

```
"""Interface for embedding models."""
```

```
from abc import ABC, abstractmethod
```

```
from typing import List
```

```
class Embeddings(ABC):
```

```
    """Interface for embedding models."""
```

```
    @abstractmethod
```

```
    def embed_documents(self, texts: List[str]) -> List[List[float]]:
```

```
        """Embed search docs."""
```

```
    @abstractmethod
```

```
    def embed_query(self, text: str) -> List[float]:
```

```
        """Embed query text."""
```

```
    async def aembed_documents(
```

```
        self, texts: List[str]
```

```
    ) -> List[List[float]]:
```

```
        """Embed search docs."""
```

```
        raise NotImplementedError
```

```
    async def aembed_query(self, text: str) -> List[float]:
```

```
        """Embed query text."""
```

```
        raise NotImplementedError
```