Primitive Data Types are Always Available

Primitive data types
- part of the **C** language
- include **unsigned**, **2's complement**, and **IEEE floating-point**
- 8-bit primitive data types can also be used to store **ASCII** characters

Pitfall #3: Primitive Data Types Depend on the System

Since the **C** language was designed to be efficient, **primitive data types are tuned to the system**.

Unfortunately, that means the actual data type can vary from one compiler to another.

For example, `long int` may be a **32-bit 2's complement** value, or it may be a **64-bit 2's complement** value.

Primitive Integer and Floating-Point Types in C

	2's complement	unsigned
8 bits	`char`	`unsigned char`
16 bits	`short` `short int`	`unsigned short` `unsigned short int`
16 or 32 bits	`int`	`unsigned` `unsigned int`
32 or 64 bits	`long` `long int`	`unsigned long` `unsigned long int`
64 bits	`long long` `long long int`	`unsigned long long` `unsigned long long int`

IEEE 754 single-precision floating-point (32 bits) `float`
IEEE 754 double-precision floating-point (64 bits) `double`

Standard Integer Types in C

ISA-independent integer types
- available in `<stdint.h>`.
- We will use them except for `main` and some library calls.

	2's complement	unsigned
8 bits	`int8_t`	`uint8_t`
16 bits	`int16_t`	`uint16_t`
32 bits	`int32_t`	`uint32_t`
64 bits	`int64_t`	`uint64_t`

