



FROM DOCUMENTATION TO PROGRAMS

Using technical documentation to write programs

COMMON DOCUMENT SECTIONS

Info ▼	Src ▶	cplusplus.com	Microsoft	Unix / Linux
Function Name	Function	Function	Unlabeled	Name
Prototype	Follows function	Follows function	Syntax	Synopsis
Header File	Follows function	Follows function	Requirements	Synopsis
Brief Description	Derived - function	Derived - function	Unlabeled	Description
Input	Parameters	Parameters	Parameters	SPECIAL VALUES
Return Value	Return Value	Return Value	Output	N/A
Example	Example	Example	Example	N/A
Related Docs	See also	See also	See Also	See Also



TYPE ALIASES: PORTABLE DATA TYPES

- C++ is deliberately vague about the size and sign of some data types
 - C was designed as a partial replacement for assembly, so it is tied to hardware more than other languages – the ANSI standard calls this “implementation dependent”
- Type aliases
 - usually end with `_t`
 - are replaced with “real” types by the compiler
 - `typedef size_t unsigned int`
 - `typedef size_t unsigned long`



TYPE ALIAS EXAMPLE

- `size_t strlen(const char* str);`

```
int main()
{
    char* s = "Hello world");
    for (size_t i = 0; i < strlen(s); i++)
        cout << s[i] << endl;
    return 0;
}
```



ERROR REPORTING

```
int main()
{
    double x = sqrt(-2);

    if (errno != 0)
        cout << errno << endl;
    else
        cout << x << endl;

    return 0;
}
```

- Four bit patterns are not numbers
 - \pm NaN – Not a Number
 - \pm INF – infinity
- `errno`
 - `int`
 - `errno_t`
 - EDOM – domain error
- `perror("sqrt error");`
 - `sqrt error: Domain error`



SIMPLE FUNCTIONS

- `double pow(double base, double exponent);` $\text{base}^{\text{exponent}}$
- `double pow(double x, double y);` X^y
- The result of raising *base* to the power *exponent*.

- `cout << pow(3.14159, 2.0) << endl;`
- `double result = pow(h, 2);`
- `double payment = p * r / (1 - pow(1 + r, -n));`



STRUCTURE ARGUMENTS (WINDOWS VERSION)

- `#include <sys/timeb.h>`
`errno_t _ftime_s(struct _timeb* timeptr);`
- `struct _timeb* start; // error!!`
`_ftime_s(start);`
- `struct _timeb start; // correct, windows`
`_ftime_s(&start);`