

passes: 838 failures: 0 duration: 8.02s

100%

core.js test suite

Scratch Page

- [Scratch Page](#)

Mocha Test Suite

[\[RUN ALL\]](#)

`_hyperscript bootstrapping`

- ✓ on a single div ▶
- ✓ toggles ▶
- ✓ can target another div ▶
- ✓ hyperscript can have more than one action ▶
- ✓ can wait ▶
- ✓ can change non-class properties ▶
- ✓ can send events ▶
- ✓ can respond to events on other elements ▶
- ✓ can take a class from other elements ▶
- ✓ can set properties ▶
- ✓ can set styles ▶
- ✓ can send events with args ▶
- ✓ can call functions ▶

`the _hyperscript tokenizer`

- ✓ handles basic token types ▶
- ✓ handles whitespace properly ▶
- ✓ handles comments properly ▶
- ✓ handles class identifiers properly ▶
- ✓ handles id references properly ▶
- ✓ handles identifiers properly ▶
- ✓ handles identifiers with numbers properly ▶
- ✓ handles numbers properly ▶
- ✓ handles strings properly ▶
- ✓ handles all special escapes properly ▶
- ✓ handles hex escapes properly ▶
- ✓ handles strings properly 2 ▶
- ✓ handles operators properly ▶
- ✓ handles look ahead property ▶
- ✓ handles template bootstrap properly ▶
- ✓ handles \$ in template properly ▶

the line info parser

- ✓ debug
- ✓ get source works for expressions
- ✓ get source works for statements
- ✓ get line works for statements

passes: 838 failures: 0 duration: 8.02s

100%

the __hyperscript parser

- ✓ basic parse error messages work
- ✓ continues initializing elements in the presence of a parse error
- ✓ can have comments in scripts
- ✓ can have comments in attributes
- ✓ can have alternate comments in scripts
- ✓ can have alternate comments in attributes
- ✓ can have alternate multiline comments in scripts
- ✓ can have multiline comments in attributes
- ✓ can support parenthesized commands and features

__hyperscript regressions

- ✓ can pick detail fields out by name
- ✓ can trigger htmx events
- ✓ can remove class by id
- ✓ can remove by clicks elsewhere
- ✓ me and it is properly set when responding to events
- ✓ me symbol works in from expressions
- ✓ can refer to function in init blocks 41ms
- ✓ can create a paragraph tag
- ✓ async exception
- ✓ return followed by boundary returns an error
- ✓ extra chars cause error when evaling
- ✓ string literals can dot-invoked against
- ✓ button query in form
- ✓ can invoke functions w/ numbers in name
- ✓ listen for event on form

__hyperscript API

- ✓ processNodes does not reinitialize a node already processed

the __hyperscript runtime

- ✓ has proper stack
- ✓ has proper stack from event handler
- ✓ hypertrace is reasonable
- ✓ hypertrace from javascript is reasonable
- ✓ async hypertrace is reasonable 102ms
- ✓ arrays args are handled properly wrt Promises
- ✓ scalar args are handled properly wrt Promises

__hyperscript runtime errors

- ✓ reports basic function invocation null errors properly
- ✓ reports basic function invocation null errors properly w/ possessives
- ✓ reports basic function invocation null errors properly w/ of

- ✓ reports null errors on sets properly
- ✓ reports null errors on settle command properly
- ✓ reports null errors on add command properly
- ✓ reports null errors on remove command properly
- ✓ reports null errors on toggle command properly
- ✓ reports null errors on show command properly
- ✓ reports null errors on hide command properly
- ✓ reports null errors on put command properly
- ✓ reports null errors on transition command properly
- ✓ reports null errors on send command properly
- ✓ reports null errors on trigger command properly
- ✓ reports null errors on increment command properly
- ✓ reports null errors on decrement command properly
- ✓ reports null errors on default command properly
- ✓ reports null errors on measure command properly

passes: 838 failures: 0 duration: 8.02s

100%

security options

- ✓ on a single div

scoping

- ✓ locally scoped variables work
- ✓ locally scoped variables don't clash with built-in variables
- ✓ locally scoped variables do not span features
- ✓ element scoped variables work
- ✓ element scoped variables work w/short syntax
- ✓ element scoped variables support pseudo-possessive syntax
- ✓ element scoped variables span features
- ✓ element scoped variables span features
- ✓ element scoped variables are local only to element
- ✓ global scoped variables work
- ✓ global scoped variables work w/ short syntax
- ✓ setting an element scoped variable spans features
- ✓ setting a global scoped variable spans features
- ✓ basic behavior scoping works
- ✓ behavior scoping is at the element level
- ✓ behavior scoping is isolated from the core element scope
- ✓ behavior scoping is isolated from other behaviors
- ✓ variables are hoisted
- ✓ local variables can override element variables
- ✓ explicit element variable references works
- ✓ global scoped variables span features
- ✓ set favors local variables over global variables

the on feature

- ✓ can respond to events with dots in names
- ✓ can respond to events with colons in names
- ✓ can respond to events with minus in names
- ✓ can respond to events on other elements
- ✓ listeners on other elements are removed when the registering element is removed
- ✓ listeners on self are not removed when the element is removed
- ✓ supports "elsewhere" modifier

- ✓ supports "from elsewhere" modifier
- ✓ can pick detail fields out by name
- ✓ can pick event properties out by name
- ✓ can fire an event on load
- ✓ can be in a top level script tag
- ✓ can have a simple event filter
- ✓ can refer to event properties directly in filter
- ✓ can refer to event detail properties directly in filter
- ✓ can click after a positive event filter
- ✓ multiple event handlers at a time are allowed to execute with the every keyword
- ✓ can have multiple event handlers
- ✓ can have multiple event handlers, no end
- ✓ can queue events 64ms
- ✓ can queue first event 63ms
- ✓ can queue last event 62ms
- ✓ can queue all events 61ms
- ✓ queue none does not allow future queued events
- ✓ can invoke on multiple events
- ✓ can listen for events in another element (lazy)
- ✓ can filter events based on count
- ✓ can filter events based on count range
- ✓ can filter events based on unbounded count range
- ✓ can mix ranges
- ✓ can listen for general mutations 51ms
- ✓ can listen for attribute mutations 58ms
- ✓ can listen for specific attribute mutations 50ms
- ✓ can listen for specific attribute mutations and filter out other attribute mutations 51ms
- ✓ can listen for childList mutations 52ms
- ✓ can listen for childList mutation filter out other mutations 52ms
- ✓ can listen for characterData mutation filter out other mutations 51ms
- ✓ can listen for multiple mutations 51ms
- ✓ can listen for multiple mutations 2 50ms
- ✓ can listen for attribute mutations on other elements 51ms
- ✓ each behavior installation has its own event queue 102ms
- ✓ can catch exceptions thrown in js functions
- ✓ can catch exceptions thrown in hyperscript functions
- ✓ can catch top-level exceptions
- ✓ can catch async top-level exceptions
- ✓ async exceptions don't kill the event queue
- ✓ exceptions in catch block don't kill the event queue
- ✓ uncaught exceptions trigger 'exception' event
- ✓ caught exceptions do not trigger 'exception' event
- ✓ rethrown exceptions trigger 'exception' event
- ✓ basic finally blocks work
- ✓ finally blocks work when exception thrown in catch
- ✓ async basic finally blocks work
- ✓ async finally blocks work when exception thrown in catch
- ✓ async exceptions in finally block don't kill the event queue
- ✓ exceptions in finally block don't kill the event queue
- ✓ can ignore when target doesn't exist
- ✓ can handle an or after a from clause

passes: 838 failures: 0 duration: 8.02s

100%

the def feature

passes: 838 failures: 0 duration: 8.02s

100%

- ✓ can define a basic no arg function
- ✓ can define a basic one arg function
- ✓ functions can be namespaced
- ✓ is called synchronously
- ✓ can call asynchronously
- ✓ can return a value synchronously
- ✓ can exit
- ✓ can return a value asynchronously
- ✓ can interop with javascript
- ✓ can interop with javascript asynchronously
- ✓ can catch exceptions
- ✓ can rethrow in catch blocks
- ✓ can return in catch blocks
- ✓ can catch async exceptions
- ✓ can catch nested async exceptions
- ✓ can rethrow in async catch blocks
- ✓ can return in async catch blocks
- ✓ can install a function on an element and use in children w/ no leak
- ✓ can install a function on an element and use in children w/ return value
- ✓ can install a function on an element and use me symbol correctly
- ✓ finally blocks run normally
- ✓ finally blocks run when an exception occurs
- ✓ finally blocks run when an exception expr occurs
- ✓ async finally blocks run normally
- ✓ async finally blocks run when an exception occurs

the worker feature

- ✓ can define a basic no arg function in a worker 353ms
- ✓ can define a basic one arg function 101ms
- ✓ can call functions from within _hyperscript 151ms
- ✓ can evaluate expressions in worker functions
- ✓ workers can be namespaced
- ✓ can access global scope in worker function
- ✓ can import external scripts

the behavior feature

- ✓ can define behaviors
- ✓ can install behaviors
- ✓ can pass arguments to behaviors
- ✓ supports init blocks in behaviors
- ✓ can pass element arguments to listen to in behaviors
- ✓ can refer to arguments in init blocks
- ✓ can declare variables in init blocks

The (top-level) js feature

- ✓ can run js at the top level
- ✓ can expose globals
- ✓ can expose functions
- ✓ can hide functions
- ✓ does not expose variables

the init feature

- ✓ can define an init block inline
- ✓ can define an init block in a script
- ✓ can initialize immediately

passes: 838 failures: 0 duration: 8.02s

100%

the set feature

- ✓ can define variables with let at the element level

the socket feature

- ✓ can handle different url schemes

the add command

- ✓ can add class ref on a single div
- ✓ can add class ref w/ double dash on a single div
- ✓ can add class ref on a single form
- ✓ can target another div for class ref
- ✓ can add to query in me
- ✓ can add to children
- ✓ can add non-class attributes
- ✓ can add css properties
- ✓ can add templated css properties
- ✓ can add multiple class refs
- ✓ can add class refs w/ colons and dashes
- ✓ can filter class addition via the when clause
- ✓ can filter property addition via the when clause

the append command

- ✓ can append a string to another string
- ✓ can append a value into an array
- ✓ can append a value to 'it'
- ✓ can append a value to a DOM node
- ✓ can append a value to a DOM element
- ✓ can append a value to I
- ✓ can append a value to an object property
- ✓ multiple appends work
- ✓ append to undefined ignores the undefined

the async command

- ✓ is async
- ✓ can trigger an event on the original element

the remove command

- ✓ can remove class ref on a single div
- ✓ can remove class ref on a single form
- ✓ can target another div for class ref
- ✓ can remove non-class attributes
- ✓ can remove elements
- ✓ can remove other elements
- ✓ can remove parent element

- ✓ can remove multiple class refs
- ✓ can remove query refs from specific things

passes: 838 failures: 0 duration: 8.02s

100%

the toggle command

- ✓ can toggle class ref on a single div
- ✓ can toggle class ref on a single form
- ✓ can target another div for class ref toggle
- ✓ can toggle non-class attributes
- ✓ can toggle for a fixed amount of time
- ✓ can toggle until an event on another element
- ✓ can toggle between two classes
- ✓ can toggle multiple class refs
- ✓ can toggle display
- ✓ can toggle opacity
- ✓ can toggle opacity
- ✓ can toggle display w/ my
- ✓ can toggle display w/ my
- ✓ can toggle display w/ my
- ✓ can toggle display on other elt
- ✓ can toggle display on other elt
- ✓ can toggle display on other elt

the hide command

- ✓ can hide element with no target
- ✓ hide element then show element retains original display
- ✓ can hide element with no target followed by command
- ✓ can hide element with no target followed by then
- ✓ can hide element with no target with a with
- ✓ can hide element, with display:none by default
- ✓ can hide element with display:none explicitly
- ✓ can hide element with opacity:0
- ✓ can hide element with opacity style literal
- ✓ can hide element, with visibility:hidden
- ✓ can hide other elements
- ✓ can hide with custom strategy
- ✓ can set default to custom strategy

the show command

- ✓ can show element, with display:block by default
- ✓ can show form, with display:block by default
- ✓ can show element with display:block explicitly
- ✓ can show element with custom display value
- ✓ can show element with inline-block display value
- ✓ can show element with opacity:1
- ✓ can show element with opacity style literal
- ✓ can show element, with visibility:visible
- ✓ can show other elements
- ✓ can show multiple elements with inline-block display value
- ✓ can show multiple elements as class with inline-block display value
- ✓ can use a when clause to show or hide an element
- ✓ can use a when clause and a with clause to show or hide an element

- ✓ can filter over a set of elements using the its symbol
- ✓ starting off with display none does not stick

passes: 838 failures: 0 duration: 8.02s

100%

the wait command

- ✓ can wait on time
- ✓ can wait on event
- ✓ waiting on an event sets 'it' to the event
- ✓ can destructure properties in a wait
- ✓ can wait on event on another element
- ✓ can wait on event or timeout 1
- ✓ can wait on event or timeout 2

the send command

- ✓ can send events
- ✓ can reference sender in events
- ✓ can send events with args
- ✓ can send events with dots
- ✓ can send events with dots with args
- ✓ can send events with colons
- ✓ can send events with colons with args
- ✓ can send events to any expression

the trigger command

- ✓ can trigger events
- ✓ can trigger events with args
- ✓ can trigger events with dots
- ✓ can trigger events with dots with args
- ✓ can trigger events with colons
- ✓ can trigger events with dots with colons

the take command

- ✓ can take a class from other elements
- ✓ can take a class from other forms
- ✓ can take a class for other elements
- ✓ a parent can take a class for other elements
- ✓ can take an attribute from other elements
- ✓ can take an attribute with specific value from other elements
- ✓ can take an attribute value from other elements and set specific values instead
- ✓ can take an attribute value from other elements and set value from an expression instead
- ✓ can take an attribute for other elements
- ✓ a parent can take an attribute for other elements
- ✓ can take multiple classes from other elements
- ✓ can take multiple classes from specific element

the log command

- ✓ can log single item
- ✓ can log multiple items
- ✓ can log multiple items with debug
- ✓ can log multiple items with error

the call command

- ✓ can call javascript instance functions
- ✓ can call global javascript functions
- ✓ can call no argument functions
- ✓ can call functions w/ underscores
- ✓ can call functions w/ dollar signs
- ✓ call functions that return promises are waited on

passes: 838 failures: 0 duration: 8.02s

100%

the make command

- ✓ can make objects
- ✓ can make named objects
- ✓ can make named objects w/ global scope
- ✓ can make objects with arguments
- ✓ can make named objects with arguments
- ✓ can make elements
- ✓ can make elements with id and classes
- ✓ creates a div by default

pseudoCommands

- ✓ Basic instance function with expression
- ✓ Basic instance function with expression and with
- ✓ Basic instance function with expression and on
- ✓ Basic instance function with me target
- ✓ Can use functions defined outside of the current element
- ✓ Basic instance function with me target no preposition
- ✓ functions defined alongside can be invoked
- ✓ Can use indirect functions with a symbol root
- ✓ Can use indirect functions with a function root
- ✓ Can use nested indirect functions with a symbol root
- ✓ non-function pseudo-command is an error

the put command

- ✓ can set properties
- ✓ can put directly into nodes
- ✓ can put nodes into nodes
- ✓ can put directly into symbols
- ✓ me symbol doesn't get stomped on direct write
- ✓ can set styles
- ✓ can set javascript globals
- ✓ can set into class ref w/ flatmapped property
- ✓ can set into class ref w/ flatmapped property using of
- ✓ can set local variables
- ✓ can set into id ref
- ✓ can insert before
- ✓ can insert after
- ✓ can insert after beginning
- ✓ can insert before end
- ✓ can set into attribute ref
- ✓ can set into indirect attribute ref
- ✓ can set into indirect attribute ref 2
- ✓ can set into indirect attribute ref 3

- ✓ can set into style ref
- ✓ can set into indirect style ref
- ✓ can set into indirect style ref 2
- ✓ can set into indirect style ref 3
- ✓ waits on promises
- ✓ can put properties w/ array access syntax
- ✓ can put properties w/ array access syntax and var
- ✓ can put array vals w/ array access syntax
- ✓ can put array vals w/ array access syntax and var
- ✓ properly processes hyperscript in new content in a symbol write
- ✓ properly processes hyperscript in new content in a element target
- ✓ properly processes hyperscript in before
- ✓ properly processes hyperscript at start of
- ✓ properly processes hyperscript at end of
- ✓ properly processes hyperscript after

passes: 838 failures: 0 duration: 8.02s

100%

the set command

- ✓ can set properties
- ✓ can set indirect properties
- ✓ can set complex indirect properties lhs
- ✓ can set complex indirect properties rhs
- ✓ can set chained indirect properties
- ✓ can set styles
- ✓ can set javascript globals
- ✓ can set local variables
- ✓ can set into id ref
- ✓ can set into class ref
- ✓ can set into attribute ref
- ✓ can set into indirect attribute ref
- ✓ can set into indirect attribute ref 2
- ✓ can set into indirect attribute ref 3
- ✓ can set into style ref
- ✓ can set into indirect style ref
- ✓ can set into indirect style ref 2
- ✓ can set into indirect style ref 3
- ✓ set waits on promises
- ✓ can set many properties at once with object literal
- ✓ can set props w/ array access syntax
- ✓ can set props w/ array access syntax and var
- ✓ can set arrays w/ array access syntax
- ✓ can set arrays w/ array access syntax and var
- ✓ handles set url regression properly

the default command

- ✓ can default variables
- ✓ can default attributes
- ✓ can default properties
- ✓ default variables respect existing values
- ✓ default attributes respect existing values
- ✓ default properties respect existing values

the if command

passes: 838 failures: 0 duration: 8.02s

100%


- ✓ basic true branch works
- ✓ basic true branch works with multiple commands
- ✓ basic true branch works with end
- ✓ basic true branch works with naked else
- ✓ basic true branch works with naked else end
- ✓ basic else branch works
- ✓ basic else branch works with end
- ✓ basic else if branch works
- ✓ basic else if branch works with end
- ✓ otherwise alias works
- ✓ triple else if branch works
- ✓ triple else if branch works with end
- ✓ basic else branch works with multiple commands
- ✓ true branch with a wait works
- ✓ false branch with a wait works
- ✓ if properly passes execution along if child is not executed
- ✓ if properly supports nested if statements and end block
- ✓ if on new line does not join w/ else
- ✓ passes the sieve test

the repeat command

- ✓ basic for loop works
- ✓ basic for loop with null works
- ✓ waiting in for loop works 51ms
- ✓ basic raw for loop works
- ✓ basic raw for loop works
- ✓ waiting in raw for loop works 41ms
- ✓ repeat forever works
- ✓ repeat forever works w/o keyword
- ✓ basic in loop works
- ✓ index syntax works
- ✓ indexed by syntax works
- ✓ by syntax works
- ✓ while keyword works
- ✓ until keyword works
- ✓ until event keyword works
- ✓ only executes the init expression once
- ✓ can nest loops
- ✓ basic times loop works
- ✓ times loop with expression works
- ✓ loop continue works
- ✓ loop break works

the fetch command

- ✓ can do a simple fetch 53ms
- ✓ can do a simple fetch w/ a naked URL 50ms
- ✓ can do a simple fetch w/ html 50ms
- ✓ can do a simple fetch w/ json 51ms
- ✓ can do a simple fetch w/ json using Object syntax 51ms
- ✓ can do a simple fetch w/ json using Object syntax and an 'an' prefix 52ms

✓ can do a simple fetch with a response object	52ms	passes: 838 failures: 0 duration: 8.02s	
✓ can do a simple fetch w/ a custom conversion	51ms		
✓ can do a simple post	52ms		
✓ can do a simple post alt syntax without curlies	51ms		
✓ can do a simple post alt syntax w/ curlies	51ms		
✓ can put response conversion after with	51ms		
✓ can put response conversion before with	51ms		
✓ triggers an event just before fetching	51ms		
✓ submits the fetch parameters to the event handler	51ms		
✓ allows the event handler to change the fetch parameters	51ms		
✓ can catch an error that occurs when using fetch	51ms		

the throw command

✓ can throw a basic exception	
✓ can throw an async exception	
✓ async exceptions propagate properly	
✓ async exceptions as throws propagate properly	
✓ exceptions propagate from a worker	
✓ async exceptions propagate from a worker	
✓ exceptions propagate from a worker through a function	
✓ exceptions propagate from an async worker through a function	
✓ exceptions propagate from an async worker through an async function	
✓ can throw inside an event handler	
✓ can respond to exceptions in an event handler with an event handler	
✓ can respond to async exceptions in an event handler with an event handler	

The (inline) js command

✓ can run js	
✓ can deal with empty input list	
✓ can access values from _hyperscript	
✓ can return values to _hyperscript	
✓ can do both of the above	

the transition command

✓ can transition a single property on current element	
✓ can transition with parameterized values	
✓ can transition a single property on form	
✓ can transition a single property on current element with the my prefix	
✓ can transition two properties on current element	
✓ can transition on another element	
✓ can transition on another element no element prefix	
✓ can transition on another element no element prefix + possessive	
✓ can transition on another element no element prefix with it	
✓ can transition with a custom transition time	
✓ can transition with a custom transition time via the over syntax	
✓ can transition a single property on current element using style ref	
✓ can transition a single property on form using style ref	
✓ can transition a single property on current element with the my prefix using style ref	
✓ can use initial to transition to original value	

the measure command

- ✓ can measure me
- ✓ can measure another element
- ✓ can assign measurements to locals
- ✓ can measure all the supported properties

passes: 838 failures: 0 duration: 8.02s

100%

the pick command

- ✓ can pick items from an array
- ✓ can pick a single item from an array
- ✓ can use 'end' when picking items from an array
- ✓ can use 'start' when picking items from an array
- ✓ can use negative indices when picking items from an array
- ✓ can pick items from a string
- ✓ can pick a single item from a string
- ✓ can use 'end' when picking items from a string
- ✓ can use 'start' when picking items from a string
- ✓ can use negative indices when picking items from a string
- ✓ can pick a single regex match
- ✓ can pick all regex matches

the tell command

- ✓ establishes a proper beingTold symbol
- ✓ does not overwrite the me symbol
- ✓ works with an array
- ✓ restores a proper implicit me symbol
- ✓ ignores null
- ✓ you symbol represents the thing being told
- ✓ your symbol represents the thing being told
- ✓ attributes refer to the thing being told
- ✓ yourself attribute also works
- ✓ tell terminates with a feature

the settle command

- ✓ can settle me no transition 1001ms
- ✓ can settle target no transition 1000ms

the unless command modifier

- ✓ unless modifier can conditionally execute a command

the increment command

- ✓ can increment an empty variable
- ✓ can increment a variable
- ✓ can increment refer to result
- ✓ can increment an attribute
- ✓ can increment an floating point numbers
- ✓ can increment a property
- ✓ can increment by zero
- ✓ can increment a value multiple times
- ✓ can decrement an empty variable
- ✓ can decrement a variable
- ✓ can decrement an attribute

- ✓ can decrement an floating point numbers
- ✓ can decrement a property
- ✓ can decrement a value multiple times
- ✓ can decrement by zero

passes: 838 failures: 0 duration: 8.02s

100%

the string expression

- ✓ handles strings properly
- ✓ string templates work properly
- ✓ string templates work w/ props
- ✓ string templates work w/ props w/ braces
- ✓ string templates work properly w braces
- ✓ string templates preserve white space
- ✓ should handle strings with tags and quotes
- ✓ should handle back slashes in non-template content

the string postfix expression

- ✓ handles basic postfix strings properly
- ✓ handles basic postfix strings with spaces properly
- ✓ handles expression roots properly

the number expression

- ✓ handles numbers properly

the idRef expression

- ✓ basic id ref works
- ✓ basic id ref works w no match
- ✓ template id ref works
- ✓ id ref works from a disconnected element

the classRef expression

- ✓ basic classRef works
- ✓ basic classRef works w no match
- ✓ dashed class ref works
- ✓ colon class ref works
- ✓ multiple colon class ref works
- ✓ template classRef works
- ✓ leading minus class ref works

the closest expression

- ✓ basic query return values
- ✓ parent modifier works
- ✓ attributes resolve as attributes
- ✓ attributes can be looked up and referred to in same expression
- ✓ attributes can be set via the closest expression
- ✓ parenthesizing allows you to nest to modifiers properly
- ✓ attributes can be set via the closest expression
- ✓ returns an array where appropriate

the attributeRef expression

- ✓ attributeRef with no value works ▶
- ✓ attributeRef with dashes name works ▶
- ✓ attributeRef can be set as symbol ▶
- ✓ attributeRef can have value used in add commands ▶
- ✓ attributeRef can have value in quotes used in add commands ▶
- ✓ attributeRef can have value in quotes with spaces used in add commands ▶
- ✓ attributeRef can be set as prop ▶
- ✓ attributeRef can be set through possessive ▶
- ✓ attributeRef can be set indirectly ▶
- ✓ attributeRef can be put indirectly ▶
- ✓ attributeRef can be put as symbol ▶
- ✓ attributeRef with no value works w/ short syntax ▶
- ✓ attributeRef with dashes name works w/ short syntax ▶
- ✓ attributeRef can be set as symbol w/ short syntax ▶
- ✓ attributeRef can be set as prop w/ short syntax ▶
- ✓ attributeRef can be set through possessive w/ short syntax ▶
- ✓ attributeRef can be set indirectly w/ short syntax ▶
- ✓ attributeRef can be put indirectly w/ short syntax ▶
- ✓ attributeRef can be put as symbol w/ short syntax ▶
- ✓ attributeRef can have value used in add commands w/ short syntax ▶
- ✓ attributeRef can have value in quotes used in add commands w/ short syntax ▶
- ✓ attributeRef can have value in quotes with spaces used in add commands w/ short syntax ▶

passes: 838 failures: 0 duration: 8.02s

100%

the symbol expression

- ✓ resolves local context properly ▶
- ✓ resolves global context properly ▶

the objectLiteral expression

- ✓ empty object literals work ▶
- ✓ one field object literal works ▶
- ✓ multi-field object literal works ▶
- ✓ strings work in object literal field names ▶
- ✓ hyphens work in object literal field names ▶
- ✓ expressions work in object literal field names ▶

the boolean literal expression

- ✓ true boolean literals work ▶
- ✓ false boolean literals work ▶

the null literal expression

- ✓ null literal work ▶

the arrayLiteral expression

- ✓ empty array literals work ▶
- ✓ one element array literal works ▶
- ✓ multi element array literal works ▶

the blockLiteral expression

- ✓ basic block literals work ▶
- ✓ basic identity works ▶

- ✓ basic two arg identity works
- ✓ can map an array

passes: 838 failures: 0 duration: 8.02s

100%

the not expression

- ✓ not inverts true
- ✓ not inverts false
- ✓ two nots make a true

the no expression

- ✓ no returns true for null
- ✓ no returns false for non-null
- ✓ no returns true for empty array
- ✓ no returns true for empty selector
- ✓ no returns false for non-null

the some expression

- ✓ some returns false for null
- ✓ some returns true for non-null
- ✓ some returns false for empty array
- ✓ some returns false for empty selector
- ✓ some returns true for nonempty selector
- ✓ some returns true for filled array

the in expression

- ✓ basic no query return values
- ✓ basic query return values
- ✓ id returns values
- ✓ id template returns values
- ✓ class returns values
- ✓ class template returns values
- ✓ query returns values
- ✓ query template returns values
- ✓ in expression binds to unaryOperators

the mathOperator expression

- ✓ addition works
- ✓ string concat works
- ✓ subtraction works
- ✓ multiplication works
- ✓ division works
- ✓ mod works
- ✓ addition works w/ more than one value
- ✓ unparenthesized expressions with multiple operators cause an error
- ✓ parenthesized expressions with multiple operators work
- ✓ can use mixed expressions 40ms

the logicalOperator expression

- ✓ and works
- ✓ or works
- ✓ and works w/ more than one value

✓ unparenthesized expressions with multiple operators cause an error ▶

✓ parenthesized expressions with multiple operators work ▶

passes: 838 failures: 0 duration: 8.02s

100%

the comparisonOperator expression

✓ less than works ▶

✓ less than or equal works ▶

✓ greater than works ▶

✓ greater than or equal works ▶

✓ equal works ▶

✓ triple equal works ▶

✓ not equal works ▶

✓ triple not equal works ▶

✓ is works ▶

✓ equals works ▶

✓ is equal to works ▶

✓ is really equal to works ▶

✓ really equals works ▶

✓ is not works ▶

✓ is not equal to works ▶

✓ is not really equal to works ▶

✓ is in works ▶

✓ is not in works ▶

✓ I am in works ▶

✓ I am not in works ▶

✓ match works ▶

✓ does not match works ▶

✓ match works w/ strings ▶

✓ does not match works w/ strings ▶

✓ contains works with elts ▶

✓ contains works with arrays ▶

✓ contains works with css literals ▶

✓ include works ▶

✓ includes works with arrays ▶

✓ includes works with css literals ▶

✓ does not contain works ▶

✓ is empty works ▶

✓ is not empty works ▶

✓ is a works ▶

✓ is not a works ▶

✓ english less than works ▶

✓ english less than or equal works ▶

✓ english greater than works ▶

✓ english greater than or equal works ▶

✓ exists works ▶

✓ does not exist works ▶

the typecheck expression

✓ can do basic string typecheck ▶

✓ can do null as string typecheck ▶

✓ can do basic non-string typecheck failure ▶

✓ can do basic string non-null typecheck ▶

✓ null causes null safe string check to fail ▶

function call expressions

- ✓ can invoke global function
- ✓ can invoke function on object
- ✓ can invoke global function w/ async arg 43ms
- ✓ can invoke function on object w/ async arg
- ✓ can invoke function on object w/ async root & arg

passes: 838 failures: 0 duration: 8.02s

100%

the cookies identifier

- ✓ basic set cookie values work
- ✓ update cookie values work
- ✓ basic clear cookie values work
- ✓ iterate cookies values work

the async expression works

- ✓ simple async expression works
- ✓ async argument works w/ non-async value
- ✓ async argument works w/ async value

the queryRef expression

- ✓ basic queryRef works
- ✓ basic queryRef works w/ multiple matches
- ✓ basic queryRef works w/ properties
- ✓ basic queryRef works w/ funny selector
- ✓ basic queryRef works w/ div selector
- ✓ basic queryRef works w no match
- ✓ basic queryRef works w properties w/ strings
- ✓ queryRef w/ \$ works
- ✓ queryRef w/ \$ no curlies works
- ✓ can interpolate elements into queries
- ✓ queryRefs support colons properly

as operator

- ✓ converts null as null
- ✓ converts value as String
- ✓ converts value as Int
- ✓ converts value as Float
- ✓ converts value as Fixed
- ✓ converts value as Number
- ✓ converts value as Date
- ✓ can use the a modifier if you like
- ✓ converts value as JSON
- ✓ converts string as Object
- ✓ can use the an modifier if you'd like
- ✓ converts value as Object
- ✓ converts an input element into Values
- ✓ converts a form element into Values
- ✓ converts a query selector into Values
- ✓ converts radio buttons into a Value correctly
- ✓ converts checkboxes into a Value correctly
- ✓ converts multiple selects into a Value correctly

- ✓ converts a complete form into Values
- ✓ converts an element into HTML
- ✓ converts a NodeList into HTML
- ✓ converts an array into HTML
- ✓ converts numbers things 'HTML'
- ✓ converts strings into fragments
- ✓ converts elements into fragments
- ✓ converts arrays into fragments
- ✓ can accept custom conversions
- ✓ can accept custom dynamic conversions
- ✓ converts a form element into Values JSON
- ✓ converts a form element into Values Form Data

passes: 838 failures: 0 duration: 8.02s

100%

propertyAccess

- ✓ can access basic properties
- ✓ is null safe
- ✓ of form works
- ✓ of form works w/ complex left side
- ✓ of form works w/ complex right side
- ✓ works properly w/ boolean properties

the relative positional expression

- ✓ next works properly among siblings
- ✓ next works properly among siblings with wrapping
- ✓ relative next works properly among siblings w/ query
- ✓ relative next works properly among siblings w/ class
- ✓ relative next works properly among siblings w/ query & class
- ✓ previous works properly among siblings
- ✓ previous works properly among siblings with wrapping
- ✓ relative previous works properly among siblings w/ query
- ✓ relative previous works properly among siblings w/ class
- ✓ relative previous works properly among siblings w/ query & class
- ✓ properly constrains via the within modifier
- ✓ next works properly with array-like
- ✓ next works properly with array-like and wrap
- ✓ next works properly with array-like no match
- ✓ next works properly with array-like no match and wrap
- ✓ previous works properly with array-like
- ✓ previous works properly with array-like and wrap
- ✓ previous works properly with array-like no match
- ✓ previous works properly with array-like no match and wrap

the positional expression

- ✓ first works
- ✓ last works
- ✓ first works w/ array-like
- ✓ last works w/ array-like
- ✓ first works w/ node
- ✓ last works w/ node
- ✓ is null safe

possessiveExpression

passes: 838 failures: 0 duration: 8.02s

100%

- ✓ can access basic properties
- ✓ is null safe
- ✓ can access my properties
- ✓ my property is null safe
- ✓ can access its properties
- ✓ its property is null safe
- ✓ can access properties on idrefs
- ✓ can access properties on idrefs 2
- ✓ can access properties on classrefs
- ✓ can access properties on classrefs 2
- ✓ can access properties on queryrefs
- ✓ can access properties on queryrefs 2
- ✓ can access basic attribute
- ✓ can access my attribute
- ✓ can access multiple basic attributes
- ✓ can set basic attributes
- ✓ can set multiple basic attributes
- ✓ can access basic style
- ✓ can access my style
- ✓ can access multiple basic styles
- ✓ can set root styles
- ✓ can set basic styles
- ✓ can set multiple basic styles

array index operator

- ✓ can create an array literal
- ✓ can index an array value at the beginning of the array
- ✓ can index an array value in the middle of the array
- ✓ can index an array value at the end of the array
- ✓ can index an array value
- ✓ can index an array value with an expression
- ✓ can get the range of first values in an array
- ✓ can get the range of middle values in an array
- ✓ can get the range of middle values in an array WITHOUT EXTRA SPACES
- ✓ can get the range of middle values in an array using an expression
- ✓ can get the range of last values in an array
- ✓ can get the range of last values in an array WITHOUT EXTRA SPACES
- ✓ errors when index exceeds array length
- ✓ errors when indexed value is not an array

the beep! expression

- ✓ beeps a basic value
- ✓ beeps a null value
- ✓ beeps a formatted string value
- ✓ beeps the result of an ElementCollection
- ✓ can be cancelled
- ✓ can capture information from event

Templating

- ✓ can render

- ✓ escapes html, with opt-out
- ✓ supports repeat
- ✓ supports if

passes: 838 failures: 0 duration: 8.02s



Work Area
