Racket + PLAI + DrRacket

Hakan Dingenc

EECS 321, Spring '19

Part I: Racket + PLAI

- **Variables and Functions**

  definition, application, and **`lambda`**

- **Data**
- **Control Flow**
- **Testing**

Variables

Variable = Identifier (Name) + Value

For example:

```
(define x 10)
```

Function application

`(eq? "racket" "plai")` → `false`

`(= 3 3)` → `true`

`(3)` → Error: "application: not a procedure"

Most likely because of too many pairs of parentheses.

Function definition

```
  (define (square x) (* x x))

(define (length x y)
  (sqrt (square x) (square y)))
```

Anonymous functions

```
(lambda (x) (* x x))
```

```
((lambda (x) (>= x 10)) 5) → false
```

- **Variables and Functions**
- **Data**

  built-in and user-defined

- **Control Flow**
- **Testing**

Built-in Data

- Booleans: `true` and `false`
- Numbers: `1`, `2`, `0.5`, `1/2`, ...
- Strings: `"me"`, `"racket"`, ...

Symbols

Symbols are quoted identifiers.

Examples: `'x`, `'y`, `'foo`, ...

Lists

`empty` or `'()`

`(cons 1 empty)` or `(list 1)` or `'(1)`

`(cons 1 (cons 2 empty))` or `(list 1 2)` or `'(1 2)`

`(first (list 1))` → `1`

`(rest (list 1 2 3))` → `'(2 3)`

More Lists

```
(length '(1 2 3)) → 3

(map (lambda (x) (+ x 1)) '(1 2 3)) →
                '(2 3 4)

(foldl + 0 '(1 2 3)) → 6
```

User-defined data : `define-type`

```
(define (rgb-num? n)
  (and (integer? n) (<= 0 n 255)))
(define (cmyk-num? n)
  (<= 0 n 1))


(define-type color
  [RGB (red rgb-num?)
       (green rgb-num?)
       (blue rgb-num?)]
  [CMYK (cyan cmyk-num?)
        (magenta cmyk-num?)
        (yellow cmyk-num?)
        (black cmyk-num?)])
```

User-defined data : `define-type`

```
(RGB 1 1 1)
```

`(RGB -1 2 2)` → Error

`(RGB 1 2)` → Error

- **Variables and Functions**
- **Data**
- **Control Flow**
  - `if`, `cond`, and `type-case`
- **Testing**

if

```
(define (digit? n)
  (if (and (<= 0 n) (<= n 9)) "yes" "no"))
```

`cond` : `if` on steroids

```
(define (CTEC-meaning score)
  (cond
    [(<= 5 score 6) "Great"]
    [(<= 4 score 5) "Good"]
    [(<= 3 score 4) "Meh"]
    [else "Uhh..."]))
```

type-case

```
(define (print-color my-color)
  (type-case color my-color
    [RGB (red green blue)
         (print "RGB")]
    [CMYK (cyan magenta yellow black)
          (print "CMYK")]))
```

```
(define (color-value-sum my-color)
  (type-case color my-color
    [RGB (r g b) (+ r g b)]
    [CMYK (c m y k) (+ c m y k)]))
```

Field binding names can be arbitrary but it’s a good idea to use the same names as the fields or an abbreviation thereof!

- **Variables and Functions**
- **Data**
- **Control Flow**
- **Testing**

Testing

```
(test (CTEC-meaning 5.4) "Great")
```

Testing

To test errors, use `test/exn` instead.

Part II: DrRacket

Testing

Cmd/Ctrl-L → Show details → Syntactic test suite coverage

Keybindings

- Show/hide interactions: Cmd/Ctrl-E
- Auto-indent the entire source: Cmd/Ctrl-I
- Show/hide definitions: Cmd/Ctrl-D

Parens

- Parens, brackets, and braces are all equivalent, but you have to correctly match them.
- It doesn’t matter which closing character key you press on your keyboard to close matching parens, brackets, or braces.

DrRacket Themes

- Recommendation:
  https://github.com/takikawa/drracket-solarized

Resources

- Course book
- Racket and PLAI docs: https://docs.racket-lang.org/search/index.html
  - Quick search: Press F1 with the cursor on the word to search for.
- The Racket guide: https://docs.racket-lang.org/guide/index.html
- Piazza
- Office hours