

1. **str.capitalize()** Retorna una copia de la cadena con el primer carácter en mayúsculas y el resto en minúsculas.
2. **str.casefold()** Retorna el texto de la cadena, normalizado a minúsculas. Los textos así normalizados pueden usarse para realizar búsquedas textuales independientes de mayúsculas y minúsculas.
3. **str.count(sub[, start[, end]])** Retorna el número de ocurrencias no solapadas de la cadena sub en el rango [start, end]. Los parámetros opcionales start y end se interpretan como en una expresión de slice.
4. **str.find(sub[, start[, end]])** Retorna el menor índice de la cadena s donde se puede encontrar la cadena sub, considerando solo el intervalo s[start:end]. Los parámetros opcionales start y end se interpretan como si fueran índices de un slice. Retorna -1 si no se encuentra la cadena. Nota El método find() se debe usar solo si se necesita saber la posición de la cadena sub. Si solo se necesita comprobar si sub es una parte de s, es mejor usar el operador in.
5. **str.isalpha()** Retorna True si todos los caracteres de la cadena son alfabéticos y hay, al menos, un carácter. En caso contrario, retorna False.
6. **str.isdigit()** Retorna True si todos los caracteres de la cadena son dígitos y hay, al menos, un carácter. En caso contrario, retorna False.
7. **str.isnumeric()** Retorna True si todos los caracteres de la cadena son caracteres numéricos y hay, al menos, un carácter. En caso contrario, retorna False.
8. **str.islower()** Retorna True si todos los caracteres de la cadena que tengan formas en mayúsculas y minúsculas están en minúsculas y hay, al menos, un carácter de ese tipo. En caso contrario, retorna False.
9. **str.isupper()** Retorna True si todos los caracteres de la cadena que tengan formas en mayúsculas y minúsculas están en mayúsculas y hay, al menos, un carácter de ese tipo. En caso contrario, retorna False.
10. **str.upper()** Retorna una copia de la cadena, con todos los caracteres con formas mayúsculas/minúsculas pasados a mayúsculas.
11. **str.lower()** Retorna una copia de la cadena, con todos los caracteres con formas mayúsculas/minúsculas pasados a minúsculas.
12. **str.replace(old, new[, count])** Retorna una copia de la cadena con todas las ocurrencias de la cadena old sustituidas por new. Si se utiliza el parámetro count, solo se cambian las primeras count ocurrencias.
13. **str.split(sep=None, maxsplit=- 1)** Retorna una lista con las palabras que componen la cadena de caracteres original, usando como separador el valor de sep.
14. **str.strip([chars])** Retorna una copia de la cadena con los caracteres indicados eliminados, tanto si están al principio como al final de la cadena. El parámetro opcional chars es una cadena que especifica el conjunto de caracteres a eliminar. Si se omite o se usa None, se eliminan los caracteres de espacio en blanco.