

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
Department of Electrical Engineering and Computer Science  
6.001—Structure and Interpretation of Computer Programs  
Spring 2004

**Recitation 19**  
**Interpretation**

```
(define (eval exp)
  (cond
```

1. **self-evaluating** - numbers, booleans, strings  
Test:  
Evaluation rule:

2. **if** - (if test cons alt)  
Test:  
Evaluation rule:

```
(define (eval-if exp)
```

3. **name** - x, foo, +, make-named-object  
Test:  
How to store bindings?

```
(define binding-list  
  
(define (lookup-variable varname)
```

4. **define** - (define x 7)

Test:

Accessors for define:

```
(define (define-variable exp)
```

```
(define (define-value exp)
```

```
(define (eval-define exp)
```

5. **combination** - (+ 3 4)

Test:

Evaluation rule:<sup>1</sup>

```
(define (eval-combination exp)
```

---

<sup>1</sup>You can use `apply` to call the builtin procedure

## Pop-Quiz: Interpretation

Name:

Section: 4 or 6

### 1. Evaluate the following:

```
(dfinissez (nom d'individu d'faire-appeler-objet)
  (laissez ((racine-partie (individu d'faire-racine-objet))
    (lambda (message)
      (message de cas
        ((TYPE) (lambda () (type-prolongez l''racine-partie d'appeler-objet)))
        ((NOM) nom (de lambda ()))
        ((INSTALLER) (lambda () 'INSTALL))
        ((DTRUISEZ) (lambda () 'DTRUIT))
        (autrement (de racine-partie de message d'obtenir-mthode))))))
(dfinition le foo (d'crer-appeler-objet 'foo)
(demandez foo 'NOM)
```

### 2. Announced?

Was this pop-quiz announced in lecture last week?

### 3. Screw

Have you voted in the “Big Screw” competition this week?