Unidad 2, Anexo I: Expresiones regulares

- PHP permite utilizar funciones para expresiones regulares como **preg_grep**.
- Una expresión regular permite comparar un patrón frente a un texto, para comprobar si el texto contiene lo especificado en el patrón.
- Las expresiones regulares se deben encerrar entre delimitadores. Un delimitador puede ser cualquier carácter no alfanumérico que no sea una barra invertida (\) ni un espacio en blanco, se suelen usar los caracteres /, #, +, %, (), {}, [] y <>.
- Si utilizamos un delimitador con apertura y cierre como los paréntesis, llaves, corchetes, etc debemos usar el de apertura al principio y el de cierre al final. Si es otro delimitador debemos usar el mismo al principio y al final de la expresión regular.
- Los delimitadores no forman parte de la expresión, por lo que en los siguientes ejemplos no se incluyen aunque haya que ponerlos en la llamada a **preg_grep**.
- Ejemplos de patrones de búsqueda:
  - Patrón: in

    Coinciden:

    intensidad

    c**in**ta

    interior
  - Patrón: [mp]adre

    Coindicen:

    Mi **madre** se llama Luisa

    Tu **padre** es jardinero

Expresiones regulares: Sintaxis básica

- El punto
  - El punto representa cualquier carácter. Desde la A a la Z (en minúscula y mayúscula), del 0 al 9, o algún otro símbolo.

    ca.a coincide con ca**n**a, ca**m**a, ca**s**a, ca**j**a, etc...

    No coincide con ca**st**a ni *caa*
- Principio y fin de cadena
  - Si queremos indicar al patrón qué es el principio de la cadena o qué es el final, debemos hacerlo con **^ para inicio y $ para final**.

    “^olivas” coincide con “**olivas** verdes“,

    pero no con “quiero olivas”

Expresiones regulares: Sintaxis básica

- Cuantificadores
  - Para indicar que cierto elemento del patrón va a repetirse un **número indeterminado de veces**, usaremos + (una o más veces) o * (cero o más veces) .

"gafas+" coincide con "gafa**ssss**"

pero no con "gafa"

"clo*aca" coincide con "claca", "cloaca",

"clooooooooaca", etc..

Expresiones regulares: Sintaxis básica

- El interrogante indica que un elemento **puede que esté (una vez) o puede que no**:

  “coches?” coincide con “coche” y con “coches”

- Las llaves { } definen la **cantidad de veces que va a repetirse el elemento**:

  “abc{4}” coincide con “abcccc“

  pero no con “abc” ni “abcc“, etc...

  “abc{1,3}” coincide con “abc“, “abcc“, “abccc“,

  pero no con “abcccc”

Expresiones regulares: Sintaxis básica

- Si un parámetro queda vacío, significa “un **número indeterminado**“. Por ejemplo:

  “x{5,}” la x ha de repetirse 5 veces, o más.

- Rangos
  - Los corchetes [] incluidos en un patrón permiten especificar el **rango de caracteres** válidos a comparar.

    “c[ao]sa” coincide con “casa” y con “cosa”

    “[a-f]” coincide con todos los caracteres alfabéticos de la “a” a la “f”

    “[0-9][2-6][ANR]” coincide con “12A“, “35N“, “84R“, etc..

    pero no con “2**1**A“, ni “33**L**“, ni “3A“, etc...

Expresiones regulares: Sintaxis básica

- Dentro de los corchetes el símbolo ^ es un negador, es decir:

  "[^a-Z]" coincidirá con cualquier texto que NO tenga ningún carácter alfabético (ni minúsculas ni mayúsculas)

  "[^@]" coincide con cualquier carácter excepto "@" y "espacio"

- Alternancia
  - Para alternar entre varias opciones usaremos el símbolo |. Si una de las opciones coincide el patrón será cierto.

    "aleman(ia|es)" coincide con "aleman**ia**" y con "aleman**es**"

    "(norte|sur|este|oeste)" coincide con cualquiera de los puntos cardinales.

Expresiones regulares: Sintaxis básica

- Agrupadores
  - Los paréntesis nos sirven para agrupar un subconjunto.

    “(abc)+” coincide con “abc“, “abcabc“, “abcabcabc“, etc

    “ca(sca)?da” coincide con “cascada” y con “cada”
- Escapar caracteres
  - Si queremos utilizar caracteres especiales en el patrón hubiese sin que se interprete como metacáracter, tendremos que “escaparlo”. Esto se hace poniendo una barra invertida justo antes:

    “\.” o “*”