

Test Support

Version 7.7.0.901

Kathryn Gray

July 30, 2020

Contents

1	Using Check Forms	3
2	GUI Interface	7

1 Using Check Forms

```
(require test-engine/racket-tests)    package: htdp-lib
```

This module provides test forms for use in Racket programs, as well as parameters to configure the behavior of test reports.

Each check form may only occur at the top-level; results are collected and reported by the test function. Note that the check forms only register checks to be performed. The checks are actually run by the test function.

```
(check-expect expr expected-expr)
```

Checks whether the value of the *expr* expression is `equal?` to the value produced by the *expected-expr*.

It is an error for *expr* or *expected-expr* to produce a function value or an inexact number.

```
(check-random expr expected-expr)
```

Checks whether the value of the *expr* expression is `equal?` to the value produced by the *expected-expr*.

The form supplies the same random-number generator to both parts. If both parts request `random` numbers from the same interval in the same order, they receive the same random numbers.

Examples:

```
> (check-random (random 10) (random 10))
> (check-random
  (begin (random 100) (random 200))
  (begin (random 100) (random 200)))
> (test)
Both tests passed!
```

If the two parts call `random` for different intervals, they are likely to fail:

Examples:

```
> (check-random
  (begin (random 100) (random 200))
  (begin (random 200) (random 100)))
> (test)
Ran 1 check.
```

```
0 checks passed.  
Actual value differs from the expected value.  
  Actual value: 137  
  Expected value: 68  
  
At line 2 column 0
```

It is an error for *expr* or *expected-expr* to produce a function value or an inexact number.

```
(check-satisfied expr property?)
```

Checks whether the value of the *expr* expression satisfies the *property?* predicate (which must evaluate to a function of one argument).

Examples:

```
> (check-satisfied 1 odd?)  
> (check-satisfied 1 even?)  
> (test)  
Ran 2 checks.  
1 of the 2 checks failed.  
  
Actual value 1 does not satisfy "even?".  
  
At line 3 column 0
```

Changed in version 1.1 of package `htdp-lib`: allow the above examples to run in BSL and BSL+

```
(check-within expr expected-expr delta-expr)  
  
delta-expr : number?
```

Checks whether the value of the `test` expression is structurally equal to the value produced by the *expected* expression; every number in the first expression must be within *delta* of the corresponding number in the second expression.

It is an error for *expr* or *expected* to produce a function value.

```
(check-error expr)  
(check-error expr msg-expr)  
  
msg-expr : string?
```

Checks that evaluating *expr* signals an error, where the error message matches the string (if any).

```
(check-member-of expr expected-expr ...)
```

Checks whether the value of the *expr* expression is `equal?` to any of the values produced by the *expected-exprs*.

It is an error for *expr* or any of the *expected-exprs* to produce a function value or an inexact number.

```
(check-range expr min-expr max-expr)
```

```
expr : number?
```

```
min-expr : number?
```

```
max-expr : number?
```

Checks whether value of *expr* is between the values of *min-expr* and *max-expr* inclusive.

```
(test)
```

Runs all of the tests specified by check forms in the current module and reports the results. When using the gui module, the results are provided in a separate window, otherwise the results are printed to the current output port.

```
(test-format)
```

```
→ (or/c (-> any/c (or/c (is-a?/c snip%) string?))  
    (-> any/c output-port? void?))
```

```
(test-format format) → void?
```

```
format : (or/c (-> any/c (or/c (is-a?/c snip%) string?))  
          (-> any/c output-port? void?))
```

A parameter that stores the formatting function for the values tested by the check forms.

If the parameter is a function of two arguments, then it is supplied the value and an output port to render the value to. Otherwise, if it is a function of one argument, then the resulting string is used to render the value.

The default value accepts two arguments and `prints` the `any/c` argument to the given port.

```
(test-silence) → boolean?
```

```
(test-silence silence?) → void?
```

```
silence? : any/c
```

A parameter that stores a boolean, defaults to `#f`, that can be used to suppress the printed summary from test.

```
(test-execute) → boolean?  
(test-execute execute?) → void?  
  execute? : any/c
```

A parameter that stores a boolean, defaults to #t, that can be used to suppress evaluation of test expressions.

2 GUI Interface

```
(require test-engine/racket-gui)    package: htdp-lib
```

This module requires produces an independent window when displaying test results. It provides the same bindings as `test-engine/racket-tests`.