SENAI - Cândido Athayde

Nome : ANTONIO PEREIRA CONCEIÇÃO JÚNIOR

Trabalho sobre models

O que são models:

O models é uma biblioteca com definições pré-definidas

Que serve para organizar e tratar e armazenar diferentes tipos de dados. junto com uma framework Django eles deixam os dados de uma formar que eles ficam mais compactar ele pode interagir com o usuário permitido que ele faça a criação , a leitura , atualização , e exclusão de registros .

Exemplo : quantos você vai organizar seu livros você pode organizar por ordem alfabética essa é a forma 1

Ou:

Organizar por classificação essa seria a forma 2

Sendo assim se os dados forem inteiros eles vão ser tratados de uma forma diferentes dos dados tipos texto

**Model CharField :**

**Em uma frameworks como o Django ele é um campo bastante usando para armazenar strings de textos com o comprimento limitado . você precisa especificar o parâmetro ‘max-length’ que determinar o comprimento máximo de strings que podem ser armazenadas neste campo. Ele oferece validação automática e é amplamente utilizado para representar dados de textos em aplicações web**

**Tipo :**

**O nome de uma pessoa .**

**Exemplo de como utilizar :**

**from django.db import models**

**class MinhaClasse(models,Model);**

**meu\_charfield =models.CharField(max\_length=100)**

**model BooleanField**

**É um tipo campo usado em um modelo de dados em uma frameworks de desenvolvimento web django para armazenar valores booleanos, ou seja valores que representam verdadeiro (true) ou falso (false).**

**Como funcionar**

**from django.db import models**

**class Exemplo(models.model):**

**meu\_booleano = models.booleanFiel(defult=False)**

**Model TextField :**

**é um tipo de campo utilizado para armazenar dados de textos de tamanhos variável o**

**TextField é usado para armazenar grandes blocos de textos.**

**Tipo:**

**Como artigos , descrições ou comentários .**

**Exemplo de como utilizar:**

**from django .db import models**

**class ExemploModelo(models,Model);**

**campo\_texto\_longo = models.TextField()**

**você não precisa especificar um tamanho máximo**

**Obs:**

**Funcionalidade Adicionais TextField também suporta funcionalidades como busca de texto completo (full-text search), se o banco de dados subjacente oferecer suporte a esse recurso.**

**Model IntegerField :**

**É um tipo de campo usado como modelos para armazenar números inteiros . utilizado quando você precisa armazenar dados numéricos que não possam ter casa decimais.**

**Tipo:**

**IDs , contagens , quantidades entre outros**

**Exemplo de como utilizar:**

**from django.db import models**

**class ExemploModelo(models,Model):**

**quantidade = models.IntegerField()**

**sem limite de tamanho .**

**validação automática O django realiza a validação automática dos dados inseridos no IntegerField . ele garante que apenas os dados inteiros sejam aceitos .**

**Model FloatField :**

**é um tipo de campo usado em modelos numéricos de pontos flutuantes . utilizado quando você armazenar valores numéricos que pode conter casas decimais .**

**Tipo:**

**Preços , Porcentagens , coordenadas geográficas , entre outros**

**Exemplo de como utilizar:**

**from django.db import models**

**class ExemploModelo(models,Model):**

**preco = models .FloatField()**

**validação automática O django realiza validação automática dos dados inseridos no FloatField . ele garante que apenas valores numéricos sejam aceitos, tanto como inteiros quando de pontos flutuantes .**

**Sem limite se tamanho**

**Model DateField :**

**Um tipo de campo usando para criação de datas**

**Exemplo:**

**data\_ nascimento = models.DateField()**

**Model ForeignKey :**

**Usado para criar relação de um para vários .**

**Parâmetro Obrigatório: “To”, Indica o modelo ao qual a chave estrangeira se refere.**

**Parâmetro Obrigatório: “on delete”**

**Model DateTimeField :**

**Usado para armazenar datas e horas em criação, atualização.**

**Exemplo:**

**from django.db import models**

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

**minha\_data\_hora = models.DataTimeField()**

**Model ForeignKey:**

**Usado para fazer uma conexão de um para todos**

**Exemplo:**

**from django.db import models**

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

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

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

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

**author = models.ForeignKey(Author, on\_delete=models.CASCADE)**

**Model OneToOneField:**

**Usado para fazer uma conexão de um para um.**

**Exemplo:**

**from django.db import models**

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

**user = models.OneToOneField(User, on\_delete=models.CASCADE)**

**bio = models.TextField()**

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

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

**email = models.EmailField()**

**Model ManyToManyField:**

**Usado para fazer conexão de vários para vários**

**Exemplo:**

**from django.db import models**

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

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

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

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

**authors = models.ManyToManyField(Author)**

**Model PositiveIntegerField:**

**Um tipo de campo que só permite números inteiros positivos.**

**Exemplo:**

**from django.db import models**

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

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

**quantity = models.PositiveIntegerField()**

**Model FileField:**

**Usado para representar e armazenar arquivos no servidor.**

**Exemplo:**

**from django.db import models**

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

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

**file = models.FileField(upload\_to='documents/')**

**Model EmailField:**

**Usado para armazenar endereço de email.**

**Exemplo:**

**from django.db import models**

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

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

**email = models.EmailField()**

**Model ImageField:**

**Usado para armazenar imagens .**

**Exemplo:**

**from django.db import models**

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

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

**image = models.ImageField(upload\_to='product\_images/')**