elect the statement below that is true. Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The number of key-value pairs The number of key-value pairs The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above		
regunta 2: Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values can contain duplicates. Oboth hash keys and hash values can contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and contain duplicates; hash values cannot contain duplicates. Oboth hash keys and contain duplicates; hash keys cannot contain duplicates. Oboth hash keys and the number contain duplicates. Oboth hash keys and the number contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash keys cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash keys and hash values cannot contain duplicates. Oboth hash values cannot and places. Oboth hash values can contain duplicates. Oboth hash values cannot cont	false	
elect the statement below that is true. Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash values cannot contain duplicates. The number of keys cannot contain duplicates. The number of keys and hash value pairs The number of keys and the number of values The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above	O 14400	
elect the statement below that is true. Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash values cannot contain duplicates. The number of keys cannot contain duplicates. The number of keys and hash value pairs The number of keys and the number of values The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above		
elect the statement below that is true. Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The number of key-value pairs The number of key-value pairs The number of keys and the number of values The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above		
elect the statement below that is true. Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The number of key-value pairs The number of key-value pairs The number of keys and the number of values The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above		
elect the statement below that is true. Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The number of key-value pairs The number of key-value pairs The number of keys and the number of values The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above		
elect the statement below that is true. Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. The state of key value pairs The number of key-value pairs The number of keys and the number of values The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above New Jersey		
Both hash keys and hash values cannot contain duplicates. Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The hash values can contain duplicates; hash keys cannot contain duplicates. The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values Tequnta 4: Why are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above	regunta 2:	atement helps that is true
Both hash keys and hash values can contain duplicates. Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The same of the Jength method return when called on a Hash? The number of key-value pairs The number of keys and the number of values The number of values The number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above		
Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The number of key-value pairs The number of key-value pairs The number of values The number of values The sum of the number of keys and the number of values Tegunta 4: Thy are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax to create a symbol has a string More efficient because symbols have less methods than strings All of the above	O Both ha	sh keys and hash values cannot contain duplicates.
Hash keys can contain duplicates; hash values cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. Hash values can contain duplicates; hash keys cannot contain duplicates. The number of key-value pairs The number of key-value pairs The number of values The number of values The sum of the number of keys and the number of values Tegunta 4: Thy are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax to create a symbol has a string More efficient because symbols have less methods than strings All of the above	O Both ha	sh keys and hash values can contain dunlicates
Tregunta 3: that does the Jength method return when called on a Hash? The number of key-value pairs The number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Tegunta 4: thy are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above regunta 5: states - (0): "New Jersey", Wr. "New York", RY. "Casass")	0 2011110	on togo and man various dan contain capitalists.
regunta 3: The number of key-value pairs The number of keys and the number of values The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Tequnta 4: If you are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above	O Hash ke	eys can contain duplicates; hash values cannot contain duplicates.
regunta 3: The number of key-value pairs The number of keys and the number of values The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Tequnta 4: If you are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above		
# The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above Regunta 5: states - (10: "New Jersey", NY: "New York", NY: "Casses") states (10 New Jersey") p states (100) Kansas New Jersey	Hash va	ılues can contain duplicates; hash keys cannot contain duplicates.
# The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above Regunta 5: states - (10: "New Jersey", NY: "New York", NY: "Casses") states (10 New Jersey") p states (100) Kansas New Jersey		
# The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above Regunta 5: states - (10: "New Jersey", NY: "New York", NY: "Casses") states (10 New Jersey") p states (100) Kansas New Jersey		
# The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above Regunta 5: states - (10: "New Jersey", NY: "New York", NY: "Casses") states (10 New Jersey") p states (100) Kansas New Jersey		
# The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above Regunta 5: states - (10: "New Jersey", NY: "New York", NY: "Casses") states (10 New Jersey") p states (100) Kansas New Jersey		
The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Tegunta 4: Thy are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above regunta 5: states - (10: "New Jersey", NY: "New York", NY: "Kansas") states (10: "New Jersey", NY: "New York", NY: "Kansas") p states (10: "New Jersey", NY: "New York", NY: "Kansas") p states (10: "New Jersey", NY: "New York", NY: "Kansas") New Jersey		
The number of key-value pairs The number of key-value pairs The number of keys The number of values The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Tegunta 4: Thy are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above regunta 5: states - (10: "New Jersey", NY: "New York", NY: "Kansas") states (10: "New Jersey", NY: "New York", NY: "Kansas") p states (10: "New Jersey", NY: "New York", NY: "Kansas") p states (10: "New Jersey", NY: "New York", NY: "Kansas") New Jersey	regunta 3	
The number of keys The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above Tegunta 5: states - (10: "New Jersey", Wi: "New York", KY: "Kanses")		e .length method return when called on a Hash?
The number of keys The number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values The sum of the number of keys and the number of values Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above Tregunta 5: states - (101: "New Jersey", Wi: "New York", KY: "Canses")	♠ The nur	wher of keywalue pairs
The number of values The sum of the number of keys and the number of values regunta 4: they are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above a states - (10: "New Jersey", NY: "New York", NY: "Yansse") states - (10: "New Jersey", NY: "New York", NY: "Yansse") states - (10: "New Jersey", NY: "New York", NY: "Yansse") p states - (10: "New Jersey", NY: "New York", NY: "Yansse") p states - (10: "New Jersey", NY: "New York", NY: "Yansse") p states - (10: "New Jersey", NY: "New York", NY: "Yansse") New Jersey		
The sum of the number of keys and the number of values regunta 4: Why are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above regunta 5: states - (Uz: "New Jersey", W: "New York", Nr: "Kansss") states (Uz: "New Jersey", W: "New York", Nr: "Kansss") states (Uz: "New Jersey", W: "New York", Nr: "Kansss") states (Uz: "New Jersey", W: "New York", Nr: "Kansss") States (Uz: "New Jersey", W: "New York", Nr: "Kansss") New Jersey	O The nur	mber of keys
The sum of the number of keys and the number of values regunta 4: Why are symbols advantageous as hash keys when compared to strings? Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above regunta 5: states - (lu: "key Jersey", W: "key York", K!: "Kasss!")		
regunta 4: //thy are symbols advantageous as hash keys when compared to strings? // Easier syntax to create a symbol than a string // Easier syntax when creating key-value pairs in a hash // More efficient because symbols have less methods than strings // All of the above // Cases - (Uz: 'New Zersey', W: 'New York', KY: 'Kasess') // States - (Uz: New Zersey', W: 'New York', KY: 'Kasess') // States - (Uz: New Zersey', W: 'New York', KY: 'Kasess') // Kaneas // New Jersey	O The nur	nber of values
regunta 4: //thy are symbols advantageous as hash keys when compared to strings? // Easier syntax to create a symbol than a string // Easier syntax when creating key-value pairs in a hash // More efficient because symbols have less methods than strings // All of the above // Cases - (Uz: 'New Zersey', W: 'New York', KY: 'Kasess') // States - (Uz: New Zersey', W: 'New York', KY: 'Kasess') // States - (Uz: New Zersey', W: 'New York', KY: 'Kasess') // Kaneas // New Jersey		
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey	O The sur	n of the number of keys and the number of values
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey		
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey		
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey		
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey		
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey		
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey		
## States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "Rew York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") States - (DD: "New Jersey", Wi: "New York", KY: "Kasss!") New Jersey		
Easier syntax to create a symbol than a string Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings All of the above All of the above States = (00: "New Jersey", Wi: "New York", Ki: "Kanssi")		
Easier syntax when creating key-value pairs in a hash More efficient because symbols have less methods than strings (a) All of the above Fregunta 5: states - (ID: "New Jersey", Wi: "New York", KY: "Kansas")		hols advantaneous as hash kews when compared to strings?
More efficient because symbols have less methods than strings		bols advantageous as hash keys when compared to strings?
More efficient because symbols have less methods than strings	Vhy are sym	
regunta 5: states - (ID: "New Jersey", IW: "New York", IV: "Kanses!") states (ID: "New Jersey", IW: "New York", IV: "Kanses!") p states (IX!) Kaneas New Jersey	Vhy are sym	syntax to create a symbol than a string
regunta 5: states - (ID: "New Jersey", IW: "New York", IV: "Kanses!") states (ID: "New Jersey", IW: "New York", IV: "Kanses!") p states (IX!) Kaneas New Jersey	Vhy are sym	syntax to create a symbol than a string
regunta 5: states - (Uz: "New Jersey", Wz: "New York", KY: "Kanasa") states [:W] - "Kentucky" p states[:XY] Kanasas	Easier s	syntax to create a symbol than a string
states - (DD: "New Jersey", W: "New York", KY: "Kanass!") states[:KY] - "Kentucky"	Easier s	syntax to create a symbol than a string
states - (DD: "New Jersey", W: "New York", KY: "Kanass!") states[:KY] - "Kentucky"	Easier s Easier s More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
states - (DD: "New Jersey", W: "New York", KY: "Kanass!") states[:KY] - "Kentucky"	Easier s Easier s More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
states - (DD: "New Jersey", W: "New York", KY: "Kanass!") states[:KY] - "Kentucky"	Easier s Easier s More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
states - (DD: "New Jersey", W: "New York", KY: "Kanass!") states[:KY] - "Kentucky"	Easier s Easier s More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
states - (DD: 'New Jersey', W: 'New York', KY: 'Kansas') states[:KY] - 'Kentucky' p states[:KY] Kansas New Jersey	Easier s Easier s More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
states - (DD: 'New Jersey', W: 'New York', KY: 'Kansas') states[:KY] - 'Kentucky' p states[:KY] Kansas New Jersey	Easier s Easier s More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
states[stV] - "Xentucky" p states[stV] O Kaneas New Jersey	Easier s Easier s More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
p street[stv] O Kansas O New Jersey	O Easier : Casier : More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings
O New Jersey	Casier : Cas	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings e above
O New Jersey	Casier s Easier s More ef All of th	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings a above s - (0): "New Jersey", Wi: "New York", Ki: "Kansas") 1(xi) - "Kentacky"
	Casier a Easier a Easier a More ef All of th	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings a above s - (0): "New Jersey", Wi: "New York", Ki: "Kansas") 1(xi) - "Kentacky"
	Easier of Easier of More ef	syntax to create a symbol than a string syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings a above s - (0): "New Jersey", Wi: "New York", Ki: "Kansas") 1(xi) - "Kentacky"
	○ Easier s ○ Easier s ○ Easier s ○ More et	syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings a above s - (02: "New 3ersey", Wi: "New York", XY: "Kansss") [[xy] - "Kensuky" tes[[xy]]
	Easier of Easier	syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings a above s - (02: "New 3ersey", Wi: "New York", XY: "Kansss") [[xy] - "Kensuky" tes[[xy]]
	Easier of Easier	syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings a dove s - (02: "New Jersey", NY: "New York", KY: "Kansas") [[xy] - "kentacky" tes[[xy]]
	Easier of Easier	syntax when creating key-value pairs in a hash syntax when creating key-value pairs in a hash ficient because symbols have less methods than strings a bove s - (0): "New Jersey", IVI: "New York", IVI: "Kansas") [[x0] - "Kensucky" tes[[x0]]



¡Buen trabajo! Ya estás listo para pasar a la siguiente clase.

✓ Lo que sabes 6

TRUE or FALSE: A hash is the best data type for storing items in a sequenced order.

Select the statement below that is true.

What does the .length method return when called on a Hash?

Why are symbols advantageous as hash keys when compared to strings?

states = {NJ: "New Jersey", NY: "New York", KY: "Kansas"} states[:KY] = "Kentucky" p states[:KY]