在 Python 正则表达式中：

**✅ (?:...) 表示 非捕获组（non-capturing group）**

**🔹 它的作用是：**

像普通的括号 (...) 一样 **对内容进行分组**，但 **不会捕获这个分组内容供后续引用或返回**。

**📌 为什么需要非捕获组？**

有时你只是想用括号来：

* 控制正则运算优先级；
* 使用 | 或重复次数（比如 {n}）；
* 但你**不需要拿到这个括号里的内容**。

这时就用 (?:...)，可以避免多余的 group() 结果。

**✅ 举例说明：**

import re

text = "abc123"

res = re.match(r"(abc)(123)", text)

print(res.groups()) # 输出：('abc', '123')

这个例子中有两个捕获组。

如果你只想捕获 "123"，"abc"只是为了结构用途，可以写成：

res = re.match(r"(?:abc)(123)", text)

print(res.groups()) # 输出：('123',)

**✅ 常见用途示例**

**1. 多选项分组但不捕获：**

pattern = r"cat(?:fish|woman|nap)"

匹配 catfish、catwoman、catnap，但不会单独捕获 fish、woman 等内容。

**2. 重复结构但只捕获外层：**

pattern = r"(?:\d{3}-){2}\d{4}" # 匹配 123-456-7890

你只是想重复 \d{3}- 两次，但不想捕获中间结果。

**✅ 总结：**

| **表达式** | **作用** |
| --- | --- |
| (abc) | 捕获组，group(1) 会返回 'abc' |
| (?:abc) | 非捕获组，不会出现在 groups() 结果中 |
| (?P<name>abc) | 命名捕获组，可用 group("name") 访问 |

如果你想知道怎么配合使用非捕获组和回溯、命名组、替换等，也可以告诉我，我可以详细举例。