

1. `/^\d{3}-\d{2}-\d{4}$/` --> que empiece por 3 dígitos seguidos de un "-" luego 2 dígitos seguidos de un "-" y que finalice con 4 dígitos.
(123-45-678)
2. `/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/` --> que empiece por uno o mas caracteres de los cuales son mayúsculas, minúsculas, números y los caracteres permitidos, luego seguido de una "@", seguido uno o mas caracteres de los cuales son mayúsculas, minúsculas, números y los caracteres permitidos y al final debe tener un "." seguido de mínimo dos caracteres mayúsculas o minúsculas.
(abCD12._%+-@efGH34.-.ij)
3. `/^([01]?[0-9]|2[0-3]):[0-5][0-9]$/` --> que empiece opcionalmente 01 seguido de un número del 0 al 9 o dos números del 0 al 3, que luego contenga ":", seguido de un número del 0 al 5 y que termine con un número del 0 al 9
(14:30)
4. `/^(http|https):\/\/[a-zA-Z0-9]+\.[a-zA-Z]{2,}/` --> que empiece por "http" o "https" seguido de "://", seguidamente un o mas caracteres de minúsculas, mayúsculas y números, seguido de un "." y esto seguido de mínimo 2 caracteres de minúsculas y mayúsculas
(http://localhost.com)
5. `/^[^0-9]+$/` --> que no empiece por un número y que termine por uno o mas caracteres que no sea un número
(ruben es muy listo)
6. `/(\b\w+\b).*/` --> que encuentre una palabra completa que contenga mayúsculas, minúsculas, números y _, seguido de cualquier carácter desde 0 hasta el que sea +
(Hola mundo, esto es una prueba)
7. `/^(\+|-)?\d+$/` --> que opcionalmente empiece por "+" o "-" y que finalice por uno o mas dígitos
(+123)
8. `/^[aeiou]([a-z]|[0-9])+$/i` --> que empiece por una vocal y que finalice de uno o mas minúsculas o números del 0 al 9 y no distingue entre mayúsculas y minúsculas
(uhello123)
9. `/^\d{1,3}(\,\d{3})*(\.\d+)?$/` --> que empiece con uno a tres dígitos, seguido opcionalmente de grupos de tres dígitos separados por comas, y opcionalmente termine con un punto seguido de uno o más dígitos. Esto se usa para números con formato de miles y decimales.
(1,234)
10. `/^\s*(.*?)\s*$/` --> que permita cualquier cantidad de espacios al inicio y al final de una cadena, capturando todo lo que esté entre ellos. El `.*?` asegura que capture todo el contenido sin incluir espacios adicionales.
(texto con espacios).

11. `/^([01]?[0-9]|2[0-3]):[0-5][0-9]\s(am|pm)$/i` --> que empiece con un número de hora válido en formato de 12 o 24 horas (de 01 a 23), seguido de ":", minutos (de 00 a 59), un espacio, y termine con "am" o "pm" (sin distinguir entre mayúsculas y minúsculas por la i).

(11:30 am)

12. `/^\d{1,2}\d{1,2}\d{4}$/` --> que siga el formato de una fecha con día y mes de uno o dos dígitos, seguido de "/", y un año de cuatro dígitos.

(12/05/2023)