

α	<code>\alpha</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>
β	<code>\beta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>
χ	<code>\chi</code>	ω	<code>\omega</code>	Σ	<code>\Sigma</code>
δ	<code>\delta</code>	Ω	<code>\Omega</code>	τ	<code>\tau</code>
Δ	<code>\Delta</code>	ϕ	<code>\phi</code>	θ	<code>\theta</code>
ϵ	<code>\epsilon</code>	φ	<code>\varphi</code>	ϑ	<code>\vartheta</code>
ε	<code>\varepsilon</code>	Φ	<code>\Phi</code>	Θ	<code>\Theta</code>
η	<code>\eta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ϖ	<code>\varpi</code>	Υ	<code>\Upsilon</code>
Γ	<code>\Gamma</code>	Π	<code>\Pi</code>	ξ	<code>\xi</code>
ι	<code>\iota</code>	ψ	<code>\psi</code>	Ξ	<code>\Xi</code>
κ	<code>\kappa</code>	Ψ	<code>\Psi</code>	ζ	<code>\zeta</code>
λ	<code>\lambda</code>	ρ	<code>\rho</code>		
Λ	<code>\Lambda</code>	ϱ	<code>\varrho</code>		

These commands produce Greek letters suitable for mathematics. You can only use them within a math formula, so if you need a Greek letter within ordinary text you must enclose it in dollar signs (\$). T_EX does not have commands for Greek letters that look like their roman counterparts, since you can get them by using those roman counterparts. For example, you can get a lowercase omicron in a formula by writing the letter ‘o’, i.e., ‘`\rm o`’ or an uppercase beta (‘B’) by writing ‘`\rm B`’.

Don’t confuse the following letters:

- `\upsilon` (‘ υ ’), `\rm v` (‘v’), and `\nu` (‘ ν ’).
- `\varsigma` (‘ ς ’) and `\zeta` (‘ ζ ’).

You can get slanted capital Greek letters by using the math italic (`\mit`) font.

T_EX treats Greek letters as ordinary symbols when it’s figuring how much space to put around them.

Example:

If ρ and θ are both positive, then $f(\theta) - \Gamma_{\theta} < f(\rho) - \Gamma_{\rho}$.

produces:

If ρ and θ are both positive, then $f(\theta) - \Gamma_{\theta} < f(\rho) - \Gamma_{\rho}$.