A property is **static** when its value can be determined by looking at the text of a program.

A property is **dynamic** when the program must be executed in order to determine its value.

We can use static properties of a program to detect errors and to improve program performance at interpretation or compile time.

.

Racket--

```
<exp> ::= <varref>
        | (lambda (<var>) <exp>)
        | (<exp> <exp>)
```

.

A variable reference **is bound** or **occurs bound** in an expression if it refers to the formal parameter in the expression.

A variable reference **is free** or **occurs free** in an expression if it is not bound.

.

```
int sumOfSquares( int m, int n )
{
    return m*m + n*n;
}


int weightedSum( int m, int n )
{
    return scale*m + scale*n;
}
```

.

Free and bound variables in the Racket--:

```
(lambda (z) x)
(lambda (x) x)

((lambda (x) x) y)
(lambda (y)
  ((lambda (x) x) y) )

(lambda (f)
  (lambda (x)
    (lambda (y)
      (+ (f x) (f y)))))
```

.

Today, we use our recursive techniques
to write a program that
processes programs
in Racket--.

Our task is straightforward:

Does a variable occur bound
in a given piece of code?

.

A variable v **occurs bound** in an expression exp if and only if:

- exp is of the form (lambda (var) body) and either
  - v occurs bound in body, or
  - v occurs free in body and
    v is the same as var.

- exp is of the form (exp1 exp2) and
  v occurs bound in either exp1 or exp2.

By definition, no variable occurs bound in a variable reference.

.

A variable v **occurs free** in an expression exp if and only if:

- exp is a variable reference and
  is the same as v

- exp is of the form (exp1 exp2) and
  v occurs free in either exp1 or exp2.

- exp is of the form (lambda (var) body),
  v is the different from var, and
  v occurs free in body.

.

syntax procedures

	pair	list
type predicate	pair?	list?
access procedures	car	first
	cdr	rest
constructor	cons	list

.