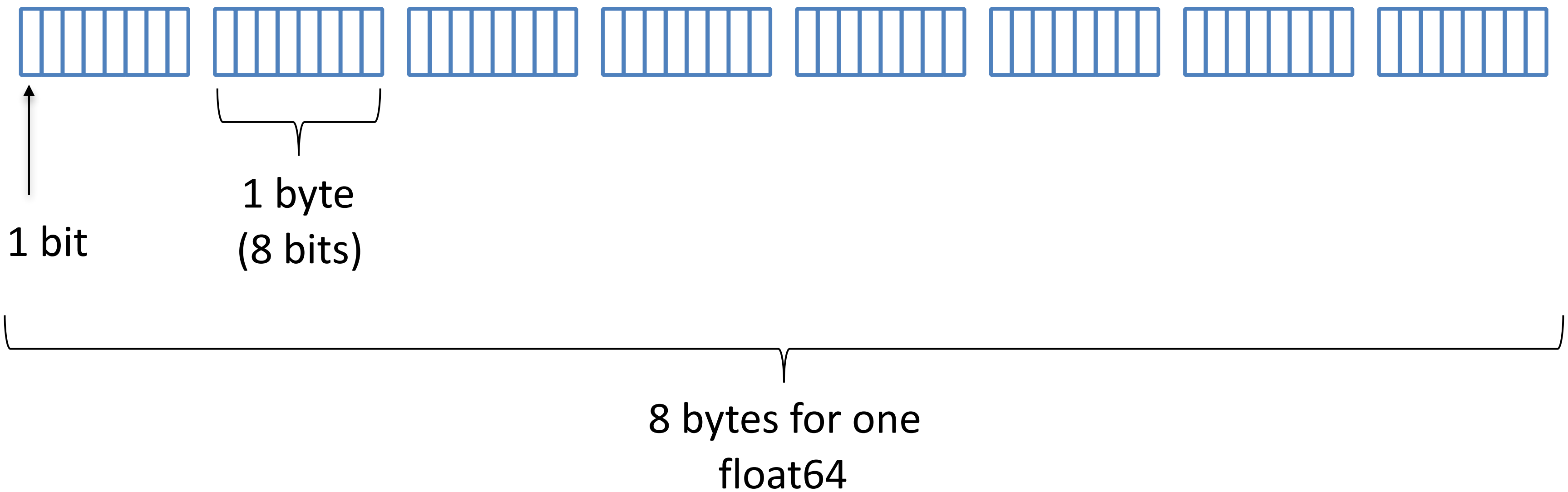
Numeric Types
in numpy

double-precision floating point (float64)



https://docs.scipy.org/doc/numpy-1.12.0/user/basics.types.html

Note that numpy supports a much greater variety of numeric types than base Python does.

Data type	Description
`bool_`	Boolean (True or False) stored as a byte
`int_`	Default integer type (same as C `long`; normally either `int64` or `int32`)
`intc`	Identical to C `int` (normally `int32` or `int64`)
`intp`	Integer used for indexing (same as C `ssize_t`; normally either `int32` or `int64`)
`int8`	Byte (-128 to 127)
`int16`	Integer (-32768 to 32767)
`int32`	Integer (-2147483648 to 2147483647)
`int64`	Integer (-9223372036854775808 to 9223372036854775807)
`uint8`	Unsigned integer (0 to 255)
`uint16`	Unsigned integer (0 to 65535)
`uint32`	Unsigned integer (0 to 4294967295)
`uint64`	Unsigned integer (0 to 18446744073709551615)
`float_`	Shorthand for `float64`.
`float16`	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
`float32`	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
`float64`	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
`complex_`	Shorthand for `complex128`.
`complex64`	Complex number, represented by two 32-bit floats
`complex128`	Complex number, represented by two 64-bit floats

binary number (base 2)

1 1 0 1 0 0 1 1

decimal number (base 10)

211

hexadecimal number (base 16)

D3

binary number (base 2)

1 1 0 1 0 0 1 1

decimal number (base 10)

211

$$= 2*10^2+1*10^1+1*2^0$$

$$= 2*100+1*10+1*1$$

$$= 211$$

hexadecimal number (base 16)

D3

binary number (base 2)

1 1 0 1 0 0 1 1

$$= 1*2^7+1*2^6+0*2^5+1*2^4+0*2^3+0*2^2+1*2^1+1*2^0$$
$$= 1*128+1*64+0*32+1*16+0*8+0*4+1*2+1*1$$
$$= 211$$

decimal number (base 10)

211

$$= 2*10^2+1*10^1+1*2^0$$
$$= 2*100+1*10+1*1$$
$$= 211$$

hexadecimal number (base 16)

D3

binary number (base 2)

1 1 0 1 0 0 1 1

$$= 1*2^7+1*2^6+0*2^5+1*2^4+0*2^3+0*2^2+1*2^1+1*2^0$$
$$= 1*128+1*64+0*32+1*16+0*8+0*4+1*2+1*1$$
$$= 211$$

decimal number (base 10)

211

$$= 2*10^2+1*10^1+1*2^0$$
$$= 2*100+1*10+1*1$$
$$= 211$$

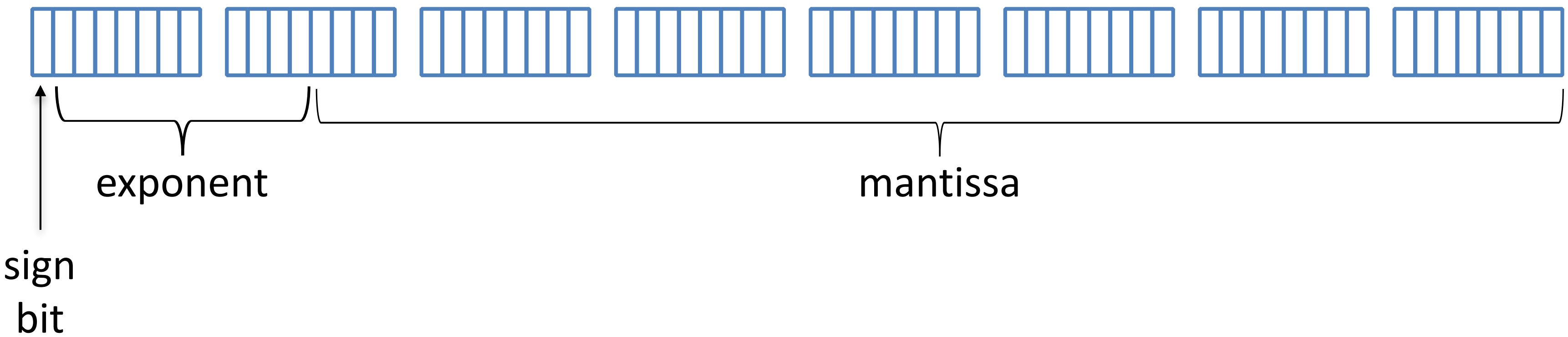
hexadecimal number (base 16)

D3

$$= D*16^1+3*16^0$$
$$= 13*16+3*1$$
$$= 211$$

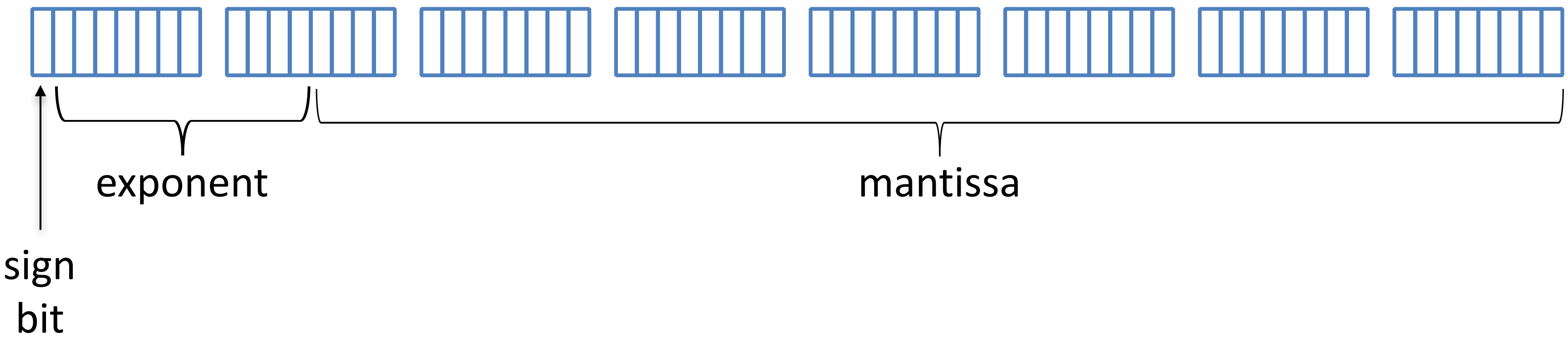
A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

double-precision floating point (float64)

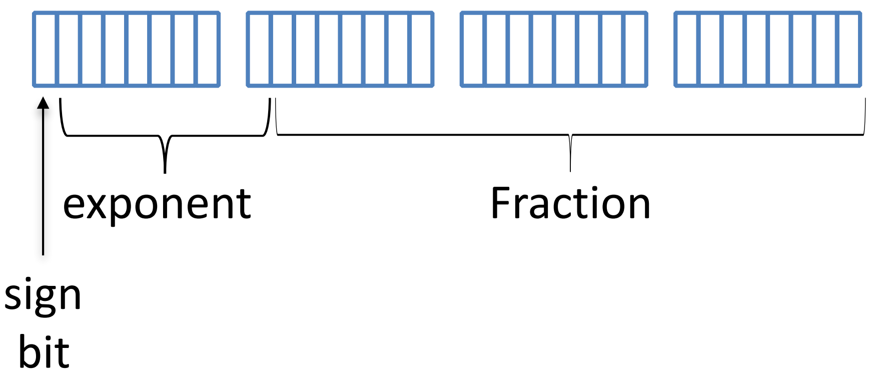


`float64` Double precision float: sign bit, 11 bits exponent, 52 bits mantissa

double-precision floating point (float64)

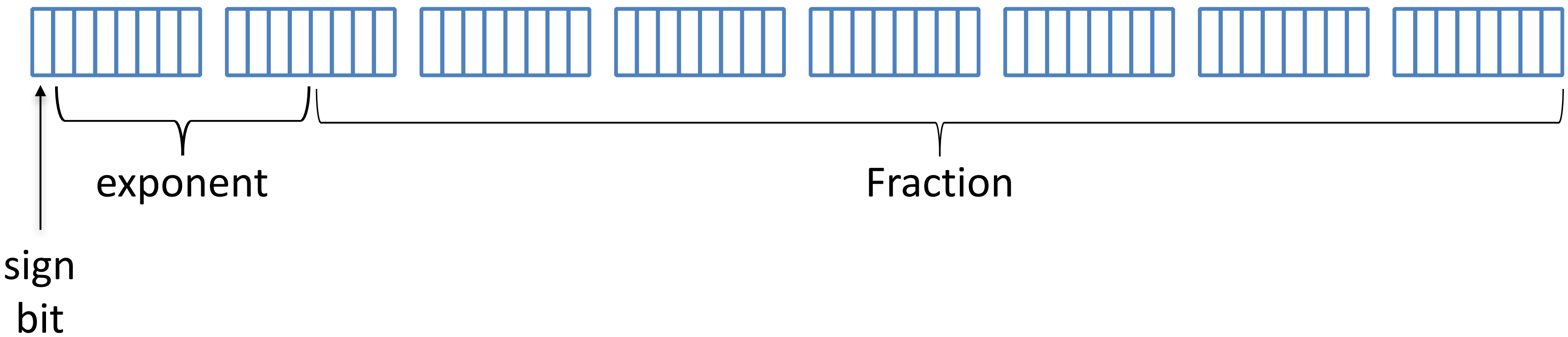


single-precision floating point (float32)

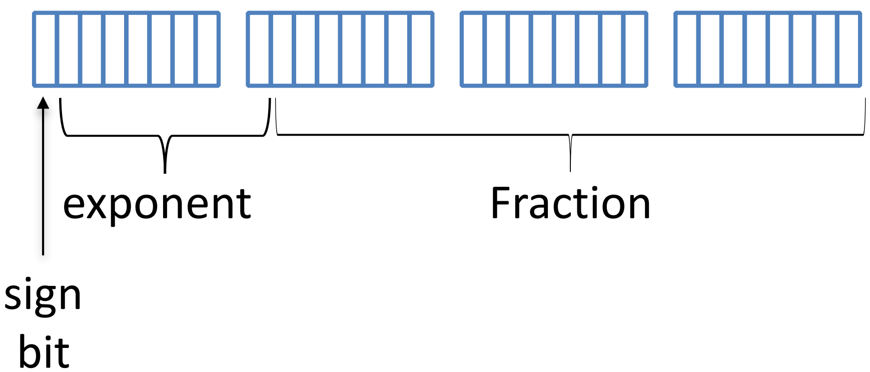


`float32` Single precision float: sign bit, 8 bits exponent, 23 bits mantissa

double-precision floating point (float64)

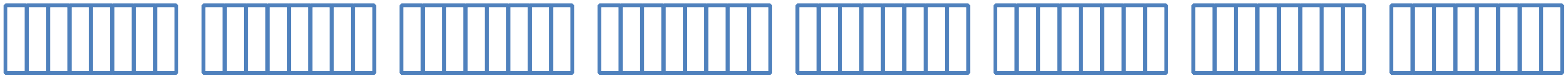


single-precision floating point (float32)

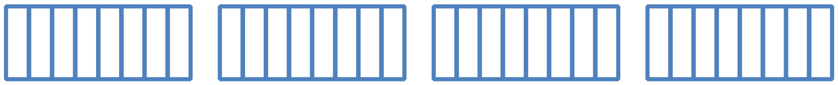


GPUs are often orders of magnitude faster to compute single precision than double precision

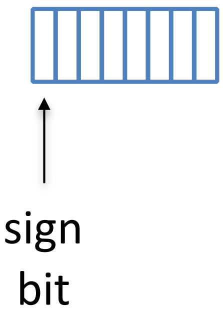
double-precision floating point (float64)

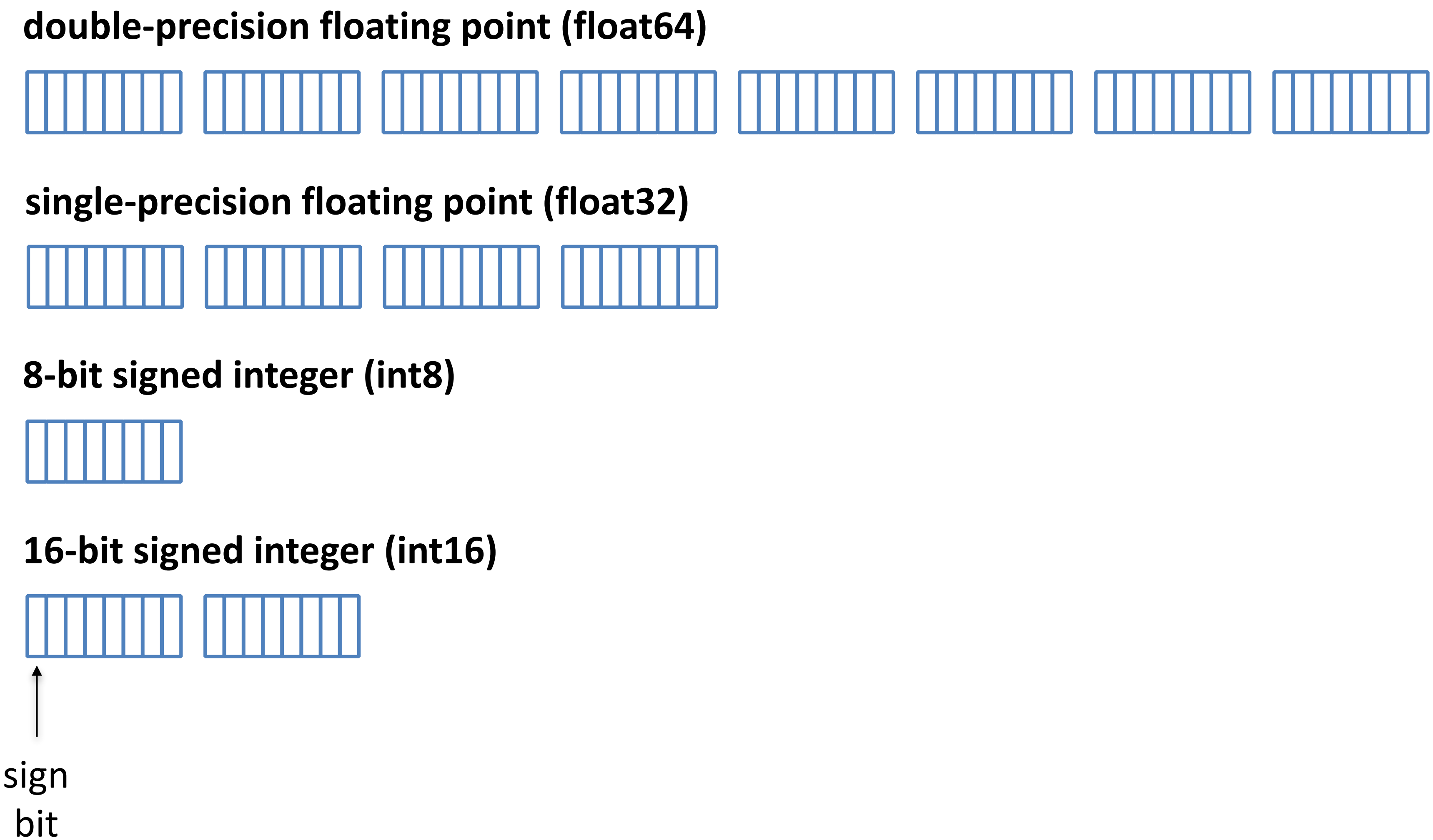


single-precision floating point (float32)

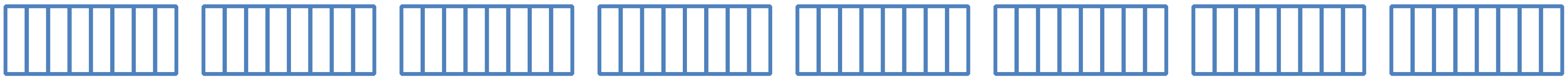


8-bit signed integer (int8)

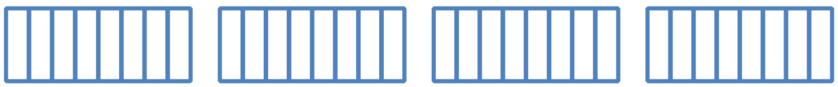



double-precision floating point (float64)
single-precision floating point (float32)
8-bit signed integer (int8)
16-bit signed integer (int16)
sign bit

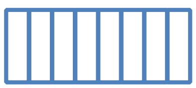
double-precision floating point (float64)



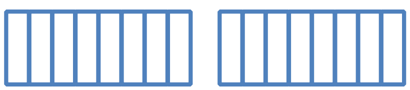
single-precision floating point (float32)



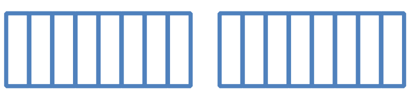
8-bit signed integer (int8)



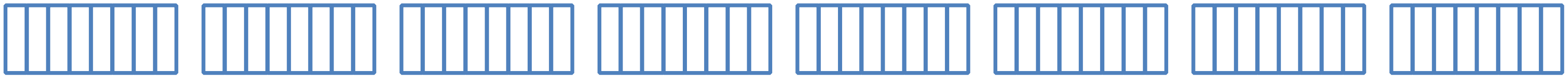
16-bit signed integer (int16)



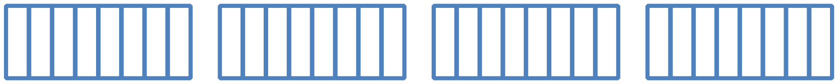
16-bit unsigned integer (uint16)



double-precision floating point (float64)

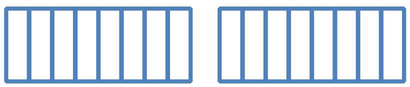


single-precision floating point (float32)

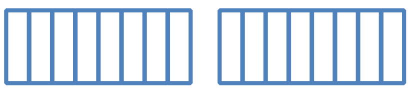


8-bit signed integer (int8)

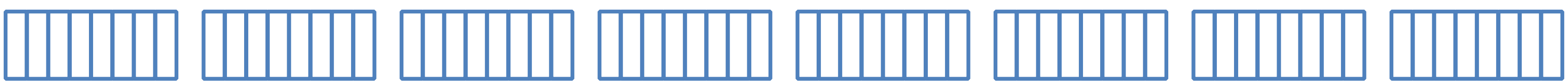
16-bit signed integer (int16)



16-bit unsigned integer (uint16)



64-bit unsigned integer (uint64)



https://www.mathworks.com/help/matlab/numeric-types.html