$Version

显示版本号

?Head

显示帮助

%

上一个输出结果

%%

上上个输出结果

Scan

p.112 Power Programming With Mathematica

Scan[Print, {1, 2}] 函数本身没有返回值，函数有副作用。除这两点外和Map 一样

(*利用Scan 的副作用实现计数*)

data =Table[Random[],{100}]; (*一百个包含0～1之间的实数List*)

hint = Table[0,{5}] (* List[0,0,0,0,0] *)

Scan[hint[[Ceiling[# 5]]]++&, data]

(*Ceiling[5 #] 5 * 0~1之间的实数，得到0~5 之间的实数，Ceiling 上取整，得到0 ~ 5 之间的整数*)

(* a++ 先返回a, 然后a = a + 1*)

hint

#

pure function 的第一参

#0

代表纯函数本身

#n

第n 参

#1,#2,#3

传入的第一参，第二参， ...

sameQ :=(Length[#1 \[Intersection] #2] == Length[#1])& (* 用交集来判断集合是否相等? *)

sameQ[{1,2,3}, {1,3,2}]

用交集来判断集合是否相等?

##

SlotSequence

所有传入参数

##&[a,b,c]

Sequence[a,b,c]

Sequence 类似 ___ (*0或多Sequence*)

##2

所有传入参数，略过第2 个之前的参数

&

前面是一个匿名函数

& 的优先级非常低

Function[body]

等价于 body &

body的计算结果就是返回值

Function[{a,b..}, body]

多参函数

[[]]

see ?Part

/@

Map[f, expr]

@@

see ?Apply

@@@

Apply at level 1

f @@@ {{a, b, c}, {d, e}}

{f[a, b, c], f[d, e]}

$

系统定义符号以大写字母或$ 开头。

`

指定精度

5.0`4 ^ 73

Precision

/.

ReplaceAll expr/.rules

applies a rule or list of rules in an attempt to transform each subpart of an expression expr.

1 + x^2 + x^4 /. x^p_ -> f[p]

1 + f[2] + f[4]

/;

Condition patt /; test

is a pattern which matches only if the evaluation of test yields True.

(*Replace all elements which satisfy the condition of being negative:*)

{6, -7, 3, 2, -1, -2} /. x_ /; x < 0 -> w

===, =!=

SameQ, UnsameQ

:->(*仅表示形状*)

ref/character/RuleDelayed

RuleDelayed (:>, :>)

输入： Esc + :> + Esc

..

Repeated

(*pattern*)

重复1 或多

...

(*pattern*)

重复0或多次

_

Blank

表示任意的一个表达式

symble_Head

前面给出名字，后面给出类型

__

BlankSequence

一个或多个表达式

___

BlankNullSequence

0 或多

Longest[p]

is a pattern object that matches the longest sequence consistent with the pattern p.

Shortest[p]

is a pattern object that matches the shortest sequence consistent with the pattern p.

Optional (:)

f[x_, y_: 0] := {x, y}

y 有一个默认值

OptionsPattern

OptionValue

有点类似特定命名空间下的枚举值

<|...|>

Association

Hash表

represents an association between keys and values.

_&

Array[_&,3]

{_,_,_}

howto/MapAFunctionOverAList

前缀形式还好，若想带多个参数，可以用Apply：Apply[f, {x, y}] 等价于f@@{x, y}，即f[x, y]。

howto/MapAFunctionOverAList

For example, this changes a sum into a product:

```
In[305]:= Apply[Times, a + b + c]
Out[305]= a b c
```

`Apply` is useful when you want to turn the elements in a list into function arguments.

Create a list of five ordered pairs `{a, b}`:

```
In[306]:= pairs = RandomInteger[{1, 10}, {5, 2}]
Out[306]= {{8, 7}, {5, 9}, {6, 8}, {7, 1}, {6, 7}}
```

`Mod` finds the remainder when dividing the first number of an ordered pair by the second:

```
In[307]:= Mod[10, 4]
Out[307]= 2
```

To apply `Mod` to all of the pairs, you need to work at level 1 of the list (specified by the `{1}`):

```
In[308]:= Apply[Mod, pairs, {1}]
Out[308]= {1, 5, 6, 0, 6}
```

You can use `@@@` as a shorthand to apply at level 1:

```
In[309]:= Mod @@@ pairs
Out[309]= {1, 5, 6, 0, 6}
```

Table[Plot[Sin[n x], {x, 0, 2 Pi}, ImageSize -> {150, 150}], {n, 1, 6}]

In[105]:=
```
Table[Plot[Sin[n x], {x, 0, 2 Pi}, ImageSize → {150, 150}], {n, 1, 6}]
```

Out[105]=