Mathematica: todo es simbólico

Listas: {1,2,3,a,b,z} ("array" in Python) order matters

ListPlot[{1,1,2,2,3,4,4}] returns 

Range[10] returns {1,2,...,10}.

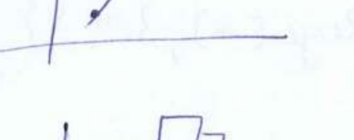
Reverse[{1,2,3}] returns {3,2,1}

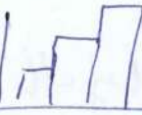
Join[{1,2,3},{1,2,3}] returns {1 2 3 1 2 3}

Range[2,5] returns {2,3,4,5}.

Range[2,10,3] —— {2,5,8}

ListLinePlot[{1,3,5,4}] returns 

BarChart[———] —— 

PieChart[———] —— 

{1,2,3}+10 returns {11,12,13}

{1,2,3}*{1,2,3} —— {1,4,9}

Range[10]^2 —— {1,4,9,...,100}

Sort[{4,2,5,3}] —— {2,3,4,5}

Total [{1,5,9}] returns 15 (= 1+5+9)

Count [{a,b,aabcab}, a] counts # a in {...}.

Part [{7,6,5}, 2] = {7,6,5}[[2]] returns 6.

Min / Max

Take [{9,8,7,6,5}, 3] returns {9,8,7} (list w/ 3 first elmts)

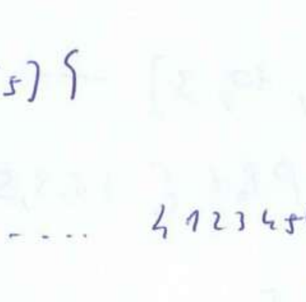
Drop [{ ———, 3] ——— {6,5}.

Table [x, 10] returns $\{\overbrace{x, x, \ldots, x}^{10)}\}$.

Table [a[n], {n,5}] —— {a[1], ..., a[5]}

Table [Range[n], {n, 5}] — {{1}, {1,2}, ... {12345}}

Manipulate [Something with parameter a] returns



% = previous output , %% = ~~dr pre~~ second prev ~~outp~~ output, ...

StringLength ["hello"] returns 5

StringReverse [—] — olleh

Characters ["hello Tomek"] returns {h, e, l, l, o, , T, o, m, e, k}

Table [x, 3] returns {x, x, x}

Table [x, 3, 2] — {{x x}, {x x}, {x x}} (as one would write a 3 x 2 matrix)

||

Table [Table [x, 2], 3] = Table [x, {i, 3}, {j, 2}]

Grid [Table [x, 3, 2]] returns

x	x
x	x
x	x

~~Table(...)~~

~~Table~~ Grid [Table [i*j, {i, 3}, {j, 2}]] returns

1	2
2	4
3	6

Grid [Table [{i, j}, {i, 3}, {j, 2}]] —

{1 1}	{1, 2}
{2, 1}	{2 2}
{3, 1}	{3 3}

RandomInteger [20] returns a integer 1, .. 20.

f[x] = x // f = f@x   f is a function

x // g // f = f@g@x || Map [f, x] = f[#]&@x

f/@ {1, 2, 3} returns {f[1], f[2], f[3]}

f@ {1 2 3} — f[{1 2 3}]

f@@ {1, 2, 3} ⤳ f [1, 2, 3]

Formally, f@@ x replaces the Head of x by f

Range /@ {3, 2, 5} returns {{1 2 3}, {1 2}, {1 2 3 4 5}}

ListLinePlot [{Table [Prime [5000]*n / 5000 , {n, 5000}] ,

Table [Prime [n], {n, 5000}]}]

<u>Recall</u> : Rotate ["one", 90 Degree] returns {one}

Rotate [#, 90 Degree] & /@ {"one", "two"} returns {one two}

Rotate ["hello", #] & /@ {30°, 90°, 180°} returns {hello, hello, hello}

f[#] & /@ {a, b, c} = f /@ {a, b, c}

se puede poner tantas veces como se quiere

f [#, {x, #}, {#, #}] & /@ {a, b, c} // Column returns f[a, {x, a}, {a, a}]
f[b, ...]
f[c, ...]

NestList [f, x, 4] returns {x, f[x], f[f[x]], $f^{(3)}$[x], $f^{(4)}$[x]}

NestList [# + 1 &, 1, 15] returns {1, 2, 3, ... 16}
[2 * # &] — {2, 4, ~~8~~ 16, ... }

2 + 2 == 4 ⇝ True
2 + 2 > 5 ⇝ False

↓ logical expr

If [something, X, Y]   returns X if sth is True
Y if sth is False

If[# <4, x, y] & /@ {1, 2, ... 7} ⇝ {x, x, x, y y y y}

<u>Logical expressions</u> : == , <= , >= , != , && (and) , || (or) , ! (not)

Select [{1, 2, 4, 7, 6, 2} , $\underbrace{\# > 2 \&}_{\text{log expression}}$]   returns {4, 7, 6} ie values that satisfy the log condition

EvenQ[#] & — {2, 4, 6}

PrimeQ[#]& — {2, 7}

(only EvenQ or PrimeQ is enough)

Select [{ — } , EvenQ[#] && # > 2 &]   returns even numbers greater that 2.

MemberQ [{1, 3, 5, 7} , 5] returns True

- We saw how to create a list {f(1), ..., f(5)} with Table [f[n] , {n, 5}] .

This can also be done w/

Array [f, 5]   returns {f[1], ..., f[5]}

Array [#^2&, 10] == Range[10]^2   == Table [n^2 , {n, 10}].

Array[f, {3,4}] returns {f[1,1], f[1,2], ... , f[3,4]}. f has two arguments

Array[#1 * #2 &, {5,5}] // Grid returns

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Array[If [#1 == #2 , x, #1 * #2] &, {5,5}] // Grid ==

== Table[If[i == j , x, i*j], {i,5}, {j,5}] // Grid

Do [Print [i^2] , {i, 4}] returns 1 4 9 16

Do [Print [{i,j}], {i, 4}, {j, i-1}] returns {2,1} {3,1} ⋮ {4,3}

$$\text{Do}[\text{——}, \{i, \underset{\text{min}}{-3}, \underset{\text{max}}{5}, \underset{\text{step}}{2}\}]$$

Do[—— , {i, list}] loops for $i \in$ list [like "for i in list :"]

~~[illegible]~~ n = 1; $\underbrace{\text{While}[n<4, \text{Print}[n]; n\text{++}]}$ return 1 2 3

means:
```
while n < 4 :
    Print(n)
    n++
```

```
While [True,
  n = Input ["enter an integer"];
  If [! IntegerQ[n] || n <= 0 , Break [ ]];
  Print [n, "=", FactorInteger [n]]
]
```

ask for an number

33 = {{3, 1}, {11, 1}}

```
For [i = 0, i < 4, i++ , Print [i]]
```

$$\text{For } [\underbrace{i = 1;\ t = x}_{\text{start}}\ ,\ \underbrace{i\hat{\ }2 < 10}_{\text{test}},\ \underbrace{i{+}{+}}_{\text{incr}},\ \underbrace{t = t\hat{\ }2 + i\ ;\ \text{Print}[t]}_{\text{body}}]$$

Transpose [{{1,2} {3,4}, {5,6}}] returns {{1 3 5}, {2 4 6}}

Flatten [{{1 2} {3 4} {5 6}}] returns {1 2 3 4 5 6} ("aplane" sublistas)

Table [i^j , {i, 3}, {j, 4}] returns ~~{{1 1 1 1} {2 4 8 16~~ }

{{1 1 1 1}, {2, 4, 8, 16}, {3, 9, 27, 81}}

i=1 i=2 i=3

$\underbrace{\text{Table [IntegerDigits [i^j], \{i,3\}, \{j,4\}]}}_{(*)}$ returns {{{1}, {1}, ...}, {{2}, {4}, {8}, {1, 6}}, ...}

(*) // Flatten [#, 1] & returns list with { } removed

Union [{1, 9, 3, 1, 5, 1, 2, 9, 3, 3, 4}] returns {1 2 3 5 9} (different elmts)

Intersection [{1 2 3 4}, {3 4 5 4}] returns {3 4} ($A \cap B$)

Complement [{1 2 3 4}, {3 4 5 4}] returns {1 2} ($A - B$

X = {a b c d e} ; X [[2]] returns b

[[-2]] — d

[[{1,2}]] — {a b}

[[2;;4]] — {b c d}

{ {a b c} {d e f} {g h i} } [[All, 1]] returns {a d g}.

Position [{a b c d b}, b] returns { {2}, {5} }

ReplacePart [{a b c d e}, 3 -> x] returns {a b x d e}

{3 -> x, 5 -> y} — {a b x d y}

3 -> Nothing {a b d e}

~~Pattern~~ • Un patrón es cualquier cosa y se denota con _ :

MatchQ [{a, x, b}, {_, x, _}] returns True

{_, x|y, _} x|y means "any x or y"

Cases [{ {a,b}, {a a a} {a} }, {_, _}] returns { {c, b} }

Cases [{ {a, a}, {b, b}, {c, a, b} }, {__, b}] returns { {b b}, {c a b} }

(eventually ending in b

/. sustitution :

{a, b, a, a, b b} /. b → x   returns {a, x, a, a, x, x}

Cases [{ {a, b}, {a, a}, {abc} }, {x_, x_}] returns { {a, a} }

(pairs that have the two entries the same)

{ f[1], g[2], f[2] } /. f [x_] → x+10   returns { 11, g[2], 12 }

Head ["hello"] es el tipo de "hello", eg String, Head [12] is Integer

Warning! f@g[x, y, z] ⇝ f[g[x y z]]

f /@ g[x y z] ⇝ g[f[x], f[y], f[z]].

g[x_, y_, z_] := x + y + z

~~g[a,b,c]~~ g[a, b, c] == g@@ {a, b, c}.

⚠ g@{x, y, z} ⇝ g[{x, y, z}] !

g @@@ {{1,2,3}, {4,5,6}} $\rightsquigarrow$ {g[1,2,3], g[4,5,6]}

Rule @@@ {{1, 10}, {2, 20}} $\rightsquigarrow$ {1 → 10, 2 → 20}

x = 42 (Then ~~then~~ x == 42)

~~Clear~~
Clear[x] (the value 42 is removed from x)

- Sometimes is useful to ~~d~~ define local variables ~~are~~ different from the rest of the program

$$\text{Module}[\underbrace{\{x = \text{Range}[10], n = 2\}}_{\substack{\text{definition of}\\ \text{variables}}}, x\hat{\ }n]$$

Module[{x,y}, x = Range[10]; y = x^2 ; y = y + 10]

$f[x\_] := x^2$ defines a function. It also works

factorial[1] = 1 ; factorial[n_Integer] := n * factorial[n-1]

Alternatively, one could use: factorial[n_Integer] := If[n==1, 1, n*factorial[n-1]]

Ctrl / $\rightsquigarrow \frac{\square}{\square}$, Crtr 6 $\rightsquigarrow \square^{\square}$ Ctrl _ , $\square_{\square}$

Ctrl ⇧ 8 $\rightsquigarrow$ write math in text Ctrl ⇧ 6 $\rightsquigarrow$ $\overset{\square}{\square}$

f[x_] := Sin[x] + x^2

f'[x] == D[f[x], x] $\leftrightarrow \frac{df}{dx}$

To evaluate, ... /. x -> 3

D[f[x], {x, n}] returns $\frac{d^n f}{d^n x^n}$

D[f, x, y, ...] — $\frac{\partial}{\partial x}\frac{\partial}{\partial y} \cdots f$

D[f, {{x, y, ...}}] — gradient $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \ldots\right)$

D[f, {{x, y, ...}, 2}] — Hessian $\left(\frac{\partial^2 f_i}{\partial x_j^2}\right)$.

Integrate[x^n, x] — $\frac{x^{n+1}}{1+n}$

Integrate[x^2, {x, a, b}] — $-\frac{a^3}{3} + \frac{b^3}{3}$

Derivative[3,5][f][x,y] — $\frac{\partial^8 f}{\partial x^3 \partial y^5}$.

Integrate[If[x^2 + y^2 < 1, 1, 0], {x, -1, 1}, {y, -1, 1}] return $\int_{x^2+y^2<1} 1 = \pi$

DSolve[y'[x] + y[x] == 1, y[x], x] solves $y'(x) + y(x) = 1$

- Write ∂ as $\boxed{esc}$ pd $\boxed{esc}$

D[f[x], x] == ∂_x f[x].

Series [Exp [x], {x, $\overset{\text{at point}}{0}$, $\overset{\text{order}}{10}$}] returns $1 + x + \frac{x^2}{2} + \dots + \frac{x^{10}}{10!} + O[x]^{11}$

Normal [%] returns same without $O[x]^{11}$

This is a SeriesData type (header) variable, InputForm[%] ~> SeriesData[x, 0, {1, 1, 1/2, 1/6, 1/24, 1/120, 0, 6}]

- Series also returns a Laurent series

Total [{a, b, c, d}] returns a + b + c + d

Sum [x^i / i, {i, 1, 7}] returns $\cancel{x} + \frac{x^2}{2} + \dots + \frac{x^7}{7}$

Product [x + i, {i, 1, 7, $\underset{\text{step}}{2}$}] — $(x+1)(x+3)(x+5)(x+7)$

Sum [1 / n^4, {n, 1, Infinity}] — $\frac{\pi^4}{90}$.

Together [$\frac{a}{b} + \frac{c}{d}$] returns $\frac{bc + ad}{bd}$.

Expand [$(x+1)^3$] — $x^3 + 3x^2 + 3x$

ExpandDenominator [(x+1)/(x+2)(x+3)] retuns $\frac{x+1}{x^2 + \dots\dots}$

ExpandNumerator

MapAt[f, {a, b, c, d}, 2] returns {a, f[b], c, d}

{{1}, {4}} — {f[a], b, c, f[d]}

Cases [~~list~~, _Integer, {1}] — {1, 2, 3}

Sqrt[Range[10]] {2} — {2, 3, 5, 6, 7, 2, 10} ← expressions that appear in the second level

∞ — {1, 2, 3, 2, 5, 5, 6, 7, ... } any single number

∪ (union) [esc] un [esc]

(7)

CoefficientRules[(x-y)^3, {x,y}] returns $\underset{x^3}{\{3,0\}} \to 1$, $\underset{x^2y}{\{2,1\}} \to -3$, $\underset{xy^2}{\{1,2\}} \to 3$, $\underset{y^3}{\{0,3\}} \to -1$

lhs = rhs an immediate assignment

lhs := rhs a delayed assignment, ie evaluated each time the value lhs is requested

Eg: f[x_] := Expand[(x+1)^2] , g[x_] = Expand[(x+1)^2]

f[y+2] ⟶ $9+6y+y^2$; g[y+2] ⟶ $1+2(y+2)+(y+2)^2$.

lhs -> rhs an immediate rule

lhs :> rhs a delayed rule , eval.....

Eg: {x, x, x} /. x -> RandomReal[] ⟶ {0.975, 0.975, 0.975}

:> ⟶ {0.243, 0.831, 0.755}

n=1 ; {x, x, a, b, x} /. x :> n++ ⟶ {1, 2, a, b, 3}

Specifying types in Patterns

gamma[n_Integer] := (n-1)!

{a, 4, 5, 6} /. x_Integer -> p[x] ⟶ {a, p[4], p[5], b} x_ only expects 1 argument

x__ expects ≥1 argument

x___ expects ≥0 arguments.

Example: f[x___Integer] := p[x, Plus[x]]

f̄ [x___]. := ____________

g [x__Integer) := ____________

h [x_Integer) := ____________

{f[], f[1), f[1,2,3], f[1,b,3]} ⟶ {p[0], p[1,1], p[1,2,3,6], f[1,b,3]}

{f̄, f, f, f} ⟶ {p[0], p[1,1), p[1,2,3,6), p[1,b,3, 4+b]}

g g g) ⟶ {g[], p[1,1) p[1,2,3,6), g[1,b,3]}

hh h h ⟶ {h[], p[1,1), h[1,2,3], h[1,b,3]}

(highlighted): Not recognized.

//. vs /. : /. apply the rules once, //. apply rules until the expression to its right does not change anymore.

$\{f[f[x]], f[x], f[g[f[x]]]\} \text{ /. } f[x\_] \to x \rightsquigarrow \{f[x], x, g[x], g[f[x]]\}$

$\text{//.} \quad \text{———} \rightsquigarrow \{x, x, g[x], g[x]\}$

rules = {Log[x_ y_] :> Log[x] + Log[y], Log[x_^k_] :> k Log[x]};

$\text{Log[Sqrt[a (b c^d)^e]] /. rules} \rightsquigarrow \frac{1}{2} \log[a(b c^d)^e]$

$\text{//.} \quad \text{—} \rightsquigarrow \frac{1}{2}[\text{Log}[a] + e(\text{Log}\, b + d \log c)]$

= vs == vs ===

=== if lhs & rhs are identically the same (also symbolic)

x === y ⇝ False, x === x ⇝ True

== if lhs & rhs are equal (up to operations or type)

2+2 == 4 (also ===)

0. == 0 → True ; but 0. === 0 → False

= just an assignment

Alternatives[a, b, c] $\rightsquigarrow$ a|b|c

Alternatives @@ {a, b, c} $\rightsquigarrow$ ——.

Collect[(x + a + 1)^4, x] $\rightsquigarrow$ $x^4 + (\dots) x^3 + (\dots) x^2 + \dots$

Join[{a, b, c}, {d, e}, {f, g}] $\rightsquigarrow$ {a, b, ..., g}. (concadenation)

x_. represents a variable that can be omitted with default value Default[f] = 0 (g)

Module[{x, y, ...}, expression] specifies that occurrences of x, y, ... in expression should be treated as local.

* Mostly used for function:

f[x_] := Module[{....}, —; —; —; output]

this is what f[5] will return

Ex 1 : Fibonacci:

fib[n_] := Module[{f}, f[1] = f[2] = 1;
f[i_] := f[i] = f[i-1] + f[i-2];
f[n]]

Ex2: Euclid's algorithm:

```
gcd[m0_, n0_] := Module[{m = m0, n = n0},
    While[n ≠ 0, {m, n} = {n, Mod[m, n]}];
    m].
```

In Module[{x}, —], internally x appears as x$100 or something like that

Timing[expr] ⤳ returns {time required to evaluate express, result of exp}

Replace: {a, b, c, d, a, b} /. a | b -> x ⤳ {x, x, c, d, x, x}

——— /. (v : a | b) -> 2v ⤳ {2a, 2b, c, d 2a, 2b}

a₂ /. (v : a | b)₂ -> v₅ ⤳ a₅

Cases: Cases[{ {1, 4, a, 0}, {b, 3}}, _Integer, 2]

⤳ {1, 4, 0, 3}
returns cases of integers
down to level 2

Cases[{ {1, 4, a, 0, {222}}, {b, 3}}, _Integer, ∞]

⤳ returns cases of integers
down to all levels

$\swarrow$ enkelbror

Thread[f[{a, b, c}]] $\rightsquigarrow$ {f[a], f[b], f[c]}

Thread[{a, b, c} == {x, y, z}] $\rightsquigarrow$ {a == x, b == y, c == z}

Thread[f[{a, b, c}, x]] $\rightsquigarrow$ {f[a, x], ... f[c, x]}

LINEAR ALGEBRA

Table[A, {i, 5}, {j, 3}] $\rightsquigarrow$ {{A[1,1], ... A[1,3]}, ... {A[5,1], .. A[5,3]}}

a matrix 5×3

Array[A, {5,3}] does the same

ConstantArray[0, {5,3}] gives $0 \in \mathcal{M}_{5\times3}(\mathbb{R})$

MatrixForm[%]

DiagonalMatrix[{a,b,c}] gives $\begin{pmatrix} a & & \\ & b & \\ & & c \end{pmatrix}$

IdentityMatrix[3] $\rightsquigarrow$ $\begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}$

A[[1,2]] $\rightsquigarrow$ A[1,2]

A[[2]] $\rightsquigarrow$ the second row

A[[All, 2]] $\rightsquigarrow$ —— column

Dimensions[A] $\rightsquigarrow$ {5,3}

{a,b} + {c,d} $\rightsquigarrow$ {a+c, b+d}

{a+b} + 2 $\rightsquigarrow$ {a+2, b+2}

3 {a,b,c} $\rightsquigarrow$ {3a, 3b, 3c}

{{a,b},{c,d}} . {x,y} $\rightsquigarrow$ {ax+by, cx+dy} (matrix multiplication)