CS61A

实例属性和类属性

实例属性是属于**对象自身**的属性

修改类属性会影响所有没有同名实例属性的对象

修改实例属性只会影响自己

Scheme

Quotation(`)

It prevents the evaluation of an expression, meaning the expression is returned exactly as it is.

代码块

```
'(a b c)
```

The result is:

代码块

```
(a b c)
```

Quasiquote

It allows for partial *evaluation*.

代码块

```
(define b 5)
`(a (+ b 1) c )
```

The result of :

代码块

```
(a 4 c)
```

Macros

Evaluation procedure of a macro call expression

- Evaluate the operate sub-expression, which evaluates to a macro
- Call the macro procedure on the operand expressions without exaluating them first
- Evaluate the expression returned from the macro procedure

代码块

```
(define-macro (twice expr)
    (list `begin expr expr))


(twice (print 2))
> (begin (print 2) (print 2))
> 2
> 2

; by compare
(define x (print 2))
> 2
> x
x
; got nothing

(define x (+ 1 2))
> x
> 3

(define (twice expr) (list `begin expr expr))
> (twice (print 2))
> 2
> (begin None None)
> (twice `(print 2))
> (begin (print 2) (print 2))
```

(f)

调用且不传递参

SQL

GROUP BY

通过某种标准分组之后选出合适的数据

代码块

```sql
SELECT size, AVG(height)
FROM dogs
GROUP BY size;
-- 按size分组后求平均身高
```

HAVING

`WHERE` 是在分组前筛选

`HAVING` 是在分组后筛选

代码块

```sql
SELECT size, AVG(height) AS avg_height
FROM dogs
GROUP BY size
HAVING avg_height > 30;
-- 按size分组后找出平均身高大于50的狗的体型
```

Problem solving

> I woule recommend that you follow the same process, whether you have a template, giving you a hint about the structure of a solution, or whether you're writing the program from scratch, that you don't start writing code right away, or even think about what code might already. But instead, think about the problem you're trying to solve.

How to Design Programs

From Problem Analysis to Data Definitions

> Identify the information that must be represented and how it is represented in the chosen programming language. Formulate data definitions and illustrate them with examples.

将信息用合适的数据结构表示

> Data actually represents the information that we want to manipulate.

制定数据定义并用示例说明

Signature, Purpose Statement, Header

> State what kind of data the desired function consumes and produces. Formulate a concise answer to the question what function computes. Define a stub that lives up to the signature.

编写文档说明内容

将要定义的函数。签名 目的说明 头部。说明函数处理和产生的数据类型。所需函数所消耗和产生的数据。

定义域和值域。简明地回答函数计算什么的问题。

Functional Examples

> Work through examples that illustrate the function's purpose

通过示例来说明函数的目的

Function Template

> Translate the data definitions into an outline of the function.

写下要用的关键函数，不一定要在正确的地方

Function Definition

> Fill in the gaps in the function template. Exploit the purpose statement and the examples.

填补函数模板的空白

Testing

> Articulate the examples as tests and ensure that the function passes all. Doing so discovers mistakes. Tests also supplement examples in that they help others read and understand the definition when the need ariesd-and it util arise for any serise program.

帮助其他人在需要时阅读和理解定义。

你写程序不仅仅是为了执行，更是为了让其他人阅读。

`example-driven` 以示例为驱动