

On Declarations

- ◆ Note that a Python function definition can be spare

No declaration of parameter types or return type

- ◆ How Can this work?

- ◆ Each **object** has all the type information bound into it

- ◆ Our **odd()** function will work with any object that implements the % operator

- ◆ Which means **any** object offering **__mod__()** or **__rmod__()** methods

- ◆ Which means most subclass of **numbers.Number**

Python 3 Type Hints

- ◆ You can include domain and range information

```
def odd(n : int) -> bool:  
    return n%2 != 0
```

- ◆ Consider the improvement in clarity
- ◆ Python 3.5+ includes sophisticated type definitions
- ◆ The **mypy** package can validate that the code is consistent with the type definitions