## **PEL Topics Index**

With PEL you can access these via		s key bindings as well, th			IU Emacs and popular of PEL provides.	- Fundageo.
he <f11> ? e r key sequence.</f11>	<u>Emacs</u>	Calc	<u>Gnus</u>	Magit Cheatsheet	Org	<u>Viper</u>
See <u>S Help/Info</u>	Emacs survival card	Dired	Gnus booklet	Magit Ref-card		VIP
➤ PEL Overview		the <u>PEL file tables</u> . Ea	**			
PEL repo	_	perience, use a browser $\underline{c}$ (version > 78) does that			•	
PEL Readme     PEL Manual		ce, you can browse thro			ormation quickly.	<
• PEL NEWS		r coding and various oth				keys.
General Information.	≽Legend	≻Recommended Ema		≻Themes		
Development Information	≻PEL	■iMenu/Speedbar sı	<u> </u>	■PEL Naming Conv	entions	
Migration Guide	<u>&gt;CRiSP                                    </u>	inventa/opecabar 30	прроте	- LE Naming Conv	<u>Chilons</u>	
OS Desktop Key Bindings (Bindings that don't clash with PEL)	<b>★</b> macOS Fct Keys	<b>≰</b> macOS Keys	Chilbronia de 04 De el			
	- macoo recreeys		<b>16.04 Desk</b>	ttop Keys		
		<b>≰</b> terminal settings	Mint 20 Desktop K	eys		
Feature Comparisons	Completion Modes	Compatibility	§ Speedbar/iMenu	Mode Compatibility	§ Shells/Terminals C	<u>omparisons</u>
Key Prefixes & Suffixes	∑ = Modifier Keys		∑ <b>≡ Numkeypad</b>	<u>≻PEL</u>	<u>=Keys - Fn</u>	<u>=Keys - F11</u>
Emacs Features	The links that start with	only ∑ Emacs generic f	eatures, the blue links a	re external packages. Th	ne green links are mostly	PEL extensions.
A Guided Tour of Emacs.     Awesome-Emacs     MELPA and GNU ELPA	∑ Abbreviations	∑ Diff & Merge	<u>∑ Grep</u>	<u>∑ Marking</u>	∑ Scrolling	T Templates
	∑ Align	<u>∑ Dired</u>	∑ Help/Info	<u>∑ Menus</u>	∑ Search/Replace	∑ Text Modes
The PEL tables named at right describe the Emacs commands and key bindings for generic Emacs concepts and features.  Emacs commands can be executed	∑ Auto-Completion	∑ Display - Lines	∑ Hide/Show	∑ Mode Line	∑ Sessions	∑ Time Tracking
	∑ Autosave/Backup	∑ Drawing	∑ Highlight (colors)	∑ Mouse	∑ start Shells/REPLs	<u> ∑ Transpose</u>
	∑ Bookmarks	∑ Enriched Text	∑ ibuffer-mode	∑ Narrowing	∑ shell-mode	∑X Treemacs
	∑ Buffers	∑ Faces/Fonts	∑ Indentation	∑ Navigation	<u>∑ term-mode</u>	<u>∑ Undo/Redo</u>
by name or bound to key sequences. The commands may have <i>arguments</i>	∑ Case Conversions	<u>∞P Fast Startup</u>	∑ Input Method	∑ Outline	<u>   ℤ vterm-mode</u>	∑ VCS-Git XMagit
and keys can express them.  Emacs Keys  Numeric Arguments You can also: Run Command by Name	∑ Close/Suspend	∑ File-mngt	∑ Inserting Text	∑ Packages	∑X Smartparens	▼ VCS-Mercurial
	∑ Comments	∑ File/Dir Variables	∑ Key-Chords	<u>∑</u> Projectile	∑ Sorting	▼ VCS-Subversion
	∑ Completion/Input	∑ Fill/Justify		<u>∑ Rectangles</u>	<u>∑ Speedbar</u>	<u>∑ Web</u>
Emacs uses a concept of modes:	∑ Counting	<u>∑ Frames</u>	<u>βίχ- Lispy</u>	<u> </u>	∑ Spell Checking	<u>  ▼ Whitespace</u>
<ul> <li>Emacs Major and Minor Modes</li> <li>Major Modes</li> <li>Minor Modes</li> <li>Choosing Modes</li> </ul>	<u>∞M CUA</u>				∑ SyntaxCheck	<u>∑ Windows</u>
	∑ Cursor					<u>∑ Xref</u> - Cross Ref
PEL provides key sequences to	∑ Customize  ∑ Cut & Paste					
toggle minor modes.		arranian Tanting)	of Haaka	rice Consequence		
£\$£ - Emacs Lisp concepts & tools	★ ERT (Emacs Lisp Re	<u> </u>	★ Hooks			a of various systemat
XRef - Cross Reference Tools					chanisms take advantag section. <b>ﷺ</b> This is work	
See also: <u>E Xref</u>	Xref-Support	A Xref-Backend				
		veral build tools but they	, are not all decumented	lin a page		
PEL supports installation and partial	I LE has support for se	verai build tools but they		' "	e user-option is tuned or	
PEL supports installation and partial setup of the following tools:	• Nix Pequires	nix-mode external pac	kage 🔼 activated	WHICH PCI-USC-IIIX-IIIOU		i.
		s <u>nix-mode</u> external pac s <u>tup-mode</u> external pac		when <b>pel-use-tup</b> user-	option is tuned on.	
setup of the following tools:		· · · · · · · · · · · · · · · · · · ·			option is tuned on.	
Build Tools & Preprocessor	• <u>Tup</u> Requires	s <u>tup-mode</u> external page			option is tuned on.	
Build Tools & Preprocessor  Data Serialization	• Tup Requires	s <u>tup-mode</u> external pad भूर - <u>Make</u>			option is tuned on.	
setup of the following tools:	• Tup PRequires  pι - M4  C CWL	s tup-mode external pace  \$\text{\$\end{t}}\$}}}\$}ence one of the constraints of the const	ckage activated		option is tuned on.	
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages	Property Requires  Property Requ	© YAML  © MIB snmp-mode  WHDL ₩future	ckage activated  S YANG	when <b>pel-use-tup</b> user-	option is tuned on.	
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages	PIL - M4  D CWL S ASN.1 asn1-mode  Verilog future M AsciiDoc	© YAML © MIB snmp-mode  VHDL ₩future  M Markdown	© YANG  M Org-Mode		option is tuned on.	
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Graphics Markup	PILP Requires  PIL - M4  D CWL S ASN.1 asn1-mode  Verilog future  M AsciiDoc  M Graphviz Dot	\$\frac{\tup-mode}{\text{p-mode}}\$ external pace \$\frac{\text{p. Make}}{\text{D}}\$ YAML \$\text{S}\$ MIB snmp-mode  VHDL ₩ tuture  M Markdown  M MscGen	© YANG  M Org-Mode M PlantUML	when pel-use-tup user-		
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Graphics Markup  Programming Languages	Proprietable  Pr	© YAML © MIB snmp-mode  VHDL ™future  M Markdown  M MscGen  e support for several pro	© YANG  M Org-Mode M PlantUML  gramming languages. F	M reStructuredText  PEL currently adds extra	support for some of the	n, listed below.
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Graphics Markup  Programming Languages  Main Paradigm of Programming  Language Families	PILP Requires  PIL - M4  D CWL S ASN.1 asn1-mode  Verilog future  M AsciiDoc M Graphviz Dot	\$\frac{\tup-mode}{\text{p-mode}}\$ external pace \$\frac{\text{p. Make}}{\text{D}}\$ YAML \$\text{S}\$ MIB snmp-mode  VHDL ₩ tuture  M Markdown  M MscGen	© YANG  M Org-Mode M PlantUML	when pel-use-tup user-		
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Graphics Markup  Programming Languages  Main Paradigm of Programming  Language Families  Actor Model: (A)  Concatenative (K)	Page 1 Page 1 Page 1 Page 2 Pa	\$\frac{\text{tup-mode}}{\text{pi} \cdot - \text{Make}}\$  \$\text{\text{Q} YAML}\$  \$\text{\text{MIB snmp-mode}}\$  VHDL \$\text{\text{\text{wf}} future}\$  \$\text{\text{M} Markdown}\$  \$\text{\text{M} MscGen}\$  \$\text{e} support for several profunctional Languages}\$  \$\text{Java Virtual Machine}\$	S YANG  M Org-Mode M PlantUML  gramming languages. F  Javascript target  ML Family	M reStructuredText  PEL currently adds extra  Lisp Family Languages  Scheme Language	support for some of their Lisp-like Languages	n, listed below.  Command Line Scripting Language OS App Control
Data Serialization Data Modelling/ Specification Data Modelling/ Data Modelling/ Specification D	Proprieta Propri	\$\frac{\text{tup-mode}}{\text{pi} \text{L-Make}}\$  \$\frac{\text{pi} \text{L-Make}}{\text{pi} \text{YAML}}\$  \$\text{MIB snmp-mode}\$  VHDL \$\text{\text{wfuture}}\$  \$\text{Markdown}\$  \$\text{M.Markdown}\$  \$\text{M.Markdown}\$  \$\text{esupport for several pro}\$  \$\text{Functional}\$  \$\text{Languages}\$  \$\text{Java Virtual Machine}\$  \$\text{Languages}\$	© YANG  M Org-Mode M PlantUML  gramming languages. F  Javascript target  ML Family Languages	M reStructuredText  PEL currently adds extra  Lisp Family Languages	support for some of thei	m, listed below.  Command Line Scripting Language
Data Serialization Data Modelling/ Specification Data Modelling/ Programming Danguages Main Paradigm of Programming Danguage Families Data Model: Actor Model: Actor Model: Concatenative Data Modelling/ Specification	Page 1 Page 1 Page 2 Pa	\$\frac{\text{tup-mode}}{\text{pi} \cdot - \text{Make}}\$  \$\text{\text{Q} YAML}\$  \$\text{\text{MIB snmp-mode}}\$  VHDL \$\text{\text{\text{wf}} future}\$  \$\text{\text{M} Markdown}\$  \$\text{\text{M} MscGen}\$  \$\text{e} support for several profunctional Languages}\$  \$\text{Java Virtual Machine}\$	© YANG  MOrg-Mode MPlantUML  gramming languages. F  Javascript target  ML Family Languages in alphabetical order.	M reStructuredText  PEL currently adds extra  Lisp Family Languages Scheme Language Dialects	support for some of their Lisp-like Languages	n, listed below.  Command Line Scripting Language OS App Control
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Fext Markup Languages  Graphics Markup  Programming Languages  Main Paradigm of Programming  Language Families  - Actor Model: (A)  - Concatenative (K)  - Concurrent: (C)  - Functional: (T) Pure: (F)  - Imperative: (T) or no token  - Object Oriented (C)	Page 1 Page 1 Page 2 Pa	\$\frac{\text{tup-mode}}{\text{pi} \text{L-Make}}\$  \$\frac{\text{pi} \text{L-Make}}{\text{pi} \text{MJL}}\$  \$\frac{\text{MJB snmp-mode}}{\text{viture}}\$  VHDL \$\frac{\text{wifuture}}{\text{future}}\$  \$\frac{\text{M Markdown}}{\text{M Markdown}}\$  \$\text{e support for several pro} \text{Functional} \text{Languages}\$  Java Virtual Machine Languages  programming languages	© YANG  MOrg-Mode MPlantUML  gramming languages. F  Javascript target  ML Family Languages in alphabetical order.	M reStructuredText  PEL currently adds extra  Lisp Family Languages Scheme Language Dialects	support for some of their Lisp-like Languages	n, listed below.  Command Line Scripting Language OS App Control
Data Serialization Data Modelling/ Specification Data Modelling/ Specification Data Markup Languages Fext Markup Languages Graphics Markup Programming Languages Main Paradigm of Programming Language Families - Actor Model: (A) - Concatenative (K) - Concurrent: (C) - Functional: (F) Pure: (F) - Imperative: (I) or no token	Proprieta Propri	© YAML © YAML © MIB snmp-mode  VHDL ™future  M Markdown  M MscGen  e support for several pro  Functional Languages  Java Virtual Machine Languages  a coarse indication of the	© YANG  MOrg-Mode MPlantUML  gramming languages. F  Javascript target  ML Family Languages in alphabetical order. ne programming languages	M reStructuredText  M reStructuredText  Lisp Family Languages Scheme Language Dialects ge family(ies).	support for some of their Lisp-like Languages  Stack Based Languages	m, listed below.  Command Line Scripting Language OS App Control Scripting Language Scala Language
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Graphics Markup  Programming Languages  Main Paradigm of Programming  Language Families  - Actor Model: (A)  - Concatenative (C)  - Functional: (T)  - Imperative: (1) or no token  - Object Oriented (C)  - Has Syntactic Macros: (T)	Programming Languages  Curly Bracket Languages  The following lists the part of the cell colours give	\$\frac{\text{tup-mode}}{\partial \text{Make}}\$  \$\frac{\partial \text{YAML}}{\partial \text{YAML}}\$  \$\frac{\partial \text{MIB snmp-mode}}{\text{Muture}}\$  VHDL \$\frac{\text{Muture}}{\text{Muture}}\$  \$\frac{\text{M Markdown}}{\text{M markdown}}\$  \$\frac{\text{Functional}}{\text{Languages}}\$  Java Virtual Machine Languages  a coarse indication of the second coarse indication of t	S YANG  M Org-Mode M PlantUML  gramming languages. F  Javascript target  ML Family Languages in alphabetical order. ne programming language  PL - Gambit (Fin)	M reStructuredText  PEL currently adds extra  Lisp Family Languages  Scheme Language Dialects  ge family(ies).	support for some of their Lisp-like Languages  Stack Based Languages  Objective-C Wature	m, listed below.  Command Line Scripting Language OS App Control Scripting Language Scala ##future
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Forgramming Languages  Main Paradigm of Programming  Language Families  - Actor Model: (A)  - Concatenative (C)  - Functional: (F) Pure: (F)  - Imperative: (1) or no token  - Object Oriented (2)  - Has Syntactic Macros: (f)	PIUP Requires  PI - M4  D CWL  S ASN.1 asn1-mode  Verilog future  M AsciiDoc  M Graphviz Dot  Emacs has major mode  BEAM Programming Languages  Curly Bracket Languages  The cell colours give  PI - AppleScript  Ada future	\$\frac{\text{stup-mode}}{\partial \text{Make}}\$ external pace \$\frac{\partial \text{YAML}}{\partial \text{YAML}}\$ \$\text{MIB snmp-mode}\$  VHDL \$\text{\text{wfuture}}\$ tuture  M. Markdown  M. MscGen  e support for several pro  Functional Languages  Java Virtual Machine Languages  programming languages a coarse indication of the  \$\frac{\partial \text{MI} - D}{\text{D}}\$ (if \text{A})  Dart \$\text{\text{wfuture}}\$	© YANG  M Org-Mode M PlantUML  gramming languages. F  Javascript target  ML Family Languages in alphabetical order. ne programming language  MI - Gambit  MI - Gerbil  MI - Gerbil  MI - Gerbil	MreStructuredText  MreStructuredText  EL currently adds extra  Lisp Family Languages  Scheme Language Dialects  ge family(ies).  PL - Janet (1) (1) (1)	support for some of their Lisp-like Languages  Stack Based Languages  Objective-C tuture  \$1 - OCaml	m, listed below.  Command Line Scripting Language  OS App Control Scripting Language  Scala ₩ future  PL - Scheme  ↑↑
Data Serialization Data Modelling/ Specification Hardware Description Languages  Text Markup Languages  Text Markup Languages  Text Markup Languages  Main Paradigm of Programming Language Families  - Actor Model: (A)  - Concatenative (K)  - Concurrent: (C)  - Functional: (T) Pure: (E)  - Imperative: (T) or no token  - Object Oriented (C)  - Has Syntactic Macros: (T)  The programming languages supported by PEL are listed here in alphabetical order.  Emacs (and PEL) also provides	PEAM Programming Languages  Curly Bracket Languages  The following lists the position of the cell colours give  PLE - AppleScript  Ada Marture	\$\frac{\text{stup-mode}}{\partial \text{Make}}\$  \$\partial \text{Make}\$  \$\partial \text{MAML}\$  \$\partial \text{MIB snmp-mode}\$  VHDL \$\text{stuture}\$  \$\partial \text{Markdown}\$  \$\partial	SYANG  MOrg-Mode MPlantUML  gramming languages. F  Javascript target  ML Family Languages in alphabetical order. he programming language  \$\text{pi} - \text{Gambit} \text{ fin}  \$\text{pi} - \text{Gerbil} \text{ fin}  \$\text{pi} - \text{GNU Guile} \text{ fin}	M reStructuredText  PEL currently adds extra  Lisp Family Languages  Scheme Language Dialects  ge family(ies).  \$\$\text{\$\texit{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$	support for some of their Lisp-like Languages  Stack Based Languages  Objective-C total of Pascal future	n, listed below.  Command Line Scripting Language OS App Control Scripting Language  Scala to future  PL - Scheme Seed7 to future  Swift to future
Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Forgramming Languages  Main Paradigm of Programming  Language Families  - Actor Model: (A)  - Concatenative (C)  - Concurrent: (C)  - Functional: (T) Pure: (C)  - Imperative: (T) or no token  - Object Oriented (C)  - Has Syntactic Macros: (T)  - The programming languages  supported by PEL are listed here in alphabetical order.  - Emacs (and PEL) also provides  basic support for other  programming languages not listed	PILP Requires  PL-M4  DCWL SASN.1 asn1-mode  Verilog Marture  MasciiDoc Maraphviz Dot  Emacs has major mode  BEAM Programming  Languages  Curly Bracket  Languages  The following lists the parameter of the cell colours give  PLE-Arc  PL-Arc  PL-C  PL-C++	\$\frac{\text{tup-mode}}{\partition{\text{pt} \cdots \text{Make}}}\$  \$\partition{\text{pt} \text{MIL}}{\partition{\text{MIB snmp-mode}}}\$  VHDL \$\text{support for several pro} \text{Punctional Languages}\$  Java Virtual Machine Languages a coarse indication of the pure for the p	MOrg-Mode MPlantUML  gramming languages. F  Javascript target  ML Family Languages in alphabetical order. ne programming language  pt - Gambit fm  pt - Gerbil fm  pt - GNU Guile fm  pt - Gleam  pt - Go	MreStructuredText  PEL currently adds extra  Lisp Family Languages  Scheme Language Dialects  ge family(ies).  \$\text{pi} - Janet	support for some of their Lisp-like Languages  Stack Based Languages  Objective-C tuture  \$\text{1} - OCaml	m, listed below.  Command Line Scripting Language  OS App Control Scripting Language  Scala to future  PL - Scheme Seed7 to future  Swift to future  PL - Tcl to future
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Fext Markup Languages  Fext Markup Languages  Main Paradigm of Programming  Language Families  - Actor Model: (A)  - Concurrent: (C)  - Functional: (F) Pure: (F)  - Imperative: (I) or no token  - Object Oriented co  - Has Syntactic Macros: (III)  The programming languages  supported by PEL are listed here in alphabetical order.  Emacs (and PEL) also provides  basic support for other	PILE APPLESCRIPT  PIL - CHEZ  PIL - M4  PERPLANCE  PASN.1 asn1-mode  PASN.1 asn1-mode  Verilog future  MASCIIDOC  MASCIID	\$\frac{\text{stup-mode}}{\partial \text{Make}}\$  \$\partial \text{Make}\$  \$\partial \text{MAML}\$  \$\partial \text{MIB snmp-mode}\$  VHDL ***future*  \$\partial \text{Markdown}\$  \$\partial Markdow	© YANG  MOrg-Mode MPlantUML  Gramming languages. F  Javascript target  ML Family Languages in alphabetical order. The programming language  PL - Gambit ← PM  PL - Gerbil ← PM  PL - Gleam  PL - Go  Groovy ##future	MreStructuredText  MreStructuredText  PEL currently adds extra  Lisp Family Languages Scheme Language Dialects  ge family(ies).  \$\text{PL} - Janet	Support for some of their Lisp-like Languages  Stack Based Languages  Objective-C the future  PI - OCaml	m, listed below.  Command Line Scripting Language OS App Control Scripting Language  Scala to future  PL - Scheme Seed7 to future Swift to future  PL - Tcl future  PL - Tcl future
Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Fext Markup Languages  Fext Markup Languages  Main Paradigm of Programming  Language Families  Actor Model: (A)  Concatenative (K)  Concurrent: (C)  Functional: (T) Pure: (E)  Imperative: (T) or no token  Object Oriented (C)  Has Syntactic Macros: (T)  The programming languages  supported by PEL are listed here in alphabetical order.  Emacs (and PEL) also provides basic support for other programming languages not listed here.	PIL - M4  D CWL S ASN.1 asn1-mode  Verilog future  M AsciiDoc  M Graphviz Dot  Emacs has major mode  BEAM Programming Languages  Curly Bracket Languages  The following lists the paragram of the cell colours give  PIL - Arc  PIL - Arc  PIL - Chez  PIL - Chibi  PIT - Chibi  PIL -	\$\frac{\text{tup-mode}}{\partition{\text{ptL-mode}}{\text{pture}}}\$ external partitions and the state of the	MOrg-Mode MOrg-Mode MPlantUML  Gramming languages. F  Javascript target  ML Family Languages  in alphabetical order.  The programming languages  in alphabetical order.  The programming languages  It - Gambit fr  PL - Gerbil fr  PL - Gleam  PL - Go  Groovy future  PL - Haskell fr	MreStructuredText  PEL currently adds extra  Lisp Family Languages  Scheme Language Dialects  Ge family(ies).  PI - Janet () () () () Java ****future  PI - Julia () Kotlin ****future  PI - LFE () () () () Lua ****future	Support for some of their Lisp-like Languages  Stack Based Languages  Objective-C tuture  PI - OCaml Pascal future  PI - Perl  PI - Python  PI - Purescript  PI - Racket  PI - Racket	m, listed below.  Command Line Scripting Language  OS App Control Scripting Language  Scala the future  PL - Scheme Seed7 the future  PL - Tcl the future  PL - Tcl the future  PL - Tvpescript the future  PL - UNIX Shell
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Forgramming Languages  Main Paradigm of Programming  Language Families  - Actor Model: (A)  - Concatenative (K)  - Concurrent: (C)  - Functional: (T)  - Pure: (C)  - Imperative: (T)  - Object Oriented co  - Has Syntactic Macros: (T)  - The programming languages  supported by PEL are listed here in alphabetical order.  - Emacs (and PEL) also provides  basic support for other  programming languages not listed here.  Future support for Crystal, Elm,  Kotlin, Lua, Purescript, ReasonML,	PILE Chicken	\$\frac{\text{tup-mode}}{\partial \text{YAML}}\$ \$\frac{\partial \text{YAML}}{\partial \text{YAML}}\$ \$\frac{\partial \text{YAML}}{\partial \text{MIB snmp-mode}}\$  VHDL ****future  M. Markdown  M. MscGen  **e support for several pro  **Functional** Languages  Java Virtual Machine** Languages  Drogramming languages  a coarse indication of the  \$\frac{\partial \text{MIL-D}}{\partial \text{MI-Languages}}\$  Dart ****future  **fifel ****future  \$\partial \text{L-Elim} \text{Minture} \text{F}  \$\partial \text{L-Elim} \text{Minture} \text{F}  \$\partial \text{L-Elim} \text{Minture} \text{F}  \$\partial \text{L-Elim} \text{Minture} \text{C-P}  \$\partial \text{L-Elim} \text{C-P} \text{C-P} \text{A}  \$\frac{\partial \text{L-Elim} \text{C-P} \text{A}  \$\text{L-Elim} \text{C-P} \text{A}  \$\text{L-Elim} \text{C-P} \text{A}	© YANG  MOrg-Mode MPlantUML  Gramming languages. F  Javascript target  ML Family Languages in alphabetical order. ne programming language  PL - Gambit ← ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑	MreStructuredText  MreStructuredText  PEL currently adds extra  Lisp Family Languages Scheme Language Dialects  ge family(ies).  \$\text{PL} - Janet	Support for some of their Lisp-like Languages  Stack Based Languages  Objective-C thruture  PI - OCaml	m, listed below.  Command Line Scripting Language  OS App Control Scripting Language  Scala toture  PL - Scheme Seed7 toture  Swift toture  PL - Tcl toture  PL - Typescript toture  PL - Typescript toture  PL - UNIX Shell  PL - V
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Graphics Markup  Programming Languages  Main Paradigm of Programming  Language Families  Actor Model: A  Concatenative (C)  Functional: Pure: C  Imperative: or no token  Object Oriented co  Has Syntactic Macros: T  The programming languages  supported by PEL are listed here in alphabetical order.  Emacs (and PEL) also provides basic support for other programming languages not listed here.  Future support for Crystal, Elm, Kotlin, Lua, Purescript, ReasonML, Seed7, Typescript, Zig and documentation of support for Ada,	PIL- M4  PREQUITES  PIL- M4  PIL- Chicken  PIL- M4  PIL- Chipure  PIL- M4  PREQUITES  PIL- Chipure  PIL- M4  PREQUITES  PREQUI	\$\frac{\text{tup-mode}}{\partial \text{Make}}\$  \$\partial \text{Make}\$  \$\part	MOrg-Mode MOrg-Mode MPlantUML  Gramming languages. F  Javascript target  ML Family Languages  in alphabetical order.  The programming languages  in alphabetical order.  The programming languages  It - Gambit fr  PL - Gerbil fr  PL - Gleam  PL - Go  Groovy future  PL - Haskell fr	MreStructuredText  MreStructuredText  Lisp Family Languages Scheme Language Dialects  pe family(ies).  pt - Janet	Support for some of their Lisp-like Languages  Stack Based Languages  Objective-C wuture  \$\$\text{\$\e	m, listed below.  Command Line Scripting Language  OS App Control Scripting Language  Scala the future  PL - Scheme Seed7 the future  PL - Tcl the future  PL - Tcl the future  PL - Tvpescript the future  PL - UNIX Shell
Build Tools & Preprocessor  Data Serialization  Data Modelling/ Specification  Hardware Description Languages  Text Markup Languages  Graphics Markup  Programming Languages  Main Paradigm of Programming  Language Families  Actor Model: (A)  Concatenative (K)  Concurrent: (C)  Functional: (F) Pure: (C)  Imperative: (1) or no token  Object Oriented (C)  Has Syntactic Macros: (T)  The programming languages  supported by PEL are listed here in alphabetical order.  Emacs (and PEL) also provides basic support for other programming languages not listed here.  Future support for Crystal, Elm, Kotlin, Lua, Purescript, ReasonML, Seed7, Typescript, Zig and	PIL- M4  PERCONCIONAL REPORTS TO THE PROPERTY OF THE PROPERTY	\$\frac{\text{tup-mode}}{\partial \text{YAML}}\$ \$\frac{\partial \text{YAML}}{\partial \text{YAML}}\$ \$\frac{\partial \text{YAML}}{\partial \text{MIB snmp-mode}}\$  VHDL ****future  M. Markdown  M. MscGen  **e support for several pro  **Functional** Languages  Java Virtual Machine** Languages  Drogramming languages  a coarse indication of the  \$\frac{\partial \text{MIL-D}}{\partial \text{MI-Languages}}\$  Dart ****future  **fifel ****future  \$\partial \text{L-Elim} \text{Minture} \text{F}  \$\partial \text{L-Elim} \text{Minture} \text{F}  \$\partial \text{L-Elim} \text{Minture} \text{F}  \$\partial \text{L-Elim} \text{Minture} \text{C-P}  \$\partial \text{L-Elim} \text{C-P} \text{C-P} \text{A}  \$\frac{\partial \text{L-Elim} \text{C-P} \text{A}  \$\text{L-Elim} \text{C-P} \text{A}  \$\text{L-Elim} \text{C-P} \text{A}	© YANG  MOrg-Mode MPlantUML  Gramming languages. F  Javascript target  ML Family Languages in alphabetical order. ne programming language  PL - Gambit ← ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑	MreStructuredText  MreStructuredText  PEL currently adds extra  Lisp Family Languages Scheme Language Dialects  ge family(ies).  \$\text{PL} - Janet	Support for some of their Lisp-like Languages  Stack Based Languages  Objective-C thruture  PI - OCaml	m, listed below.  Command Line Scripting Language  OS App Control Scripting Language  Scala toture  PL - Scheme Seed7 toture  Swift toture  PL - Tcl toture  PL - Typescript toture  PL - Typescript toture  PL - UNIX Shell  PL - V