Pydantic Model

Simple example of using PydanticAI to construct a Pydantic model from a text input.

Demonstrates:

- structured `result_type`

Running the Example

With dependencies installed and environment variables set, run:

pip

```
python -m pydantic_ai_examples.pydantic_model
```

uv

```
uv run -m pydantic_ai_examples.pydantic_model
```

This examples uses `openai:gpt-4o` by default, but it works well with other models, e.g. you can run it with Gemini using:

pip

```
PYDANTIC_AI_MODEL=gemini-1.5-pro python -m pydantic_ai_examples.pydantic_model
```

uv

```
PYDANTIC_AI_MODEL=gemini-1.5-pro uv run -m pydantic_ai_examples.pydantic_model
```

(or `PYDANTIC_AI_MODEL=gemini-1.5-flash ...`)

Example Code

pydantic_model.py

```python
import os
from typing import cast

import logfire
from pydantic import BaseModel

from pydantic_ai import Agent
from pydantic_ai.models import KnownModelName

# 'if-token-present' means nothing will be sent (and the example will work) if you don't have logfire configured
logfire.configure(send_to_logfire='if-token-present')


class MyModel(BaseModel):
    city: str
    country: str


model = cast(KnownModelName, os.getenv('PYDANTIC_AI_MODEL', 'openai:gpt-4o'))
print(f'Using model: {model}')
agent = Agent(model, result_type=MyModel)

if __name__ == '__main__':
    result = agent.run_sync('The windy city in the US of A.')
    print(result.data)
    print(result.cost())
```