

# Pydantic

Favorite Tool for Data Validation in python

---

Lewi

31.08.2023



-

# Why Pydantic

- Encourages Usage of Type Hints
- Fast
- Autogenerates JSON Schema
- Complete custom validation
- Used in many other useful Frameworks
- Awesome Documentation

---

```
from pydantic import BaseModel

class StudentRoom(BaseModel):
    room_number: str
    rent: int

r = StudentRoom(room_number='a111', rent=250)
print(r.room_number)
# > a111
r2 = StudentRoom(room_number='a111', rent=[250])
# > Input should be a valid integer
# > [type=int_type, input_value=[250], input_type=list]
```

---

## What to use it for?

- Configuration Validation
- Data Modeling
- REST Validation

---

```
StudentRoom.model_json_schema()  
#> { 'properties': {  
#>   'room_number': { 'title': 'Room Number', 'type': 'string' },  
#>   'rent': { 'title': 'Rent', 'type': 'integer' }  
#> },  
#> 'required': ['room_number', 'rent'],  
#> 'title': 'StudentRoom',  
#> 'type': 'object'  
#> }
```

---

```
from typing import Union
```

```
from fastapi import FastAPI
```

```
app = FastAPI()
```

```
@app.get("/")  
def read_root():  
    return {"Hello": "World"}
```

```
@app.get("/items/{item_id}")  
def read_item(item_id: int, q: Union[str, None] = None):  
    return {"item_id": item_id, "q": q}
```