**Expresiones regulares**

"[xyz]" Corchetes = opcionalidad (**uno** de los símbolos)

"[a-z]" "[A-Z]" "[a-zA-Z]" cualquier símbolo entre los dados (diferencia mayús. / minús.)

"a?" > Una o ninguna vez

"a\*" > 0 o más veces

"a+" > al menos una vez

"a{1,4}" > de 1 a 4 veces

"a{2,}" > mínimo 2, posible más

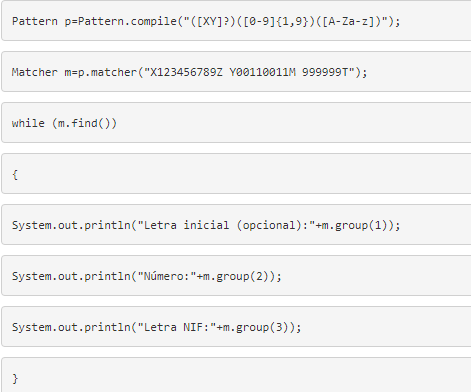
"a{5}" > **exactamente** 5 veces

"[^abc]" > ^ = negación (admite símbolos que no sean los dados)

"." > cualquier carácter

"(#[01]){2,3}" > Paréntesis = englobar símbolos para aplicar repeticiones y definir grupos

**Grupos:**



"\\" > Secuencia de escape (para símbolos especiales como paréntesis)

**Excepción: Las comillas, que se pondrían con una sola barra: "** \" **"**

"\\d". Un dígito numérico. Equivale a "[0-9]".

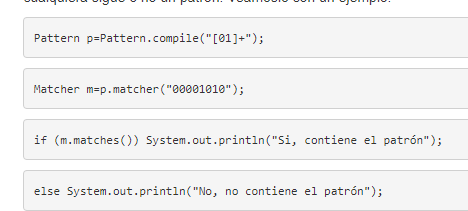
"\\D". Cualquier cosa excepto un dígito numérico. Equivale a "[^0-9]"

"\\s". Un espacio en blanco (incluye tabulaciones, saltos de línea y otras formas de espacio).

"\\S". Cualquier cosa excepto un espacio en blanco.

"\\w". Cualquier carácter que podrías encontrar en una palabra. Equivale a "[a-zA-Z\_0-9]".

**Uso:**



(crear objetos Pattern (con la expresión) y Matcher (con la cadena a comprobar) y usar métodos de Matcher)

m.matches() > encaja (exacto)

m.lookingAt() > patrón al principio de la cadena

m.find() > en cualquier posición:

* Obtener pos: m.start() y m.end()
* siguiente coincidencia: volver a usar el método
* reiniciar método: m.reset()