

Manejo de Software en Ingeniería

Prof.: Dr. Alejandro S. M. Santa Cruz

J.T.P.: Dr. Juan Ignacio Manassaldi

Aux. 2^{ra}: Sr. Alejandro Jesús Ladreyt

- Definir una variable que represente el radio de una circunferencia y darle valor a dicha variable.
- Calcular la longitud de la circunferencia correspondiente, indicando en la línea de comandos que es lo que se calcula y sin que aparezca el resultado.
- Comprobar que el resultado esté en la variable ans.
- Obtener las variables del espacio de trabajo.
- Limpiar el espacio de trabajo y comprobarlo.

```
--> radio=5;
--> radio*2*pi;
--> ans
ans =

    31.415927
```

```
--> who_user
User variables are:
```

```
ans      radio
```

```
Using 0 elements
```

```
ans =
```

```
!ans  !
!      !
!radio !
```

```
--> clear
```

```
--> who_user
ans =
```

```
[]
```

Variable Browser				
	Name	Value	Type	Visibility
	ans	31.4	Double	local
	radio	5	Double	local

- Definir una variable que represente un ángulo de 45° .
- Obtener su expresión en radianes.
- Calcular su coseno.
- Redondear hacia $+\infty$ el resultado del coseno.
- Calcular el logaritmo natural del último resultado.

```
--> angulo_grados=45;
```

```
--> angulo_radianes=angulo_grados*%pi/180  
angulo_radianes =
```

```
0.7853982
```

```
--> c=cos(angulo_radianes)  
c =
```

```
0.7071068
```

```
--> redondeo=ceil(c)  
redondeo =
```

```
1.
```

```
--> l=log(redondeo)  
l =
```

```
0.
```

- Calcular la raíz cuadrada de -1 .
- Definir un número complejo de parte real 3 e imaginaria 5.
- Mostrar su parte real.
- Mostrar su parte imaginaria.
- Calcular su complejo conjugado.
- Calcular la magnitud y ángulo en radianes para poder expresar el número complejo en forma polar.
- Calcular la solución de la ecuación cuadrática: $ax^2+bx+c=0$ donde: $a=1$; $b=4$ y $c=13$.
- Comprobar que las soluciones son correctas.

```
--> sqrt(-1)
```

```
ans =
```

```
i
```

```
--> complejo=3+5*%i
```

```
complejo =
```

```
3. + 5.i
```

```
--> parte_real=real(complejo)
```

```
parte_real =
```

```
3.
```

```
--> parte_imaginaria=imag(complejo)
```

```
parte_imaginaria =
```

```
5.
```

```
--> conjugado=conj(complejo)
```

```
conjugado =
```

```
3. - 5.i
```

```
--> [modulo, angulo] = polar(complejo)
```

```
angulo =
```

```
1.0303768
```

```
modulo =
```

```
5.8309519
```

```
--> a=1; b=4; c=13;
```

```
--> x1=(-b + sqrt(b^2-4*a*c))/(2*a)
```

```
x1 =
```

```
-2. + 3.i
```

```
--> x2=(-b - sqrt(b^2-4*a*c))/(2*a)
```

```
x2 =
```

```
-2. - 3.i
```

```
--> P=poly([c b a], 'x', 'coeff')
```

```
P =
```

```
          2  
13 +4x +x
```

```
--> horner(P, x1)
```

```
ans =
```

```
0.
```

```
--> horner(P, x2)
```

```
ans =
```

```
0.
```

- Calcular la función seno, evaluando la función cada $\pi/10$. Para ello construir el vector x mediante la notación ($:$)
- Acceder a los valores primero y último de la función seno calculado de forma individual y de forma conjunta.
- Calcular el mismo vector x mediante la función `linspace`.

```
--> x=(0:0.1:1)*%pi;
```

```
--> y=sin(x);
```

```
--> y(1)
```

```
ans =
```

```
0.
```

```
--> y(length(y))
```

```
ans =
```

```
1.225D-16
```

```
--> clear
```

```
--> x=linspace(0,%pi,11);
```

```
--> y=sin(x);
```

```
--> y(1)
```

```
ans =
```

```
0.
```

```
--> y(length(y))
```

```
ans =
```

```
1.225D-16
```

```
--> y([1 length(y)])
```

```
ans =
```

```
0.    1.225D-16
```

- Emplear Scilab para resolver el siguiente sistema de ecuaciones con tres incógnitas:

$$\begin{array}{rclclcl} x_1 & + & 2x_2 & + & 3x_3 & = & 366 \\ 4x_1 & + & 5x_2 & + & 6x_3 & = & 804 \\ 7x_1 & + & 8x_2 & & & = & 351 \end{array}$$

Para ello definir las matrices y resolver el problema mediante la división izquierda y mediante la función `inv`, previamente comprobar si la inversa de A es distinta de cero.

```
--> A=[1 2 3; 4 5 6; 7 8 0];
```

```
--> b=[366; 804; 351];
```

```
--> x=A\b
```

```
x =
```

25.

22.

99.

```
--> determinante_de_A=det(A)
```

```
determinante_de_A =
```

27.

```
--> xi=inv(A)*b
```

```
xi =
```

25.

22.

99.

- Crear un vector x de 100 datos entre 0 y 10.
- Calcular y como el seno de x .
- Crear un vector z cuyos valores sean el seno del vector x y tenga fijo a cero los valores negativos del seno de x .

```
--> x=linspace(0,10,100);
```

```
--> y=sin(x);
```

```
--> z=(y>=0).*y;
```

- Crear una función llamada “circulo “ que tenga:
 - Argumentos de entrada: el valor del radio, “r”.
 - Argumentos de salida: el valor de la longitud, “longitud” y el valor de área, “area”.
- Realizar una llamada a la función

```
--> function [long, area]=circulo(r)
> long=2*%pi*r;
> area=%pi*(r^2);
> endfunction

--> [l, a]=circulo(3)
a =

    28.274334

l =

    18.849556
```

- Crear un vector x mediante un bucle for que contenga el seno de $0.1\pi, 0.2\pi, 0.3\pi, \dots, \pi$
- Escribir lo mismo mediante un array o matriz equivalente.
- Escribir lo mismo utilizando un bucle while.

```
--> for i=1:10  
> x(i)=sin(n*%pi/10);  
> end
```

Undefined variable: n

```
--> for i=1:10  
> x(i)=sin(i*%pi/10);  
> end
```

```
--> n=1:10;
```

```
--> xa=sin(n*%pi/10);
```

```
--> m=1;
```

```
--> while m<11  
> xw(m)=sin(m*%pi/10);  
> m=m+1;  
> end
```

- Introducir el valor del número de piezas disponibles. (usar: input)
- Calcular su coste total a 25 cada una.
- Si el número de piezas es mayor que 5 aplica el 20% de descuento, si es menor, el 10%

```
--> piezas=input('Introducir el numero de piezas ')  
Introducir el numero de piezas 6
```

```
piezas =
```

```
6.
```

```
--> costo = piezas*25;
```

```
--> if piezas > 5
```

```
> costo = 0.8*costo;
```

```
> else
```

```
> costo = 0.9*costo;
```

```
> end
```

```
--> costo
```

```
costo =
```

```
120.
```

```
piezas=input('Introducir el numero de piezas ')  
costo = piezas*25;  
if piezas > 5  
    costo = 0.8*costo;  
else  
    costo = 0.9*costo;  
end  
costo
```

archivo.sce