

# Chapter 9

Fields Onchange 

# onchange

CLICK HERE

- Field **recomputed instantly** if another field is modified **in the view**
- Defined using **api decorator "onchange"** on method

```
from odoo import api, fields, models

class TestModel(models.Model):
    _name = "test.model"
    _description = "Test Model"

    name = fields.Float()
    partner_id = fields.Many2one("res.partner")

    @api.onchange("partner_id")
    def _onchange_partner_id(self):
        self.name = "House - %s" % (self.partner_id.name)
```



**Never use onchange for business logic !**

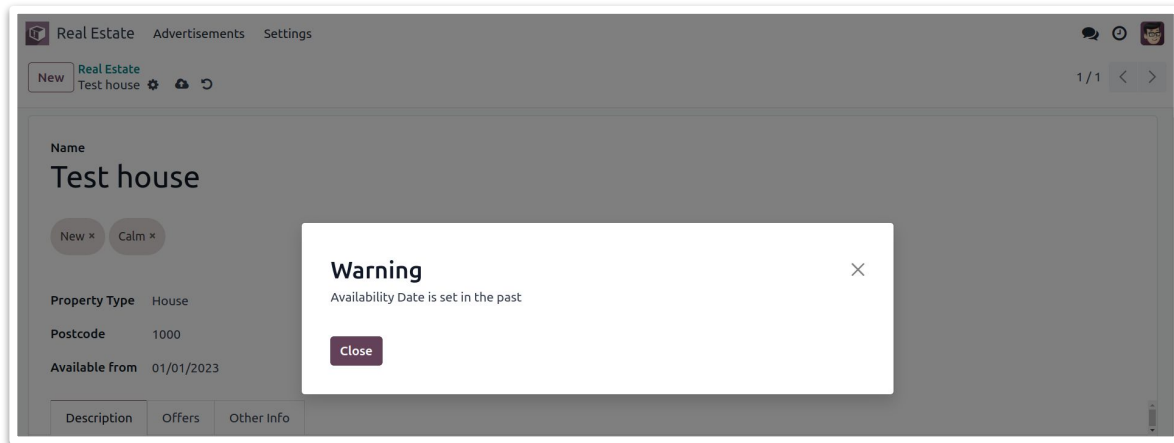


onchanges are only triggered updating fields in the **form view**



# Non blocking warning

Return non-blocking  
pop-up warning in  
onchange on condition



```
amount = fields.Float()

@api.onchange("amount")
def _onchange_amount(self):
    if self.amount < 0:
        return {"warning": {"title": _("Warning"), "message": _("Amount is negative")}}
```



# Exercise

## Chapter 9

On change

Set values for **garden area** and **orientation**.

Create an **onchange** in the **estate.property** model in order to set values for the garden area and orientation when **garden** is set to **True**.

When unset, clear the fields.



# Exercise

## Chapter 9

On change

Raise a **soft warning**.

Create a **soft warning** in the `estate.property` model if the field `date_availability` is set to a date **prior than today's**.

