```
 import numpy as np
 from tensorflow.keras.models import Sequential
 from tensorflow.keras.layers import SimpleRNN, Dense
 from tensorflow.keras.utils import to_categorical
```

```
 text = "Hello World! Welcome to RNNs."
```

```
chars = sorted(list(set(text)))  # Unique characters
char_to_index = {c: i for i, c in enumerate(chars)}
index_to_char = {i: c for i, c in enumerate(chars)}
```

```
sequence_length = 10
sequences = []
next_chars = []
for i in range(len(text) - sequence_length):
 sequences.append(text[i:i + sequence_length])
 next_chars.append(text[i + sequence_length])
```

```
X = np.zeros((len(sequences), sequence_length, len(chars)), dtype=np.float32)
y = np.zeros((len(sequences), len(chars)), dtype=np.float32)
for i, seq in enumerate(sequences):
 for t, char in enumerate(seq):
  X[i, t, char_to_index[char]] = 1
  y[i, char_to_index[next_chars[i]]] = 1
```

```
 model = Sequential([
 SimpleRNN(128, input_shape=(sequence_length, len(chars))),
 Dense(len(chars), activation='softmax')
 ])
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204: UserWarning: Do not pass an `input_shape`/`input_dim` argument
  super().__init__(**kwargs)
```

```
 model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
 model.fit(X, y, epochs=5, batch_size=64)
```

```
Epoch 1/5
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - accuracy: 1.0000 - loss: 0.3048
Epoch 2/5
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 1.0000 - loss: 0.2648
Epoch 3/5
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 1.0000 - loss: 0.2306
Epoch 4/5
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - accuracy: 1.0000 - loss: 0.2014
Epoch 5/5
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 1.0000 - loss: 0.1764
<keras.src.callbacks.history.History at 0x79d3715af040>
```

```
 def generate_text(seed, length=50):
    generated_text = seed
    for _ in range(length):
        x_pred = np.zeros((1, sequence_length, len(chars)))
        for t, char in enumerate(seed):
            x_pred[0, t, char_to_index[char]] = 1

        predictions = model.predict(x_pred, verbose=0)[0]
        next_index = np.argmax(predictions)
        next_char = index_to_char[next_index]

        seed = seed[1:] + next_char  # Slide the window
        generated_text += next_char
    return generated_text
```

```
 seed_text = "Hello Worl"
 print("Generated Text:")
 print(generate_text(seed_text))
```

```
Generated Text:
Hello World! Welcome to RNNs.!lWelcome to RNNs.!lWelcome to
```

Start coding or generate with AI.