

ERUBY

if - <% if \${1:condition} %>
ife - <% if \${1} %>
unless - <% unless \${1} %>
unlesse - <% if \${1} %>

ERUBY-RAILS

conf - <% content_for :
cs - <%= collection_select
ct - <%= content_tag
end - <% end -%>
ff - <% form_for
ffcb - <%= .check_box
ffe - <% error_messages_for
ffff - <%= .file_field
ffhf - <%= .hidden_field
ffl - <%= .label
ffpf - <%= .password_field
ffrb - <%= .radio_button
ffs - <%= .submit
ffta - <%= .text_area
fftf - <%= .text_field
fields - <% fields_for
for - <% for
ft - <% form_tag
il18 - I18n.t('\${1:type.key}')\${2}
it - <%= image_tag
jit - <%= javascript_include_tag
jsit - <%= javascript_include_tag
lia - <%= link_to :action %>
liai - <%= link_to :action :id
lic - <%= link_to :controller
lica - <%= link_to :controller :action
licai - <%= link_to :controller :action :id
lim - <%= link_to <+model+>
lt - <%= link_to
ofcfs - options_from_collection_for_select
rf - <%= render :file
rp - <%= render :partial
rt - <%= render :template
slt - <%= stylesheet_link_tag
sslt - <%= stylesheet_link_tag
st - <%= submit_tag "<+Save

HTML

base - <base href
body - <body id
doctype - <!DOCTYPE
form - <form
h1 - <h1
head - <head>
href - <a href=

ifie - <!--[if IE \${1:6}]>\${2}<![endif]
img - <img src
input - <input type
inputl - <label for=
label - <label for
link - <link rel=
mailto - <a href="mailto:
meta - <meta name=
option - <option
select - <select name="\${1}" id="\${2}">
style - <style type="text/css"
table/basic - <table class="\${1}">
table/hardcore - <table<+ width="<+100%+>" border="<+0+>"
cellspacing="<+0+>" cellpadding="<+5+>"<+>+>>
textarea - <textarea

RUBY

am - alias_method
bm - Benchmark
case - case \${1}
class - class
classe - class \${1:ErrorClassName}
def - def \${1:name}
defi - def initialize\${1}
defmm - def method_missing(meth, *args, &blk)
defs - def self.\${1:class_method_name}
deft - def test_\${1:case_name}
detect - detect { |\${1:element}| \${2:body} }
detecto - detect do |\${1:element}|
dglob - Dir.glob(\${1:"<+dir"+>})
do - do
dop - do |\${1:param}|
ea - each { |\${1:element}| \${2:body} }
each - each { |\${1:element}| \${2:body} }
each_with_index - each_with_index {
each_with_indexo - each_with_index do
eacho - each do |\${1:element}|
eado - each do |\${1:element}|
eawi - each_with_index {
eawido - each_with_index do
elsif - elsif \${1:condition}
fileeach - File.foreach
fopen - File.open
fread - File.read(\${1})
http_get - require 'net/http'
http_post - require 'net/http'
if - if \${1:condition}
ife - if else
inj - inject {
injdo - inject do |\${2:total}, \${3:next}|
inject - inject {
injecto - inject do

```

ip      - ip_addr = request.env['REMOTE_ADDR']
map     - map { |${1:element}| ${2:body} }
mapdo  - map do |${1:element}|
mapo    - map do |${1:element}|
mod     - module
module  - module
nam     - namespace :${1:namespace} do
ope     - open
pathhere - File.join(File.dirname
r       - attr_reader :${1:attr_names}
reject  - reject {
rejecto - reject do
rw      - attr_accessor
select  - select {
selecto - select do
sinc    - class << self; self end
sort    - sort {
sorto   - sort do
tas     - desc "${1:Task description}"
task    - desc "${1:Task description}"
tc      - require "test/unit"
unless  - unless ${1:condition}
unlesse - unless ${1:condition}
w       - attr_writer
while   - while
xmlr    - REXML::Document.new
zip     - zip(${1:enums}) {

```

RUBY-RAILS

```

bf      - before_filter :${1:method}
bt      - belongs_to :${1:association}
crw     - cattr_accessor :${1:attr_names}
defcreate - def create
defdestroy - def destroy
defedit  - def edit
defindex - def index
defnew   - def new
defshow  - def show
defupdate - def update
fina     - find(:all
finf     - find(:first
fini     - find(<+id+>)
finl     - find(:last
flash    - flash
habtm    - has_and_belongs_to_many
hm       - has_many
hmd      - has_many
hmt      - has_many :through
ho       - has_one
ist      - image_submit_tag
log      - RAILS_DEFAULT_LOGGER.${1:debug} ${2}
logd     - logger.debug

```

```

loge    - logger.error
logf    - logger.fatal
logi    - logger.info
logw    - logger.warn
mac     - add_column
mapc    - ${1:map}.${2:connect}
mapca   - ${1:map}.catch_all
mapr    - ${1:map}.resource
maprs   - ${1:map}.resources
mapwo   - ${1:map}.with_options :
mbs     - before_save
mcc     - change_column
mccc    - t.column
mcht    - change_table
mcol    - remove_column
mct     - create_table
migration - class
mp      - map(&:${1:id})
mrc     - remove_column
mrmc    - remove_column
mrnc    - rename_column
mrw     - mattr_accessor
nc      - named_scope
ncl     - named_scope
pa      - params
ra      - render :action
ral     - render :action :layout
rest    - respond_to do |wants|
rf      - render :file
rfu     - render :file :use_full_path
ri      - render :inline
ril     - render :inline :locals
rit     - render :inline :type
rjson   - render :json
rl      - render :layout
rn      - render :nothing
rns     - render :nothing :status
rp      - render :partial
rpc     - render :partial :collection
rpl     - render :partial :locals
rpo     - render :partial :object
rps     - render :partial :status
rt      - render :text
rtl     - render :text :layout
rtlt    - render :text :layout => ${2:true}
rts     - render :text :status
ru      - render :update
rxml    - render :xml
shal    - Digest::SHA1.hexdigest(${1:string})
sweeper - class ${1}Sweeper
tcb     - t.boolean

```

tcbi - t.binary
 tcd - t.decimal
 tcda - t.date
 tcdt - t.datetime
 tcf - t.float
 tch - t.change
 tci - t.integer
 tcl - t.integer
 tcr - t.references
 tcs - t.string
 tct - t.text
 tcti - t.time
 tcts - t.timestamp
 tctss - t.timestamps
 trc - t.remove
 tre - t.rename
 tref - t.references
 va - validates_associated
 vao - validates_acceptance_of
 vc - validates_confirmation_of
 ve - validates_exclusion_of
 vf - validates_format_of
 vi - validates_inclusion_of
 vl - validates_length_of
 vn - validates_numericality_of
 vpo - validates_presence_of
 vu - validates_uniqueness_of
 wants - wants.<+js|xml|html+>
 xdelete - xhr :delete
 xget - xhr :get
 xpost - xhr :post
 xput - xhr :put

RUBY-RSPEC

aft - after({1::each}) do
 annot - any_number_of_times
 anr - and_return({1:value})
 anra - and_raise({1:exception})
 anrb - and_return { {1} }
 ant - and_throw({1:sym})
 any - and_yield({1:values})
 atl - at_least({1:n}).times
 atm - at_most({1:n}).times
 bef - before({1::each}) do
 conn - controller_name :{1:controller}
 des - describe "{1:subject}" do
 desc - require
 descr - describe controller GET|POST...
 dest - describe Type
 dests - describe Type description
 ex - exactly({1:n}).times
 isbl - it_should_behave_like

it - it '{1}' do
 its - it "should
 mat - class {1:ReverseTo}
 mm - mock_model
 moc - mock
 on - once
 resh - require spec_helper
 sef - shared_examples_for
 sh - .should
 shb - .should be
 shbc - .should be_close
 shbio - .should be_instance_of
 shbko - .should be_a_kind_of
 shbr - .should be_redirect
 shbs - .should be_success
 shdm - .should match
 she - .should eql
 sheq - .should equal
 shh - .should have().things
 shhal - .should have_at_least().things
 shham - .should have_at_most().things
 shhr - .should have().records
 shn - .should_not ==
 shnb - .should_not be
 shnbc - .should_not be_close
 shnbio - .should_not be_instance_of
 shnbko - .should_not be_a_kind_of
 shnbr - response.should_not be_redirect
 shnbs - response.should_not be_success
 shne - .should_not eql
 shneq - .should_not equal
 shnm - .should_not match
 shnp - .should_not be_
 shnr - .should_not receive
 shnre - .should_not raise_error
 shnredt - response.should_not redirect_to
 shnrt - .should_not respond_to
 shns - .should_not satisfy { }
 shnt - .should_not throw_symbol
 shp - .should be_
 shr - .should receive
 shre - .should raise_error
 shredt - .should redirect_to
 shrt - .should respond_to
 shs - .should satisfy {
 sht - .should throw_symbel
 shtemp - .should render_template
 tw - twice
 wia - with(