

GRUPOS PASIVOS

En las **expresiones regulares**, los **grupos pasivos** (o **grupos no capturantes**) son aquellos grupos que no capturan el texto coincidente para su posterior uso o referencia, a diferencia de los grupos capturantes estándar. Esto significa que el texto que coincide con un grupo pasivo no se almacena para acceder a él después, como lo haría un grupo capturante.

Sintaxis de los grupos pasivos:

regex

(?:expresión)

Donde (?:...) es la estructura de un grupo pasivo, y dentro de los paréntesis se coloca la **expresión** regular que se desea agrupar.

Diferencias entre grupos capturantes y pasivos:

Grupos capturantes: Se definen entre paréntesis normales () y **capturan** el texto coincidente. Cada coincidencia puede referenciarse posteriormente, tanto en el código como en el reemplazo.

Ejemplo:

regex

(abc|def)

1. Si la cadena es "abc", capturará "abc" como el grupo 1.

Grupos pasivos: Se definen con (?:...) y **no capturan** el texto coincidente, aunque siguen agrupando partes de la expresión para aplicar cuantificadores, alternancias, etc.

Ejemplo:

regex

(?:abc|def)

2. Si la cadena es "abc", no se capturará el texto, pero la expresión puede coincidir con "abc" o "def".

¿Cuándo usar un grupo pasivo?

Los grupos pasivos son útiles cuando deseas agrupar una parte de la expresión para aplicar operadores como *****, **+**, **?**, o **|**, pero no necesitas almacenar esa parte de la coincidencia para usarla después. Esto optimiza el uso de recursos, ya que no se guarda información innecesaria.

Ejemplo:

regex

```
(?:https?|ftp)://(www\.)?(\w+\.\w+)
```

En este ejemplo:

- `(?:https?|ftp)` es un **grupo pasivo** que no captura "http", "https", ni "ftp", pero agrupa estas alternativas para que la expresión coincida correctamente.
- `(www\.)?` y `(\w+\.\w+)` son grupos capturantes porque quizás quieras almacenar el dominio "www." y la dirección principal (por ejemplo, "example.com").

En resumen, los **grupos pasivos** son útiles cuando necesitas agrupar sin capturar, ayudando a simplificar y optimizar las expresiones regulares.