From Chapter 11 http://cran.r-project.org/doc/manuals/R-intro.pdf	
Suppose `y`, `x`, `x0`, `x1`, `x2`, … are numeric variables, `X` is a matrix and `A`, `B`, `C`, … are factors.	
`y ~ x`	Both imply the same simple linear regression model of *y* on *x*. The first has an implicit intercept term, and the second an explicit one.
`y ~ 1 + x`	
`y ~ 0 + x`	Simple linear regression of *y* on *x* through the origin (that is, without an intercept term).
`y ~ -1 + x`	
`y ~ x - 1`	
`log(y) ~ x1 + x2`	Multiple regression of the transformed variable, log(y), on *x1* and *x2* (with an implicit intercept term).
`y ~ poly(x,2)`	Polynomial regression of *y* on *x* of degree 2. The first form uses orthogonal polynomials, and the second uses explicit powers, as basis.
`y ~ 1 + x + I(x^2)`	
`y ~ X + poly(x,2)`	Multiple regression *y* with model matrix consisting of the matrix *X* as well as polynomial terms in *x* to degree 2.
`y ~ A`	Single classification analysis of variance model of *y*, with classes determined by *A*.
`y ~ A + x`	Single classification analysis of covariance model of *y*, with classes determined by *A*, and with covariate *x*.
`y ~ A*B`	Two factor non-additive model of *y* on *A* and *B*. The first two specify the same crossed classification and the second two specify the same nested classification. In abstract terms all four specify the same model subspace.
`y ~ A + B + A:B`	
`y ~ B %in% A`	
`y ~ A/B`	
`y ~ (A + B + C)^2`	Three factor experiment but with a model containing main effects and two factor interactions only. Both formulae specify the same model.
`y ~ A*B*C - A:B:C`	
`y ~ A * x`	Separate simple linear regression models of *y* on *x* within the levels of *A*, with different codings. The last form produces explicit estimates of as many different intercepts and slopes as there are levels in *A*.
`y ~ A/x`	
`y ~ A/(1 + x) - 1`	
`y ~ A*B + Error(C)`	An experiment with two treatment factors, *A* and *B*, and error strata determined by factor *C*. For example a split plot experiment, with whole plots (and hence also subplots), determined by factor *C*.
`M^n`	All terms in *M* together with "interactions" up to order *n*
`I(M)`	Insulate *M*. Inside *M* all operators have their normal arithmetic meaning, and that term appears in the model matrix.

Erika Mudrak, Stephen Parry

From http://glmm.wikidot.com/faq	
lmer or glmer functions in lme4 package	
`(1|group)`	Random group intercept
`(x|group)`	Random slope of x within group with correlated intercept
`(1+x|group)`	
`(0+x|group)`	Random slope of x within group: no variation in intercept
`(-1+x|group)`	
`(1|group)+(0+x|group)`	Uncorrelated random intercept and random slope within group
`(1|site/block)`	Intercept varying among sites and among blocks within sites (nested random effects)
`(1|site)+(1|site:block)`	
`(x|site/block)`	Slope and intercept varying among sites and among blocks within sites
`(x|site)+(x|site:block)`	
`(1+x|site)+(1+x|site:block)`	
`(x1|site)+(x2|block)`	Two different effects, varying at different levels
`(1|group1)+(1|group2)`	Intercept varying among crossed random effects

http://lme4.r-forge.r-project.org/book/Ch2.pdf (also a good resource)

glmmPQL in MASS package (can use any exponential family probability distribution)	
lme in nlme package (can only use normal distribution)	
`random=~1|group`	Random group intercept
`random=~x|group`	Random slope of x within group with correlated intercept
`random=~1|site/block`	Intercept varying among sites and among blocks within sites (nested random effects)
`random=~x|site/block`	Slope and intercept varying among sites and among blocks within sites

In glm and glmer	
`offset()`	Incorporates an offset in a GLM model.