL, "sistema\_circulatorio" text NULL, "sistema\_respiratorio" text NULL);**

**Com relação ao código acima, observe as afirmativas abaixo:**

**I - A tabela contém 3 colunas.**

**II - Apenas a coluna "grupo" e "id" são campos obrigatórios, apesar de "id" ser "autoincrement"**

**III - ('grupo', models.CharField(max\_length=100)) reflete o estado da criação da coluna "grupo"**

**IV - ('sistema\_respiratorio', models.TextField()) reflete o estado da criação da coluna "sistema\_respiratorio"**

**Estão corretas as afirmativas:**

**R:** Apenas II e III.

**Observe o seguinte trecho de código abaixo:**

**class timesComMundial(models.Model):**

**nome = models.CharField(max\_length=100)**

**pais = models.CharField(max\_length=100, blank=True, null=True)**

**qtdTitulos = models.CharField(max\_length=10)**

**ultimoConquistado = models.DateTimeField()**

**Com relação ao código acima, é correto dizer que:**

**R:** Quando um time conquistar 1000 títulos, o código não suportará, pois qtdTitulos precisará ter o max\_length igual a 100.

**Durante a aula, o professor precisou criar uma nova coluna no banco de dados (usuario) em uma tabela que já existia. Essa coluna recebeu a seguinte instrução na classe:**

**usuario = models.ForeignKey(User, on\_delete=models.CASCADE)**

**Após alguns procedimentos, ele executou o seguinte comando:**

**python manage.py sqlmigrate core 0002**

**Considerando o contexto da aula, a inclusão da nova coluna seguiu qual sequência de ações?**

**R:** A tabela pré-existente foi renomeada com o sufixo "\_old", a seguir uma nova tabela com o nome original foi criada, com isso o conteúdo da tabela pré-existente foi inserido na nova tabela e, finalizando, a tabela pré-existente foi excluída.

**Considere o seguinte trecho de código:**

**class DestaquesDaCiencia:**

**cientista = models.CharField(max\_length=100)**

**data = models.DateTimeField()**

**descricao = models.TextField()**

**antecedentes = models.TextField(blank=True, null=True)**

**contradicao = models.TextField()**

**relevancia = models.CharField(max\_length=3)**

**Com base nessa classe mediante a criação de uma tabela com esse trecho de código, avalie as afirmativas abaixo:**

**R:** Na tela de navegação do admin, para criar filtros por cientista ou por data é necessário o comando "list\_filter = ('cientista','data',)".

**Observe o seguinte trecho de código:**

**class Lars\_Von\_Trier:**

**titulo\_portugues = models.CharField(max\_length=100)**

**titulo\_original = models.CharField(max\_length=100)**

**data\_lancamento = models.DateTimeField()**

**nota\_imdb = models.CharField(max\_length=100)**

**genero = models.TextField()**

**sinopse = models.TextField()**

**critica = models.TextField()**

**Considerando que a tabela já exista e pelo admin do site você precisa registrar o último filme dele com a seguinte instrução:**

**O filme de título em português "A Casa que Jack Construiu", originalmente "The House That Jack Built", lançado no Brasil em 1 de novembro de 2018, tem a nota IMDB de 7,0/10. A sinopse é sobre um filme que retrata 12 anos de um serial killer, portanto é do gênero crime, drama e horror. As críticas do filme são muito boas, mas não devem ser registradas.**

**Sendo assim, avalie as seguintes afirmativas com relação ao exigido acima:**

**I - Para que o registro na tela do admin apareça com o título do filme em português, é necessário criar "def \_\_str\_\_(self):" e dentro dele "return self.titulo\_portugues".**

**II - O campo "critica" requer parâmetros blank=True e null=True.**

**III - list\_display = ('titulo\_portugues', 'data\_lancamento', 'genero') pode ser usado para que a tela do admin mostre apenas os campos citados entre parênteses.**

**Estão corretas as afirmativas:**

**R: I II III**

**Observe o seguinte trecho de código:**

**class BandasDeRock(models.Model):**

**nome = models.TextField(blank=True, null=True)**

**vocalista = models.TextField(blank=True, null=True)**

**guitarrista = models.TextField(blank=True, null=True)**

**guitarrista2 = models.TextField(blank=True, null=True)**

**guitarrista3 = models.TextField(blank=True, null=True)**

**baixista = models.TextField(blank=True, null=True)**

**tecladista = models.TextField(blank=True, null=True)**

**baterista = models.TextField(blank=True, null=True)**

**data\_criacao = models.DateTimeField()**

**class Meta:**

**db\_table = 'evento'**

**Após executar o comando "python manage.py sqlmigrate core 0001" aparecerá o seguinte trecho de código:**

**CREATE TABLE "\_\_\_\_\_\_\_" ......**

**Qual das opções abaixo preenche corretamente a lacuna acima?**

**R:** BandasDeRock

**Para se criar um usuário administrador é usado o seguinte comando básico:**

**python manage.py createsuperuser**

**Dessa forma o Python irá exigir que seja preenchido username, email e password. Porém, o professor encurtou esse processo já fornecendo um parâmetro para o username, restando configurar apenas email e password. Considerando que queremos um usuário com o nome thanos, qual opção abaixo é a correta?**

**R:** python manage.py createsuperuser --username thanos

--------------------------------------------------------------------------------------------

**Páginas de listagem**

**O que um template de Django pode conter?**

**R:** Há mais de duas alternativas corretas.

**Durante a aula foi utilizado a obtenção de dados filtrando por usuário autenticado. Qual das alternativas abaixo contém a correta combinação de comandos? (Cada comando separado por ";")**

**R:** artista = request.user ; obras = Obra.objects.filter(artista=artista)

**Você e uma amiga estão pensando em montar um protótipo de máquina do tempo. Ela começou o projeto primeiro e criou toda a estrutura em Python e te passou os elementos abaixo para você montar o painel em HTML:**

**I - {{**

**II - }}**

**III - {%**

**IV - %}**

**V - destino.get\_data**

**VI - for destino in destinos**

**VII - endfor**

**A sequência de inserção dos elementos acima, considerando repetições, é:**

**R:** III, VI, IV, I, V, II, III, VII, IV

**O professor ensinou como realizar redirecionamentos de um link para outro. Dois modos levemente diferentes foram ensinados, sendo que as opções abaixo devem ser combinadas corretamente para que funcionem, sendo elas:**

**I - path('', views.inicial)**

**II - path('', RedirectView.as\_view(url='/agenda/'))**

**III - return redirect('/agenda/')**

**IV - from django.views.generic import RedirectView**

**V - from django.shortcuts import redirect**

**R:** A combinação correta é I, III, V para um modo e II, IV para outro modo.

**Imagine que você possua um banco dados com bandas de rock e deseja imprimir na tela todas as bandas. Para isso é necessário o seguinte comando:**

**R:** bandas = Bandas.objects.all()

**Durante o começo da aula o professor ensinou a registrar os templates, portanto ele precisou entrar no arquivo settings.py, encontrar a variável TEMPLATES, nessa variável ir até a chave DIRS e preencher seu valor. Assinale abaixo a alternativa correta:**

**R:** 'DIRS' : [os.path.join(BASE\_DIR, 'templates')]

**Observe o seguinte trecho de código no arquivo "esqueleto.html":**

**<ul>**

**{% for osso in ossos %}**

**<li>{{ osso.nome }}</li>**

**{% endfor %}**

**</ul>**

**Com base no código acima é possível afirmar que:**

**I - A variável "ossos" é uma lista recebida por parâmetro na chamada do arquivo views.py**

**II - A variável osso recebe cada índice contido na variável ossos e trata cada elemento encontrado.**

**III - A expressão "osso.nome" diz que na tabela chamada "osso" existe uma coluna chamada "nome".**

**Estão corretas as afirmativas:**

**R:** Apenas I e II.

**Qual comando básico abaixo foi usado para formatar uma variável texto do tipo data?**

**R:** strftime

**Observe as seguintes descrições abaixo:**

**I - Cria um bloco denominado como "cinema"**

**II - Inclui na própria página o conteúdo de um arquivo HTML.**

**III - Inclui na própria página o conteúdo do arquivo model-page.html e substitui blocos (block) de outros arquivos HTML.**

**Respectivamente estamos falando dos comandos, separados por ";":**

**R:** {% block cinema %} ; {% include "cabecalho.html" %} ; {% extends "horarios.html" %}

**Observe o seguinte trecho de código no arquivo views.py:**

**def lista\_dinossauros(request):**

**evento = Especies.objects.get(id=1)**

**dados = {'especie' : especie}**

**return render(request, 'dinos.html', dados)**

**Com base no trecho de código acima, é possível dizer que:**

**I - O código também possui o comando "from core.models import Especies"**

**II - O html conterá, por exemplo, o comando " {{ especie.nome }} "**

**III - O html conterá, por exemplo, o comando " {{ dados.nome }} "**

**IV - O arquivo html se chama 'dinos.html'**

**Qual das afirmativas acima estão corretas?**

**R:** Apenas I, II e IV.

**Autenticação e Inserção de Dados**

**Analise o comando a seguir:**

**<form action= “submit” method=“POST”>**

**Qual a função do comando action nesse caso?,**

**R:** Concatenar a URL.

**Observe o seguinte trecho de código:**

**def lista\_cameras(request):**

**marca = Marca.objects.all()**

**dados = {'marcas' : marca}**

**return render(request, 'cameras.html', dados)**

**Para que a página "cameras.html" seja exibida apenas caso haja um usuário autenticado e, caso contrário, redirecionar para outra página adequada, é necessário, acima da função, escrever o código:**

**R:** @login\_required(login\_url='/login/')

**Qual a finalidade da função authenticate?**

**R:** Fazer a autenticação do usuário.

**Veja o código abaixo:**

**<h3> Aprenda a linguagem Django </h3>**

**<a href= “digital innovation one/”>**

**<button type= “button” > Redirecionar </button>**

**</a>**

**<ul style= “front-size: 20px”>**

**O que tem de errado no código?**

**R:** O link para o qual o botão vai levar.

**Sobre as funções decoradores, leia as sentenças e assinale a alternativa correta.**

**I – Decoradores permitem extrair um código comum que pode ser aplicado em diversas funções.**

**II – É uma forma simples de adicionar funcionalidades à funções, métodos e classes sem que o código seja alterado.**

**III – É um método que altera o código da função sem mudar seu comportamento.**

**R:** Apenas III está errada.

**Considere o seguinte trecho de código Python no arquivo views.py:**

**def submit\_login(request):**

**usuario = request.POST.get('nomeUsuario')**

**senha = request.POST.get('senhaUsuario')**

**Com base nesse trecho, assinale a alternativa correta:**

**R:** As variáveis usuario e senha estão recebendo dados de um formulário a partir de inputs com o atributo name igual a "nomeUsuario" e "senhaUsuario" respectivamente.

**Considere o seguinte trecho de código HTML:**

**<form action="submit" method="POST">**

**<label>Usuário:</label>**

**<input type="text" name="nome">**

**<label>Password:</label>**

**<input type="text" name="senha">**

**<button type="submit">Entrar</button>**

**</form>**

**Os dados enviados pelo formulário acima devem ser recuperados no arquivo views.py e podem ou não ser armazenados na memória. Avalie as opções abaixo:**

**I - request.POST.get('nome')**

**II - usuario = request.POST.get('name')**

**III - senha = request.POST.get('senha')**

**Estão corretas as alternativas:**

**R:** I, II e III.

**Observe o seguinte trecho de código:**

**usuario = authenticate(username=usuario, password=senha)**

**if usuario is not None:**

**login(request, usuario)**

**return redirect('/')**

**else:**

**messages.error(request, "Usuário ou senha inválido")**

**return redirect('/')**

**A partir do trecho acima, qual alternativa melhor corresponde ao formulário HTML que levaria os dados para esse código Python?**

**R:** O formulário HTML deve conter dois inputs, mas o atributo name pode ser qualquer um, pois as variáveis "usuario" e "senha" receberam o valor dos inputs a partir do "request.POST.get"

**Sobre o pacote de autenticação de Django, leia as sentenças e assinale a alternativa correta.**

**I – O pacote de autenticação cria permissões e tabelas de usuários.**

**II – Para ser usado, o pacote precisa estar em os aplicativos instalados na configuração.**

**III – O código dessa aplicação é: django.contrib.auth.**

**R:** Todas estão corretas.

**Analise o código abaixo.**

**<form method="post"> {% csrf\_token %}**

**Qual a função do comando csrf?**

**R:** Ele faz uma conexão para que as informações permaneçam integras até a chegada no servidor.

--------------------------------------------------------------------------------------------

**Alterando, excluindo e filtrando Dados**

**Observe o seguinte trecho de código:**

**{% for i in filosofos %}**

**<li>**

**{{ i.nome }} - {{ i.nacionalidade }}**

**( <a href="#">**

**Editar</a>**

**/ <a href="filosofos/delete/{{ i.id }}/">**

**Excluir</a>)**

**</li>**

**{% endfor %}**

**Com base no código acima, qual linha de código precisa existir no arquivo views.py para que o link "Excluir" possa ser executado adequadamente no Python?**

**R:** def apagar(request, id):

**O antigo livro conhecido como Tao Te Ching possui 81 capítulos. Por ser de origem chinesa e antigo, há diversas traduções que podem até mesmo alterar o significado da frase. Veja abaixo um trecho do capítulo 27:**

**O homem bom é modelo para o não bom**

**O homem não bom é o incentivo para o bom.**

**Mas se não reconhecemos o modelo**

**Nem cuidamos do incentivo**

**Haverá erro mesmo que haja acúmulo de conhecimento.**

**Digamos que você precisou acessar esse capítulo através de uma requisição pelo método GET, qual alternativa abaixo melhor constrói o código HTML?**

**R:** <a href="{{ livro.nome }}/?capitulo={{ livro.capitulo }}">Ler</a>

**Observe o seguinte trecho de código no arquivo views.py:**

**id\_album = request.POST.get('id\_album')**

**nome = request.POST.get('nome\_album')**

**premio = request.POST.get('premio\_album')**

**if id\_album:**

**Bandas.objects.filter(id=id\_album).update(nome=nome\_album, premio=premio\_album)**

**else**

**Bandas.objects.create(nome=nome, premio=premio\_album)**

**Com base nesse contexto, o que não é esperado conter no formulário do arquivo HTML?**

**R:** <input type="text" name="nome" size="50" value="{{ banda.nome\_album }}">

**Em uma tela de administração em HTML chamada "vitamina A", você precisa editar a descrição presente para a seguinte:**

**"A vitamina A é um micronutriente reconhecidamente relacionado às funções visuais, à integridade epitelial e ao funcionamento do sistema imunológico. Estudos indicam que a deficiência de vitamina se configura como problema de saúde pública nas regiões Nordeste, Norte e algumas áreas do Sudeste do Brasil."**

**Considere o seguinte trecho de código no arquivo views.py:**

**def vitam(request):**

**id\_v = request.GET.get('id')**

**dados = {}**

**if id\_v:**

**dados['vit'] = Vitaminas.objects.get(id=id\_v)**

**return render(request, 'vitaminas.html', dados)**

**Sendo assim, focando sobre a estrutura de formulário em HTML, qual das alternativas abaixo apresenta a solução correta que permite mostrar o texto anterior que deve ser substituído?**

**R:** <textarea name="descricao" rows="10" cols="25">{{ vit.descricao }}</textarea>

**Observe o seguinte trecho de código:**

**livros = Livros.objects.filter(autor=autor).values('titulo', 'gênero')**

**Com base no código acima e como ensinado durante a aula, o comando capaz de retornar uma lista Json através de uma requisição é:**

**R:** return JsonResponse(list(livros), safe=false)

**Observe o seguinte trecho de código:**

**@login\_required(login\_url='/login/')**

**def delete\_evento(request, id\_desenho):**

**autor = request.user**

**desenho = Desenhos.objects.get(num = id\_desenho)**

**if autor == desenho.autor:**

**desenho.delete()**

**return redirect('/')**

**O que é possível afirmar sobre o trecho de código acima e seu possível contexto?**

**I - A tabela se chama Desenhos**

**II - Existe uma coluna chamada num**

**III - O código não funcionará porque num não é nome de coluna para tabelas.**

**IV - O desenho só será excluído caso pertença ao mesmo autor da requisição de exclusão.**

**Estão corretas apenas:** I, II e IV.

**Na linguagem Python, é necessário configurar o arquivo settings.py para funcionar corretamente a busca e tratamento de dados relacionados a datas. Esse arquivo originalmente é criado com o seguinte trecho de código:**

**LANGUAGE\_CODE = 'en-us'**

**TIME\_ZONE = 'UTC'**

**USE\_TZ = True**

**Essas são as três linhas que devem, respectivamente, serem substituídas por (separados por ;)**

**R:** 'pt-br';'America/Sao\_Paulo';False

**Observe o seguinte trecho de código:**

**def lista\_eventos(request):**

**usuario = request.user**

**data\_atual = datetime.now()**

**evento = Evento.objects.filter(usuario=usuario,**

**data\_evento=data\_atual)**

**dados = {'eventos':evento)**

**return render(request, 'agenda.html', dados)**

**Para você filtrar os eventos na qual a data\_evento seja menor que a data\_atual, é necessário escrever x, caso queira eventos posteriores à data\_atual é necessário escrever y.**

**Os comandos que melhor se encaixam em x e y, respectivamente separados por ; são:**

**R:** data\_evento\_\_gt=data\_atual ; data\_evento\_\_lt=data\_atual

**Observe o seguinte trecho de código no arquivo HTML chamado "dinossauros.html":**

**<input type="text"**

**name="nome"**

**size="100"**

**value="{{ dino.nome }}">**

**Agora observe o seguinte trecho de código no arquivo views.py:**

**def dino(request):**

**id\_dino = request.GET.get('id')**

**dados = {}**

**if id\_dino:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**return render(request, 'dinossauros.html', dados)**

**Com base nesse contexto, qual linha de código abaixo se encaixa adequadamente na lacuna?**

**R:** dados['dino'] = Dinossauros.objects.get(id=id\_dino)

**Durante a aula, no arquivo models.py, na classe Evento, foi escrito o seguinte trecho de código:**

**def get\_evento\_atrasado(self):**

**if self.data\_evento < datetime.now():**

**return True**

**else:**

**return False**

**Com base em seus conhecimentos e no código acima, qual expressão abaixo seria capaz de reduzir essas 4 últimas linhas em apenas 1 linha?**

**R:** return self.data\_evento < datetime.now()