`strtof` is identical to `strtod`, except that it converts a string to a `float` value.

Returns The converted number. Returns zero if no conversion could be performed. If the number is too large to represent, returns plus or minus `HUGE_VALF`, depending on the number's sign. If the number is too small to represent, returns a value whose magnitude is no greater than the smallest normalized positive `float`. *26.2*

`strtoimax` *Convert String to Greatest-Width Integer (C99)* `<inttypes.h>`

```
intmax_t strtoimax(const char * restrict nptr,
                   char ** restrict endptr, int base);
```

`strtoimax` is identical to `strtol`, except that it converts a string to a value of type `intmax_t` (the widest signed integer type).

Returns The converted number. Returns zero if no conversion could be performed. If the number can't be represented, returns `INTMAX_MAX` or `INTMAX_MIN`, depending on the number's sign. *27.2*

`strtok` *Search String for Token* `<string.h>`

```
char *strtok(char * restrict s1,
             const char * restrict s2);
```

Searches the string pointed to by `s1` for a "token" consisting of characters not in the string pointed to by `s2`. If a token exists, the character following it is changed to a null character. If `s1` is a null pointer, a search begun by the most recent call of `strtok` is continued; the search begins immediately after the null character at the end of the previous token.

Returns A pointer to the first character of the token. Returns a null pointer if no token could be found. *23.6*

`strtol` *Convert String to Long Integer* `<stdlib.h>`

```
long int strtol(const char * restrict nptr,
                char ** restrict endptr, int base);
```

Skips white-space characters in the string pointed to by `nptr`, then converts subsequent characters into a `long int` value. If `base` is between 2 and 36, it is used as the radix of the number. If `base` is zero, the number is assumed to be decimal unless it begins with `0` (octal) or with `0x` or `0X` (hexadecimal). If `endptr` isn't a null pointer, `strtol` modifies the object pointed to by `endptr` so that it points to the first leftover character. If no `long int` value is found, or if it has the wrong form, `strtol` stores `nptr` in the object pointed to by `endptr`. If the number can't be represented, it stores `ERANGE` in `errno`.

Returns The converted number. Returns zero if no conversion could be performed. If the number can't be represented, returns `LONG_MAX` or `LONG_MIN`, depending on the number's sign. *26.2*

`strtold` *Convert String to Long Double (C99)* `<stdlib.h>`

```
long double strtold(const char * restrict nptr,
                    char ** restrict endptr);
```