

Haskell number **Types**

TYPE	explanation	Num	Integral	Fractional	Floating	Real	RealFrac	RealFloat
Int	like its JAVA equivalent	x	x			x		
Word	unsigned Int	x	x			x		
Integer	infinite precision	x	x			x		
Rational	ratio of Integers	x		x		x	x	
Float	like its JAVA equivalent	x		x	x	x	x	x
Double	like its JAVA equivalent	x		x	x	x	x	x

Haskell number **Classes**

CLASS	PROVIDES	REQUIRES
Num	+, -, *, fromInteger	Eq
Integral	div, mod, etc, toInteger	Num, Enum
Fractional	/, recip, fromRational	Num
Floating	pi, exp, acos, etc.	Fractional
Real	toRational	Num, Ord
RealFrac	ceiling, round, etc.,	Real, Fractional
RealFloat	(efficient representation access)	RealFrac

## Haskell number type conversions

from TYPE \ to type	Int	Word	Integer	Rational	Float	Double
Int	id	*	*	*	*	*
Word	*	id	*	*	*	*
Integer	+	+	id	+	+	+
Rational	?	?	?	id	/	/
Float	?	?	?	a	id	x
Double	?	?	?	a	x	id

- To convert *from* Int/Word/Integer to any of the above types you can always use `fromInteger` . `toInteger` (and possibly simplify).
- To convert *from* Rational/Float/Double to any Integral type, you need to choose something like `ceiling`.
- To convert *between* Rational/Float/Double use `fromRational` . `toRational` (and possibly simplify).