

# OOP

קוד לדוגמא-

<https://github.com/pythonai250824/18.05.2025>

בנה את המחלקה שלפניך

## OOP Exercise: Encapsulation with Customer Class

### Goal:

Create a Python class that demonstrates **encapsulation** using:

- `__init__`
- `__str__` [pythonai250824+HWOOP1@gmail.com](mailto:pythonai250824+HWOOP1@gmail.com)
- `__repr__`
- `@property` getter
- `@property.setter`

בהצלחה!

### Instructions:

1. Create a class called `Customer`.
2. The class should have the following attributes:
  - `name` (public)
  - `email` (public)
  - `__id` (private using `__`)
3. Implement the following:
  - `__init__`: initialize `name`, `email`, and `__id`
  - `@property` for reading the value of `__id`
  - `@id.setter` that allows changing the ID **only if it's a positive integer**
  - `__str__`: return a readable string like:  
`"Customer <name> (ID: <id>) - Email: <email>"`
  - `__repr__`: return a string like:  
`"Customer(name='<name>', email='<email>', id=<id>)"`

4. At the bottom of your script:

- Create two instances of the class:

```
c1 = Customer("Alice", "alice@example.com", 101)
c2 = Customer("Bob", "bob@example.com", 202)
```

- Print both using `print()` and `repr()`
- Try setting an invalid ID like `c1.id = -5` and confirm that it is rejected

#### ✓ Example Output:

```
print(c1)           # Customer Alice (ID: 101) - Email: alice@example.com
print(repr(c2))     # Customer(name='Bob', email='bob@example.com', id=202)
```

[pythonai250824+HWOOP2@gmail.com](mailto:pythonai250824+HWOOP2@gmail.com)

בהצלחה!