

NAME

_Generic – type-generic selection

SYNOPSIS

_Generic(*expression*, **type1**: *e1*, ... /*, **default**: *e* */);

DESCRIPTION

_Generic() evaluates the path of code under the type selector that is compatible with the type of the controlling *expression*, or **default**: if no type is compatible.

expression is not evaluated.

This is especially useful for writing type-generic macros, that will behave differently depending on the type of the argument.

STANDARDS

C11.

HISTORY

C11.

EXAMPLES

The following program demonstrates how to write a replacement for the standard **imaxabs**(3) function, which being a function can't really provide what it promises: seamlessly upgrading to the widest available type.

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>

#define my_imaxabs _Generic(INTMAX_C(0), \
    long:      labs, \
    long long: llabs \
    /* long long long: lllabs */ \
    )

int
main(void)
{
    off_t  a;

    a = -42;
    printf("imaxabs(%jd) == %jd\n", (intmax_t) a, my_imaxabs(a));
    printf("&imaxabs == %p\n", &my_imaxabs);
    printf("&labs == %p\n", &labs);
    printf("&llabs == %p\n", &llabs);

    exit(EXIT_SUCCESS);
}
```