Sheet1

		Compilation-only languag					Shell Scripts			Exotic languages		Functional Languages												
		C	C++	Java	Ada		Unix sh	Perl		Python	Ruby	O'Caml	Haskell	Pure LISP	librep	Common LISP	GNU CLISP	E-Lisp	Xemacs LISP	Scheme	Guile	MzScheme		
User Interface																								
REPL		No					Yes			Yes	No	Yes	Hugs	Yes	No									
	Customizable						No	No		U		Yes	Yes	No			Yes	No	No		Yes	Yes		
	Available as a library						No	Yes				No	No	No			No	No	No		Yes	Yes		
GUI		IDE	IDE	IDE	IDE		No			Libraries	Libraries			No				Emacs	Xemacs			MrEd, DrScheme		
	REPL-in-GUI	No								U	No	No	No		No		No	Yes	Yes		No	Yes		
	Toolkit available in the language	Libraries	Libraries	Native	Libraries		No	Libraries		Libraries	Libraries	Yes	Yes/U		Libraries		Libraries	No	Yes		Libraries	Yes		
	Customizable in the language (e.g. on-the-fly new widgets for new datatypes)	Libraries	Libraries	Libraries	Libraries		No	Libraries		Libraries	Libraries	Libraries	U		U		Libraries	No	Yes/U		Libraries	Yes		
Language Features																								
Typing		Yes					Yes(5)	No(6)	Yes	Yes	Yes	Yes	Yes	No	No/U		Yes				Yes			
	Extensible	Yes	Yes	Yes	Yes				Yes	Yes	Yes	Yes	Yes		U		Yes				Yes			
	Extensible with foreign types	Common headers	Common headers	Dynamically	Abstract types				U	Yes	Yes	Abstract types	Abstract types and monads		U		Yes	No/U	Yes		Yes	Yes		
	Dynamic	No							Yes	Yes	No	Yes	Yes		U		Yes				Yes			
	Late (types at eval time)		(1)	(1)					Yes	No	No	No	No		U		Yes				Yes			
	Higher-order functions	Yes(11)	Yes(11,13)	No	Yes(11)		No		(2)	Yes	Yes	Yes												
OOP		No(4)	Yes	Yes	Packages		No					Yes	No	No			CLOS		No	No		GOOPS (CLOS)	Java (22)	
	Classes and objects, methods		Yes	Yes	Yes				Yes(7)	Yes	Yes	Yes												
	Generic functions		Yes	No	Yes				No	U	No	No												
	Multimethods		Yes(2)	No	U				No	No	No	No												
	MOP		No	Yes(20)	No				No	No	No	No											No	
Libraries		Yes					No	Yes				Yes	Yes	No										
	Standard Utilites	Rich	Complete	Complete	Complete				Complete	Rich	Rich	Rich	Rich				Complete				Complete			
	Namespaces	No	Yes	No	U				U	Yes	U	No	No				No					Yes	Yes	
	Modules	No	No(3)	Packages	Yes				Yes	Yes	U	Yes	Yes				Yes				Yes			
Extensibility		Yes																						
	Dynamic modules	dlopen	dlopen	Classes are dynamic	dlopen/U		No	Yes		Yes	Yes	Yes	U	No	Yes		Yes		Yes		Yes			
	Dynamic foreign modules	dlopen	dlopen	JNI	dlopen/U		No	Yes		Yes	Yes	Yes(16)	U	No	Yes/U		No		No		Yes			
	FFI	Yes	Yes	JNI	Yes		Yes(8)	No		Yes	Yes	Yes(17)	Yes/U	No	Yes/U		Yes(23)		No		No		Yes(23)	
	Evaluator available as a library	No					No	Yes		Yes	U	No	No	No	Yes		No		No		Yes			
Compilation		Yes					No			Yes	Yes	Yes	Yes	No							No			
	To Bytecode	No	No	Yes	No				Yes	Yes	U	Yes	Yes/U				Yes		Yes				Yes	
	To Native code	Yes	Yes	Yes(9)	Yes				No	No	U	Yes	Yes				U		No				Yes	
	Special optimizations when compiling	Yes	Yes	Yes	Yes				U	U	U	Yes	Yes				U		Yes				Yes	
	Compilation time	Low	Huge	High	High				Low	Low	Low	Low	Medium				Low		Low				Low	
	Separate Compilation	Yes	Yes(21)	Yes	Package-level				Yes	U	U	Yes	Yes				Yes		Yes				Yes	
	Compiled code run-time performace	High							Good	U	U	High	Good				Good		Good				High	
Extra Features																								
	Anonymous functions	No	Yes(14)	No	No		No	No		Yes	Yes/U	Yes												
	Closures	No	No	No	No		No	Yes(15)		U	Yes/U	Yes		(18)	(18)		(18)		(18)		(18)			
	Reflection	No	No	Yes	No		No	U		No	No	No		Yes										
	Standard operator and function overloading	No	Yes	No	Yes(25)		Command aliases	No		Yes(24)	Yes	No	Yes	Yes										
	Dynamic scoping	No	No	No	No		Yes	Yes		No	No	No	No	Yes							No			
	Continuations	No(19)	No	No	No		No	No		No	U	No	No/U	No							Yes			
	Exceptions	No	Yes	Yes	Yes		No	No		Yes(24)	Yes	Yes	No	No			Yes				Yes			
Popularity Factor		Good	Max	Max	Good	Low	High	Max	Max	Good	Low	High	Good	Low	Max	Low	Max	High	Max	Max	High	Low	High	
Seriousness Factor		High	Max	Max	High	Max	Low	Low	Good	Max	U	Max	High	High	Max	High	Max	High	High	High	High	High	High	

(1) - Type anonymisers can be "emulated" with RTTI
(2) - Through the Visitor design patte
(3) - Can be emulated with design conventions
(4) - Can be emulated with design conventions
(5) - Most of the time
(6) - only character strings are suppo
(8) - command invocation
(9) - Not universally supported
(11) - function pointers
(12) - can be emulated
(13) - With function objects
(14) - With static genericity and function objects
(15) - Rarely used featu
(16) - Recent, unstable
(17) - Wrappers needed
(18) - Non-hygienic
(19) - Can be emulated with setjmp/longjmp
(20) - When using reflection only
(21) - Unless using genericity
(22) - CLOS-like functionality with Meroon
(23) - Non-standard
(24) - Yes for operators, but no general overloading
(25) - except the assignment operator