|  |  |  |
| --- | --- | --- |
| Mark Reason | | |
| Simplicity | | 1 | Readability is sacrificed by simplicity, that means, the code is relatively long than other programming languages. | |
| Orthogonality | | 6 | Control and data structures cannot be built easily. | |
| Data Types | | 1 | Four levels of variable scope ([global](http://en.wikipedia.org/wiki/Global_variable), [class](http://en.wikipedia.org/wiki/Class_variable), [instance](http://en.wikipedia.org/wiki/Instance_variable), and [local](http://en.wikipedia.org/wiki/Local_variable)) denoted by [sigils](http://en.wikipedia.org/wiki/Sigil_(computer_programming)) or the lack thereof. | |
| Syntax Design | | 8 | Succinct and flexible syntax that minimizes [syntactic noise](http://en.wikipedia.org/w/index.php?title=Syntactic_noise&action=edit&redlink=1) and serves as a foundation for [domain-specific languages](http://en.wikipedia.org/wiki/Domain-specific_languages). | |
| Support for Abstraction | | 8 | Ruby is a thoroughly [object- oriented](http://en.wikipedia.org/wiki/Object-oriented) with inheritance mixins and Meta Classes. | |
| Expressivity | | 5 | Ruby is a kind of very expressive language, so a small number of lines for code are able to accomplish high productivity. | |
| Type Checking | | 6 | Ruby supports [dynamic typing](http://en.wikipedia.org/wiki/Dynamic_typing) and [duck typing](http://en.wikipedia.org/wiki/Duck_typing). | |
| Exception Handling | | 3 | Easy to crash with any kind of reason. | |
| Restricted Aliasing | | 1 | It never considers the memory thing so aliasing will not be restricted. | |

**Ruby evaluation table**

**Notice: 10 is full mark**