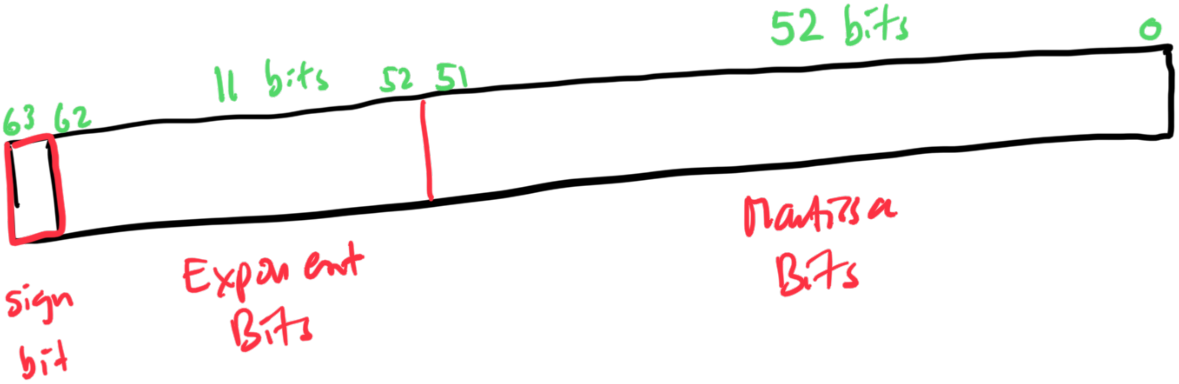
Double Precision Floating Pt.

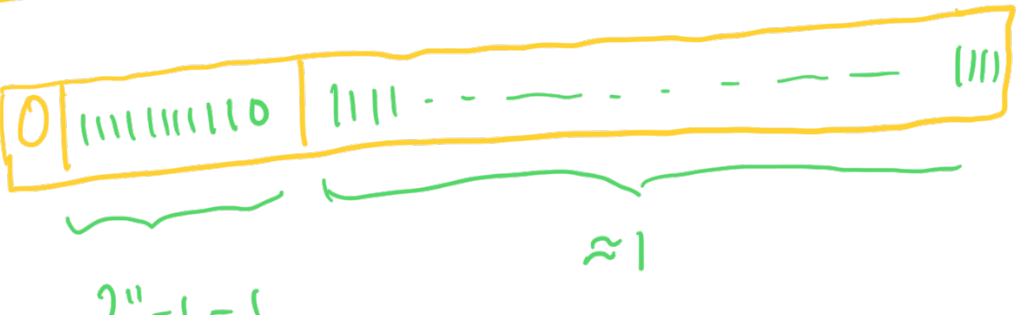
64 bits (IEEE 754)



real value (base 10) $= (-1)^{sign}\left(1+\sum_{i=1}^{52} b_{52-i}\,2^{-i}\right)$

$\times\, 2^{e-1023}$

Largest



$= 2046$

$$r_{max} = 2 \times 2^{2046-1023}$$

$$= 2 \times 2^{1023}$$

$$= 2^{1024}$$

$$\boxed{r_{max} = 1.798 \times 10^{308}}$$

Smallest

0	0000 0000 001	0 - - - - - - - . 0
	= 1	= 1

$$r_{min} = 1.0 \times 2^{1-1023}$$

$$= 1.0 \times 2^{-1022}$$

$$\boxed{r_{min} = 2.23 \times 10^{-308}}$$

Smallest difference between 2 Numbers.

$$\boxed{000 \cdots\cdots 1}$$

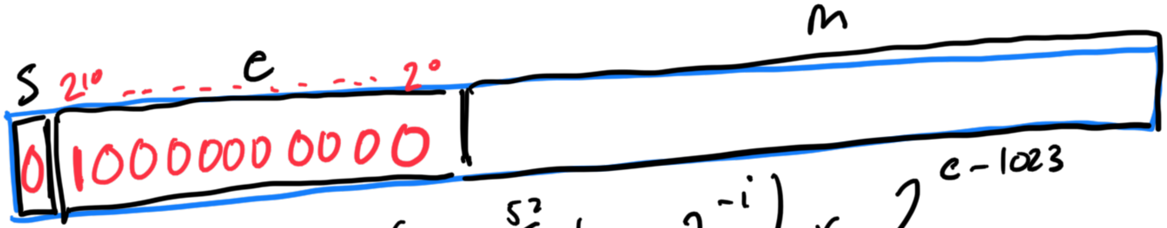
$$= 2^{-52} = \boxed{2.22 \times 10^{-16}}$$

This is why we say that double precision is accurate to 15 decimal places.

How is π stored in double precision?

$$\pi \approx 3.141592653589793$$

(to 16 digits)



$$r = (-1)^s \left(1 + \sum_{i=1}^{52} b_{52-i}\, 2^{-i}\right) \times 2^{e-1023}$$

$e - 1023 = 1 \quad \therefore e = 1024$

$$= \boxed{\quad} \times 2^{\boxed{1}}$$

$\pi = \boxed{}$ exponent must

$(1 \leftrightarrow 2)$

$\pi = \boxed{} \times 2^1$

$= \frac{\pi}{2} = 1 + 0.570796327 \ldots\ldots$

bits of mantissa

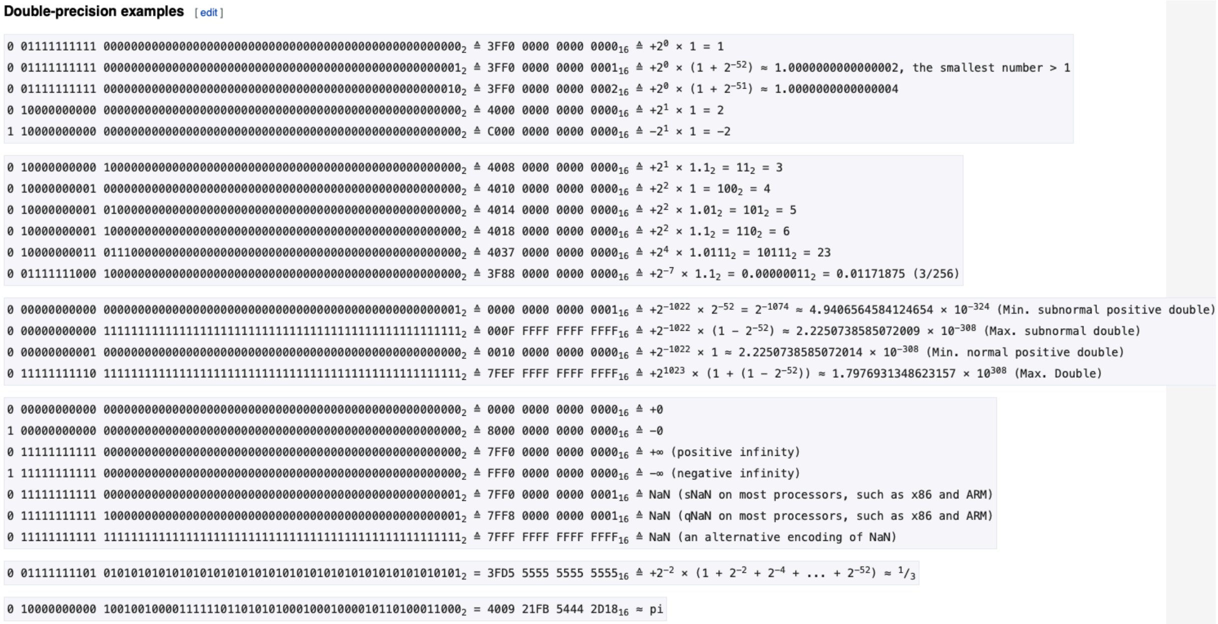
2^{-1}	2^{-2}	2^{-3}	2^{-4}	- - -							
1	0	0	1	0	0	1					
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	- - -		$\frac{1}{128}$					

$$
\begin{array}{l}
0.570796327 \\
-\ 0.5 \\
\hline
\\
-\ \frac{1}{16} \\
\hline
0.008296327 \\
-\ \frac{1}{128} \\
\hline
\end{array}
$$

$0.000483827\ldots$

etc.

1	10000 000000	1001 001 0000 111111 11000



Double-precision examples [edit]

0 01111111111 $0000000000000000000000000000000000000000000000000000_2$ ≙ 3FF0 0000 0000 0000_{16} ≙ $+2^0 \times 1 = 1$
0 01111111111 $0000000000000000000000000000000000000000000000000001_2$ ≙ 3FF0 0000 0000 0001_{16} ≙ $+2^0 \times (1 + 2^{-52}) \approx 1.0000000000000002$, the smallest number > 1
0 01111111111 $0000000000000000000000000000000000000000000000000010_2$ ≙ 3FF0 0000 0000 0002_{16} ≙ $+2^0 \times (1 + 2^{-51}) \approx 1.0000000000000004$
0 10000000000 $0000000000000000000000000000000000000000000000000000_2$ ≙ 4000 0000 0000 0000_{16} ≙ $+2^1 \times 1 = 2$
1 10000000000 $0000000000000000000000000000000000000000000000000000_2$ ≙ C000 0000 0000 0000_{16} ≙ $-2^1 \times 1 = -2$

0 10000000000 $1000000000000000000000000000000000000000000000000000_2$ ≙ 4008 0000 0000 0000_{16} ≙ $+2^1 \times 1.1_2 = 11_2 = 3$
0 10000000001 $0000000000000000000000000000000000000000000000000000_2$ ≙ 4010 0000 0000 0000_{16} ≙ $+2^2 \times 1 = 100_2 = 4$
0 10000000001 $0100000000000000000000000000000000000000000000000000_2$ ≙ 4014 0000 0000 0000_{16} ≙ $+2^2 \times 1.01_2 = 101_2 = 5$
0 10000000001 $1000000000000000000000000000000000000000000000000000_2$ ≙ 4018 0000 0000 0000_{16} ≙ $+2^2 \times 1.1_2 = 110_2 = 6$
0 10000000011 $0111000000000000000000000000000000000000000000000000_2$ ≙ 4037 0000 0000 0000_{16} ≙ $+2^4 \times 1.0111_2 = 10111_2 = 23$
0 01111111000 $1000000000000000000000000000000000000000000000000000_2$ ≙ 3F88 0000 0000 0000_{16} ≙ $+2^{-7} \times 1.1_2 = 0.00000011_2 = 0.01171875$ (3/256)

0 00000000000 $0000000000000000000000000000000000000000000000000001_2$ ≙ 0000 0000 0000 0001_{16} ≙ $+2^{-1022} \times 2^{-52} = 2^{-1074} \approx 4.9406564584124654 \times 10^{-324}$ (Min. subnormal positive double)
0 00000000000 $1111111111111111111111111111111111111111111111111111_2$ ≙ 000F FFFF FFFF FFFF_{16} ≙ $+2^{-1022} \times (1 - 2^{-52}) \approx 2.2250738585072009 \times 10^{-308}$ (Max. subnormal double)
0 00000000001 $0000000000000000000000000000000000000000000000000000_2$ ≙ 0010 0000 0000 0000_{16} ≙ $+2^{-1022} \times 1 \approx 2.2250738585072014 \times 10^{-308}$ (Min. normal positive double)
0 11111111110 $1111111111111111111111111111111111111111111111111111_2$ ≙ 7FEF FFFF FFFF FFFF_{16} ≙ $+2^{1023} \times (1 + (1 - 2^{-52})) \approx 1.7976931348623157 \times 10^{308}$ (Max. Double)

0 00000000000 $0000000000000000000000000000000000000000000000000000_2$ ≙ 0000 0000 0000 0000_{16} ≙ +0
1 00000000000 $0000000000000000000000000000000000000000000000000000_2$ ≙ 8000 0000 0000 0000_{16} ≙ −0
0 11111111111 $0000000000000000000000000000000000000000000000000000_2$ ≙ 7FF0 0000 0000 0000_{16} ≙ +∞ (positive infinity)
1 11111111111 $0000000000000000000000000000000000000000000000000000_2$ ≙ FFF0 0000 0000 0000_{16} ≙ −∞ (negative infinity)
0 11111111111 $0000000000000000000000000000000000000000000000000001_2$ ≙ 7FF0 0000 0000 0001_{16} ≙ NaN (sNaN on most processors, such as x86 and ARM)
0 11111111111 $1000000000000000000000000000000000000000000000000001_2$ ≙ 7FF8 0000 0000 0001_{16} ≙ NaN (qNaN on most processors, such as x86 and ARM)
0 11111111111 $1111111111111111111111111111111111111111111111111111_2$ ≙ 7FFF FFFF FFFF FFFF_{16} ≙ NaN (an alternative encoding of NaN)

0 01111111101 $0101010101010101010101010101010101010101010101010101_2$ = 3FD5 5555 5555 5555_{16} ≙ $+2^{-2} \times (1 + 2^{-2} + 2^{-4} + ... + 2^{-52}) \approx {}^{1}/_{3}$

0 10000000000 $1001001000011111101101010100010001000010110100011000_2$ = 4009 21FB 5444 $\mathrm{2D18}_{16}$ ≈ pi