

Quiz about Namespaces in Clojure, Blaine Mooers yr2022mo07day25hr12min49sec34

1. The `clojure.` namespace provides support for printing data objects.
2. To import only the `split` function from the `clojure.string` namespace and use the `split` function without specifying the namespace, enter (use `'[clojure.string ____ [split]]`)
3. To import the functions of the `clojure.pprint` namespace and use them without specifying their namespace, enter (____ `'clojure.pprint`)
4. The function ____ allows us to import data (or code) from a different namespace and use it in the current namespace without have to state the full namespace.
5. To import the `garden` namespace but exclude the `vegetables` var, enter (`clojure.core/refer 'garden ____ '(vegetables)`)
6. Enter the following ____ to print the documentation for every var whose documentation and name contains `"map"` .
7. Enter the following ____ to print the documentation for the `trim` function in the `clojure.string` namespace.
8. The ____ function allows us to import the functions of a namespace, but they still must be accessed via their full namespace name.
9. The function ____ switches to the specified namespace or creates it if it is missing.
10. EDN stands for ____ data notation.
11. The `clojure.` namespace provides an edn format for time that includes the time zone.
12. To import the `garden` namespace but rename only the `fruits` var with the Polish word `owoce`, enter (`clojure.core/refer 'garden ____ '{fruits owoce}`)
13. Enter the following ____ to print the documentation every var whose documentation contains `"doc"` .
14. To pretty print a table in the REPL with the `print-table` function, enter (`clojure.pprint/____ [{:text "Clojure"} {:text "is"} {:text "great"}]`).
15. The `refer` function has three keywords: `:only`, `:exclude`, and ____.
16. The default namespace is _____. It appears before the prompt.
17. The command (____ `'other-namespace`) switches to the `other-namespace`.
18. To use `pp` as an alias for `clojure.pprint`, enter (`require '[clojure.pprint ____ pp]`)
19. To use `pp` as an alias for `clojure.pprint`, enter (____ `'[clojure.pprint :as pprint]`)
20. To access the `fruits` vector in the `other-namespace` from the `new-namespace`, enter _____.
21. The ____ function implicitly uses `refer` to import the functions of a namespace without needing to access them via their full namespace name.
22. Enter _____ to import into the current namespace the functions from the `new-namespace`; we will still have to use the full namespace name to access the functions.
23. Enter the following ____ to print the list of functions in the `clojure.string` namespace.
24. To import only the `split` function from the `clojure.string` namespace and use the `split` function without specifying the namespace, enter (____ `'[clojure.string :only [split]]`)
25. The `clojure.` namespace provides the main Clojure function like `filter`, `map`, `reduce`, and `count`.
26. Enter the following ____ to print the documentation for the `dir-fn` function in the `clojure.repl` namespace.
27. A _____ is analogous to a package in Java and a module in Python.
28. The `clojure.` namespace provides support for pretty printing of data objects.
29. Enter the following ____ to print the documentation for every var whose documentation and name contains `"data structure"` .
30. The `require` function has a ____ keyword that enables renaming a namespace with a short alias.
31. To import the `garden` namespace but include only the `vegetables` var, enter (`clojure.core/refer 'garden ____ '(vegetables)`)

Answers to quiz about Namespaces in Clojure yr2022mo07day25hr12min49sec34

1. print
2. :only
3. use
4. refer
5. :exclude
6. (find-doc "map")
7. (doc clojure.string/trim)
8. require
9. in-ns
10. extensible
11. edn
12. :rename
13. (find-doc "dir")
14. edn
15. :rename
16. user
17. in-ns
18. :as
19. require
20. other-namespace/fruits
21. use
22. (clojure.core/refer 'new-namespace)
23. (dir clojure.string)
24. use
25. core
26. (doc clojure.string/dir-fn)
27. namespace
28. print
29. (find-doc "data structure")
30. :as
31. :only

References cited in quiz about Namespaces in Clojure yr2022mo07day25hr12min49sec34

1. Fahy2020TheClojureWorkshopANew,InteractiveApproachtoLearningClojure.pdf