pd2 hw31 2021-11-05 Ivan Chen
worked with no one; advised by no one

rulesToRacketBy

| capability / Racket can… | sample Racket code that uses the capability | Why would a programmer use this? | form of the language | steps that Racket takes as it implements or provides the capability |
|---|---|---|---|---|
| …bind a value to a symbol | `(define mySymbol`<br>`        (+ 2.14 1.0)`<br>` )` | To store a value to be used later. This prevents having to recalculate the same thing multiple times. | `(define symbol`<br>`        expression`<br>` )` | 0. Check if the symbol mySymbol is in the symbol table<br>1. Since the symbol mySymbol is not in the symbol table<br>2. Prepare to add slot with symbol mySymbol<br>3. Evaluate the expression for the value<br>4. Set the symbol mySymbol to identify as the value and store them in the symbol table |
| …retrieve a value that was previously bound to a symbol | `(display mySymbol)` | To get the value stored with an identified, the symbol, to be used in procedures. | `symbol` | 0. Check if the symbol mySymbol is in the symbol table<br>1. Since the symbol mySymbol is in the symbol table<br>2. Get the value associated with this symbol<br>3. Replace the symbol in the expression with its value |
| …evaluate an expression | `5 (literal)` | To get the value of an expression | `expression` | 0. Reading from left to right, evaluate the procedure. |
| …build a procedure | `(define averageoftwo`<br>`  (lambda (a b)`<br>`    (/ (+ a b) 2)`<br>`   )`<br>` )` | To have a defined function which will run a set of predefined expressions to prevent having to type the same procedure multiple times. | `(define function`<br>`   ( lambda (parameters)`<br>`        expressions`<br>`     )`<br>` )` | 0. When there is an open parenthesis followed by a lambda, start packaging the instructions in between the parenthesis. |

| **capability / Racket can…** | **sample Racket code that uses the capability** | **Why would a programmer use this?** | **form of the language** | **steps that Racket takes as it implements or provides the capability** |
|---|---|---|---|---|
| …invoke a procedure | `(display`<br>`(averageoftwo`<br>`194`<br>`926`<br>`)`<br>`)` | To execute a function without having to retype the entire function out again. | `(display`<br>`(function`<br>`arguments`<br>`)`<br>`)` | 0. First, put the invocation of the function on hold, and evaluate the given arguments in the invocation of the function. ex.) f(2*2, 8/8) = f(4, 1) or f(3, 7) = f(3, 7)<br>1. Replace the parameters on the right hand side of the equation with the corresponding evaluated arguments. ex.) f(x, y) = (x+y)/2 --> f(2, 4) = (2+4)/2<br>2. Evaluate the expression by using the specified operations. ex.) (2+4)/2 = 3<br>3. The value of the expression creates an equality with the invocation of the function. ex.) f(2, 4) = 3<br>4. 4.) Return the value produced from the evaluation of the expression. |
| …produce dots of light in the interactions window, to represent a value in a form that a human can read | `(display “Hello World!!!”)` | To show the user the output of the computer program. | *`(display expression)`* | omitted for this capability, since that implementation is hidden inside how DrRacket works, and outside the scope of this course |