

MATHEMATICA

Actividades de Clase: 40%

Tareas: 30%

Examen: 30%

¿Qué es Wolfram Language?

Wolfram Language es un *lenguaje de computación*. Es una forma de comunicarse con una computadora; en particular, para decirle lo que uno quiere que haga.

Programa

• Aritmética Elemental	Gráficas 2D y 3D
• Datos del mundo Real	Contornos
• Introducción a las funciones	Condicionales
• Un primer vistazo a las listas	Conjuntos de Datos
• Visualización de listas	Importar y Exportar
• Operaciones con listas	Escribir un código
• Colores y estilos	
Cálculo Diferencial e Integral	

ARITMÉTICA ELEMENTAL

2+2	adición
5-2	sustracción
2*3	multiplicación (2 3 también funciona)
6/2	división
3^2	elevación a una potencia (por ejemplo, al cuadrado)

- Calcular $1+2+3$.
- Sumar los números 1, 2, 3, 4, 5.
- Multiplicar los números 1, 2, 3, 4, 5.
- Calcular 5 al cuadrado
- Calcular 3 elevado a la cuarta potencia.
- Calcular 10 elevado a la potencia 12.
- Calcular 3 elevado a la potencia 7×8 .
- Colocar los paréntesis necesarios para que $4-2*3+4$ sea igual a 14.
- Calcular veintinueve mil multiplicado por setenta y tres.
- Sumar los enteros entre -3 y +3.

ARITMÉTICA ELEMENTAL

Vocabulario

Plus[2,2]	2+2	adición
Subtract[5,2]	5-2	sustracción
Times[2,3]	2*3	multiplicación (también puede escribirse como 2 3)
Divide[6,2]	6/2	división
Power[3,2]	3^2	eleva a una potencia
Max[3,4]		máximo (el mayor)
Min[3,4]		mínimo (el menor)
RandomInteger[10]		número entero aleatorio

- Calcule $7+6+5$ usando la función Plus.
- Calcule $2\times(3+4)$ usando Times y Plus.
- Utilice Max para encontrar el máximo entre 6×8 y 5×9 .
- Use RandomInteger para generar un número aleatorio entre 0 y 1000.
- Use Plus y RandomInteger para generar un número entre 10 y 20.
- Calcule $5\times 4\times 3\times 2$ usando Times.
- Calcule $2-3$ usando Subtract.
- Calcule $(8+7)\times(9+2)$ usando Times y Plus.
- Calcule $(26-89)/9$ usando Subtract y Divide.
- Calcule $100-5^2$ usando Subtract y Power.
- Encuentre el mayor entre 3^5 y 5^3 .
- Multiplique 3 por el mayor entre 4^3 y 3^4 .

Vocabulario

UnitConvert [<i>cantidad</i> , <i>unidad</i>]	convierte entre unidades
--	--------------------------

CurrencyConvert [<i>monto</i> , <i>unidad</i>]	convierte entre divisas
---	-------------------------

Convierta 4.5 lbs a kilogramos.

Convierta 60.25 mph a kilómetros por hora.

Convierta 2500 yenes japoneses a dólares US. »

Encuentre el total de 35 onzas, más 1/4 ton (ton = 907 kg), más 45 lbs, más 9 stone (stone = 14 lbs), en kilogramos.

Divisa	Abreviatura	Símbolo
Dólar estadounidense	USD	US\$
Euro	EUR	€
Yen japonés	JPY	¥
Libra esterlina	GBP	£
Dólar australiano	AUD	A\$
Dólar canadiense	CAD	C\$
Franco suizo	CHF	CHF
Renminbi/yuan chino	CNY	元
Dólar de Hong Kong	HKD	HK\$

Dólar de Nueva Zelanda	NZD	NZ\$
Corona sueca	SEK	kr
Won surcoreano	KRW	₩
Dólar de Singapur	SGD	S\$
Corona noruega	NOK	kr
Peso mexicano	MXN	\$
Rupia india	INR	₹
Rublo ruso	RUB	₽
Rand sudafricano	ZAR	R
Lira turca	TRY	₺

LISTAS

Vocabulario

<code>{1,2,3,4}</code>	lista de elementos
<code>ListPlot[{1,2,3,4}]</code>	presenta gráficamente una lista de números
<code>Range[10]</code>	una secuencia de números consecutivos
<code>Reverse[{1,2,3}]</code>	invierte el orden de una lista
<code>Join[{4,5,6},{2,3,2}]</code>	junta varias listas

- Use `Range` para crear la lista `{1, 2, 3, 4}`.
- Construya la lista de los números hasta el 100.
- Use `Range` y `Reverse` para crear `{4, 3, 2, 1}`.
- Construya la lista de los números del 1 al 50 en orden inverso.
- Use `Range`, `Reverse` y `Join` para crear `{1, 2, 3, 4, 4, 3, 2, 1}`.
- Use `Range` y `RandomInteger` para crear una lista de longitud aleatoria hasta 10.
- Encuentre una forma más simple para `Reverse[Reverse[Range[10]]]`.
- Encuentre una forma más simple para `Join[{1, 2}, Join[{3, 4}, {5}]]`.
- Encuentre una forma más simple para `Join[Range[10], Join[Range[10], Range[5]]]`.

LISTAS

Vocabulario

<code>{2,3,4}+{5,6,2}</code>	aritmética con listas
<code>Sort[{5,7,1}]</code>	ordena una lista
<code>Length[{3,3}]</code>	longitud de una lista (número de elementos)
<code>Total[{1,1,2}]</code>	total de los elementos de una lista
<code>Count[{3,2,3},3]</code>	cuenta cuántas veces aparece un elemento
<code>First[{2,3}]</code>	primer elemento de una lista
<code>Last[{6,7,8}]</code>	último elemento de una lista
<code>Part[{3,1,4},2]</code>	un elemento particular de una lista, que también se escribe como {3, 1, 4}[[2]]
<code>Take[{6,4,3,1},2]</code>	toma elementos del principio de la lista
<code>Drop[{6,4,3,1},2]</code>	desecha elementos del principio de una lista
<code>IntegerDigits[1234]</code>	lista de los dígitos de un número

- Cree una lista de los 10 primeros cuadrados en orden inverso.
- Calcule el total de los 10 primeros cuadrados.
- Muestre gráficamente los 10 primeros cuadrados, comenzando por el 1.
- Cree una lista de los 10 primeros múltiplos de 3.
- Use Sort, Join y Range para crear {1, 1, 2, 2, 3, 3, 4, 4}.
- Use Range y + para formar la lista de los números del 10 al 20, inclusive.
- Forme la lista de los 10 primeros cuadrados, usando únicamente Range y Times.
- Encuentre el número de dígitos en 2^{128} .
- Encuentre el primer dígito de 2^{32} .
- Encuentre los 10 primeros dígitos en 2^{100} .
- Encuentre el último dígito de 2^{37} .
- Encuentre el penúltimo dígito de 2^{32} .

`Rest[lista]` da todos los elementos de *lista* excepto el primero. `Most[lista]` da todos los elementos excepto el último.