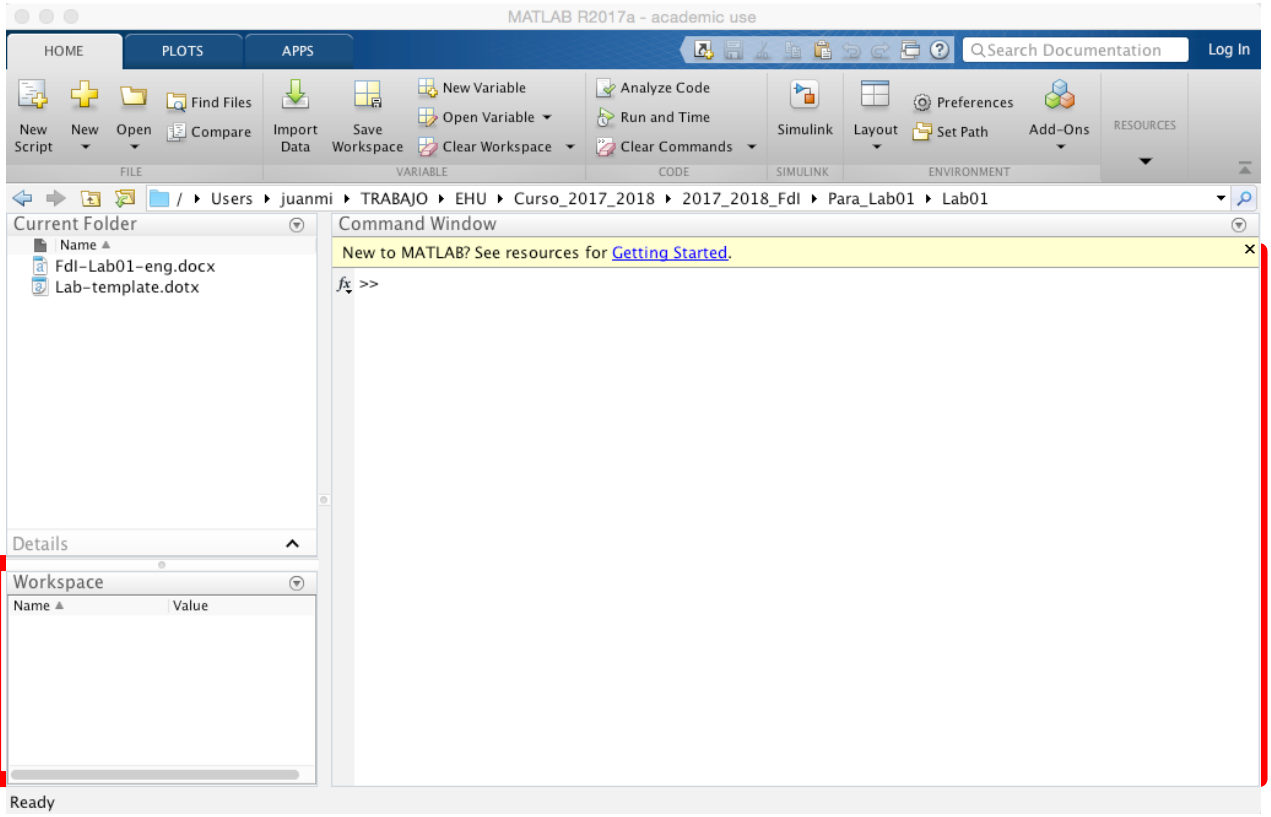
OBJETIVOS DE APRENDIZAJE

- **Entorno Matlab: Espacio de Trabajo, Ventana de Comandos, Editor**
- **Variables, Asignaciones**
- **Operadores Básicos, tipos de datos y de funciones**
- **Scripts**

INTRODUCCIÓN AL ENTORNO MATLAB

Arranca Matlab después de registrarte (nombre y contraseña LDAP). Después del arranque, se mostrará la siguiente interfaz:



Command Window (ventana de comandos) es la ventana principal del entorno de trabajo de Matlab. Se usa para introducir datos (variables), ejecutar funciones y trabajar con scripts de Matlab (más tarde se darán más detalles sobre los m-scripts).

Los comandos se escriben tras el solicitador de comando o *Prompt:* ">>". Por ejemplo, si se quisiera encontrar la respuesta del resultado de la siguiente suma:

```
a+b, where a=10 and b=23
```

escribiríamos los siguiente en la ventana de comandos (Command Window):

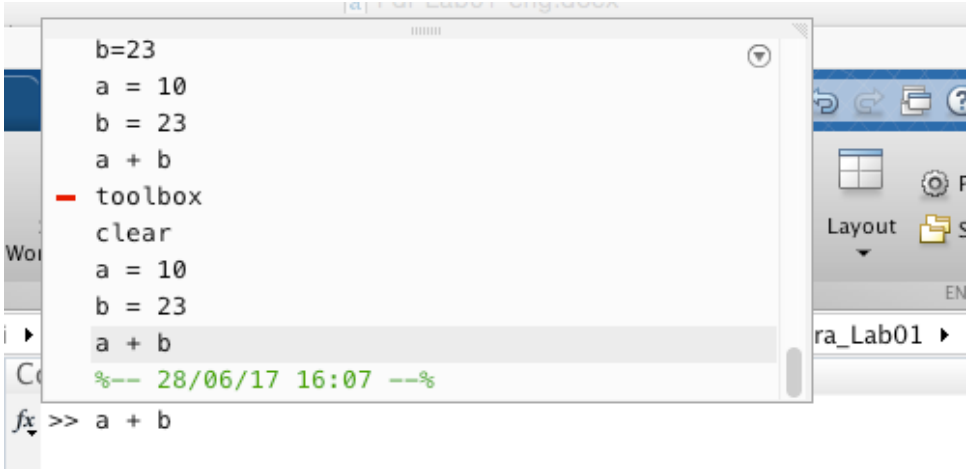
```
>> a = 10
>> b = 23
>> a + b
```

Matlab proporcionará la siguiente respuesta:

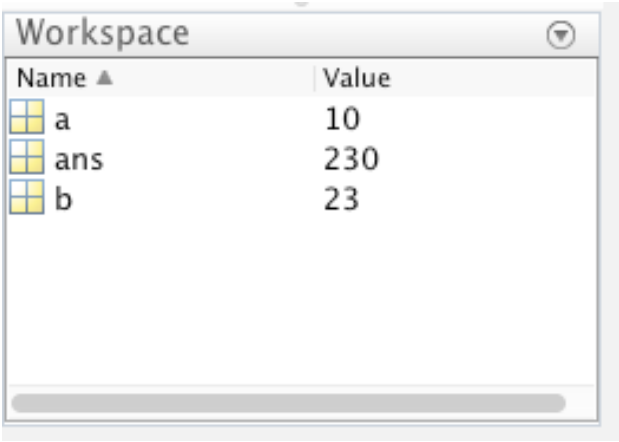
```
ans =
    33
```

Nota: *Hay que pulsar "Enter" para ejecutar un comando*

El historial de comandos (**Command History)** registra todos los comandos escritos en la ventana de comandos. De esta manera nos permite saber qué comandos se han ejecutado previamente y así como volver a ejecutarlos. Para ello se pueden usar las teclas de cursor

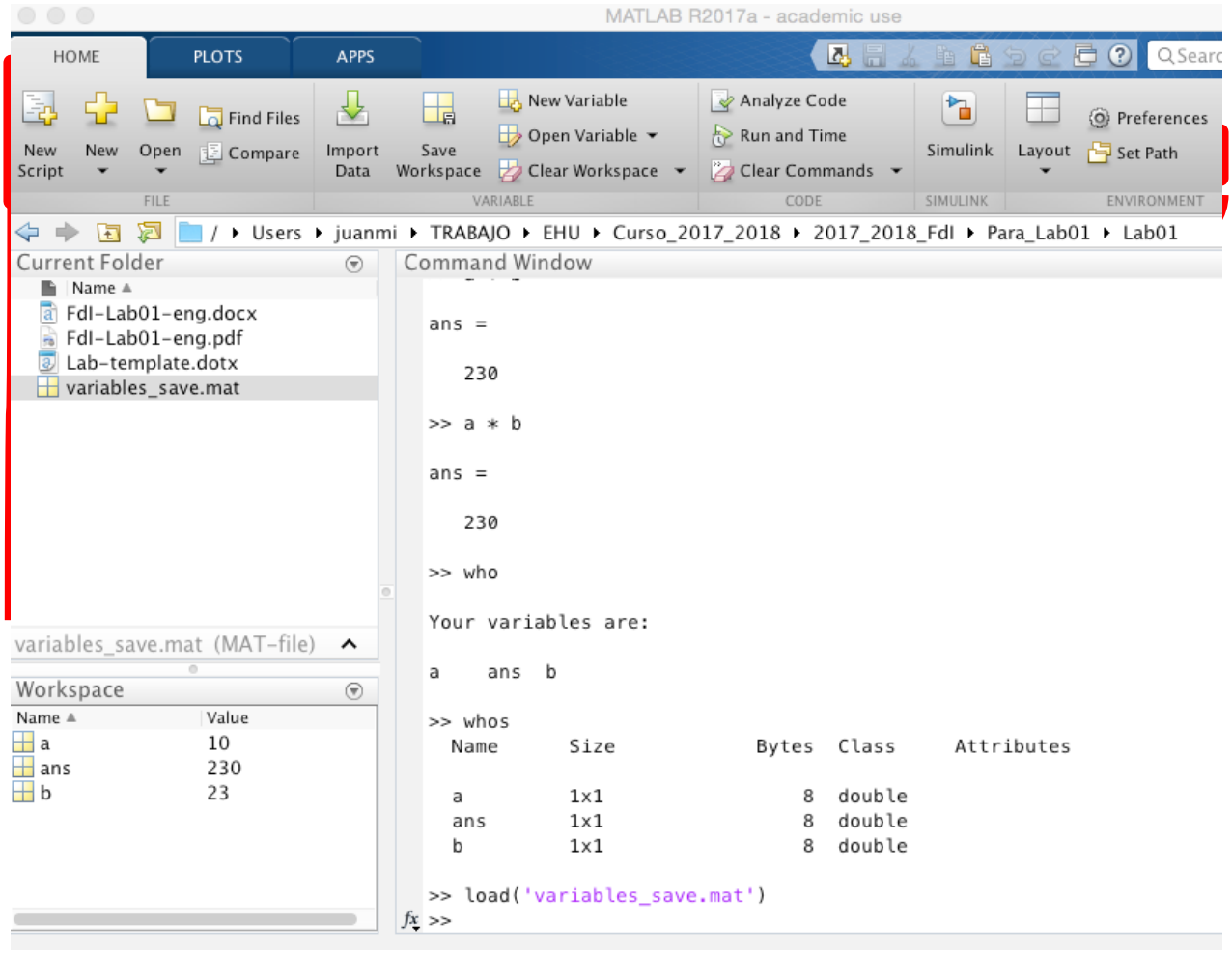


El espacio de trabajo (**Workspace)** muestra todas las variables creadas en la sesión actual de Matlab.



Las variables creadas pueden consultarse usando el comando *who.* Por otra parte, el comando *whos* nos permite ver también los valores y dimensiones de cada una de las variables. Todas las variables en el espacio de trabajo pueden borrarse usando el comando *clear,* mientras que para borrar una variable específica es necesario proporcionar como argumento la variable: *clear a*.

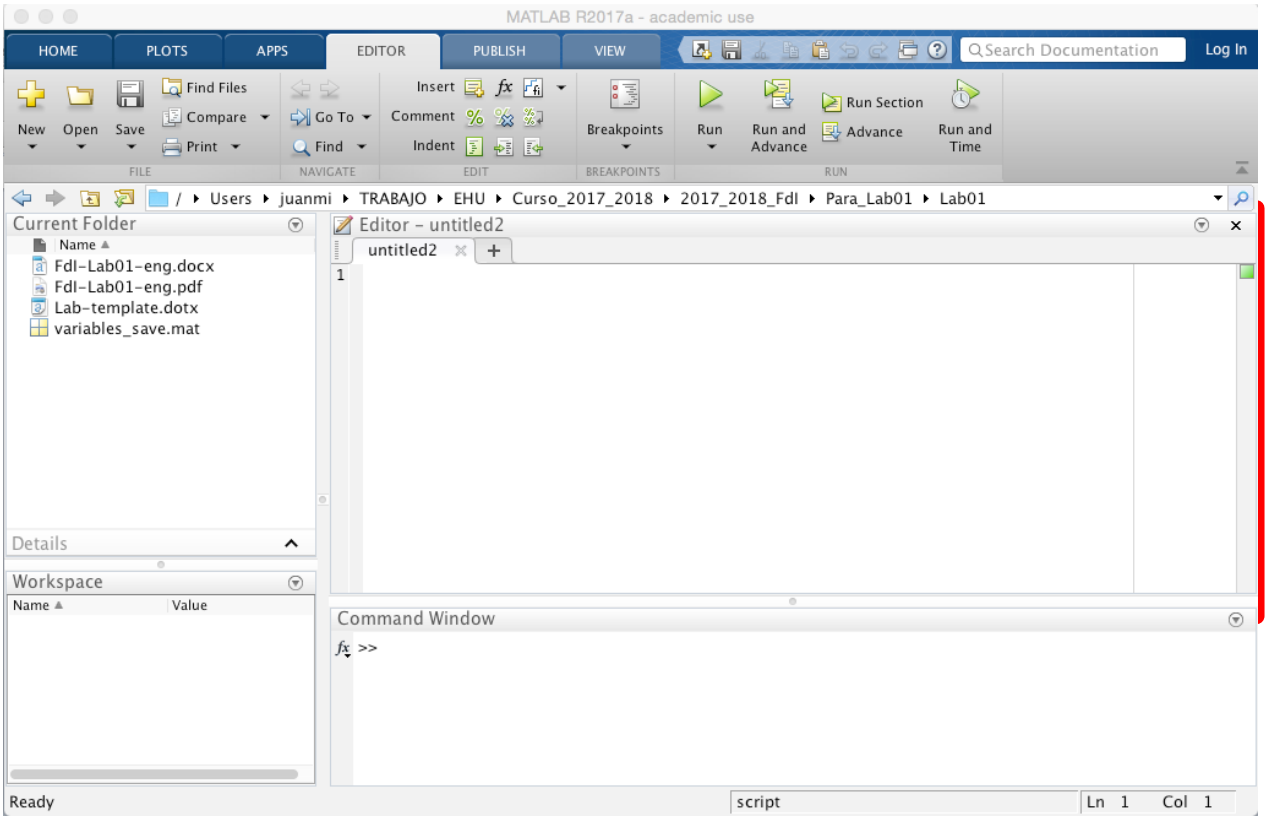
La carpeta actual (**Current Folder)** muestra la lista de todos los ficheros y carpetas disponibles en el directorio actual.



También es posible saber cuál es nuestro directorio actual mediante el comando *pwd* (*print working* directory) en la ventana de comandos.

Para que Matlab tenga acceso a los ficheros con los que se quiere trabajar, es necesario establecer como carpeta de trabajo (*working directory*) la carpeta en la que se quiere realizar el trabajo (usando el comando cd). Alternativamente, es posible añadir la carpeta de trabajo a la búsqueda de ruta (*search* path).

El **editor** se usa para crear programas o *scripts* en m-files. Pulsa el botón “New Script” en la barra de herramientas para que se muestre la venta del editor. La ventana de comandos se mantendrá en el entorno para poder validar la ejecución de los comandos.



VARIABLES Y ASIGNACIONES

Las variables almacenan datos y se definen usando el operador de asignación: “=”.

Dado que el tipo de datos de las variables en Matlab es dinámico, las variables puede ser asignadas sin declarar explícitamente el tipo. Asimismo, su tipo puede ser modificado.

Los valores de una variable puede provenir de constantes, de resultados de operaciones a partir de otras variables y del resultado de una función.

Cuando se le da nombre a una variable, es importante usar convenciones generales para especificar el contenido y su objetivo en el programa. De esta manera, el objetivo de una variable puede ser claramente determinado, ya sea en el momento de programar o cuando se está usando código programado por otra persona.

Aquí puede consultarse una guía de convenciones para nombrar variables:

https://www.ee.columbia.edu/~marios/matlab/MatlabStyle1p5.pdf

```
>>x=200
x=
  200
```

```
>>x='I am a string'
x=
  'I am a string'
>>x = [2*3, 2*pi]
x=
  6.0000 6.2832
>>y = -1*cos(x)
y= -0.9602 -1.000
```

Notas sobre Matlab:

- Matlab distingue entre mayúsculas y minúsculas en los nombres de variables y caracteres. Ello quiere decir que las variables x y X son distintas, al igual que ocurre con los *strings* 'Hello' y 'HELLO'.
- En Matlab todas las variables son matrices o vectores.
- El punto y coma (;) se usa para suprimir la salida a pantalla del output de un comando.

CONSTANTES INTEGRADAS:

MatLab incluye varias constantes integradas. Las más comunes son:

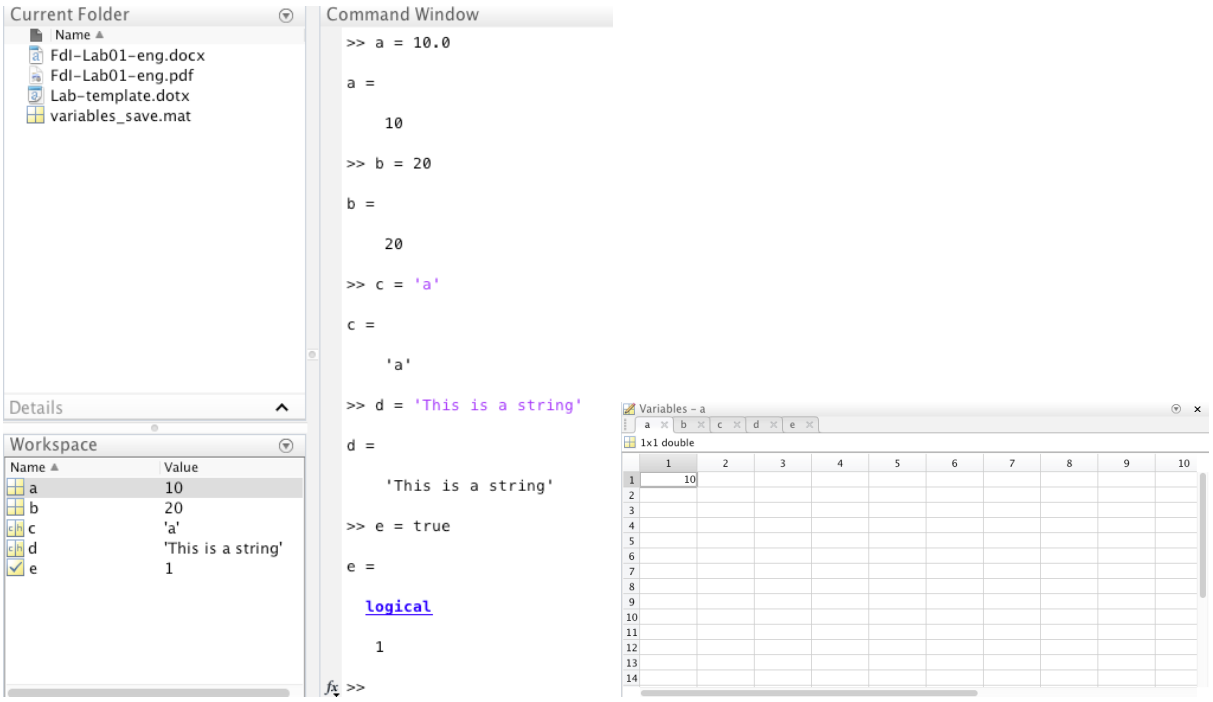
Nombre	**Descripción**
i,j	Usada para números complejos: z=2+4i
pi	¶
inf	∞, Infinito
NaN	No es un número (Not A Number). Por ejemplo, el resultado de una división por 0 es NaN.

OPERADORES BÁSICOS, TIPOS DE DATOS Y FUNCIONES

Hay varios tipos básicos para las variables. Por defecto, Matlab guarda todos los valores númericos como números reales (como flotante de doble precisión). Otros tipos de datos almacenan texto, números enteros, valores lógicos o una combinación de datos relacionados en una única variable. Los tipos de datos más habituales son: ***double, int, char, string, logical***.

- *double*: número real en coma flotante con doble precisión
- *int*: números enteros
- *char*: carácter
- *string*: contenedor para fragmentos de texto
- *logical*: valores booleanos (verdadero o falso)

El siguiente ejemplo muestra las cinco variables, una de cada tipo. Si se pulsa dos veces sobre una variable de la ventana Espacio de Trabajo se abre una pestaña:



Operaciones aritméticas básicas también están incluidas en Matlab:
+,-,*,/,\,res

Además, MatLab incluye una serie de funciones matemáticas habituales que se pueden usar en comandos y programas:

Expresión matemática	**Expresión en Matlab**
Logaritmo Napieriano	log(x)
Logaritmo de Base 10	log10(x)
Raíz cuadrada (square root)	sqrt(x)
Exponencial	exp(x)
Elevado al cuadrado	x^2
Seno (sine)	sin(x)
Coseno	cos(x)
Tangente	tan(x)
Valor absoluto	abs(x)

Note: *Se pueden añadir comentarios en el código usando el símbolo "%". Por ejemplo: %Esto es un comentario*

SCRIPTS

La mayoría de las veces es necesario usar el código que escribimos más de una vez. Para ello, en lugar de escribir todos los comandos necesarios para una tarea

específica, es posible guardarlos en un fichero y ejecutar los comandos que contiene. Al fichero que contiene comandos de programación y que puede ser ejecutado se denomina “script”.

Es posible crear un script con la opción de menú New → Script. En la nueva ventana, se escriben el conjunto de comandos y se guarda. Si guardamos un script con el nombre “example.m”, es posible ejecutarlo desde Matlab mediante el siguiente comando:

```
>>example
```

Nota: para que Matlab encuentre el script al que se está llamando, no se debe incluir la extensión (.m) del fichero. Además, el script debe estar en el directorio actual.

Data Input
Cuando al escribir un script se desconocen los valores a ser usados, es posible preguntar al usuario por los valores que se deben proporcionar como entrada para la ejecución del script. De esta manera, no es necesario modificar el script antes de cada ejecución.

Por ejemplo, el siguiente comando requiere como *input* la altura:

```
>> height = input('what is the height');
```

Una vez es ejecutado este comando, la variable *height* almacenará el valor proporcionado por el usuario, el cual puede ser usado más adelante para realizar los cálculos que sean necesarios.

Si la información requerida se refiere a un *string* en lugar de un valor numérico, es necesario añadir un segundo parámetro con el valor ‘s’ (este valor indica que el valor a introducir debe ser un *string*):

```
>> name = input('give me your name','s');
```

Otro ejemplo:

prompt = 'Introduce a positive number: ';
x = input(prompt)

DATA OUTPUT

Para imprimir en pantalla el valor (escalar) de una variable es suficiente escribir el nombre de la variable (sin punto y coma).

```
>> a = 23;
>> a
a=
  23
```

Si se quiere imprimir los valores de una matriz, también se puede usar el comando display:

```
>> m= [3 4; 2 3];
```

```
>> display(m);
>> ans=
>>      3 4
>>      2 3
```

Si se quiere imprimir texto en la pantalla, entonces es necesario usar el comando fprintf:

```
>> fprintf('Hello my friend!')
>> Hello my friend
```

EJERCICIOS

Programa un script para los siguientes problemas:

(a) Calcular el valor absoluto de un número dado como entrada.

(b) Sumar dos números enteros (los dos números se proporcionan como entrada).

(c) Convertir la temperatura de grados Celsius (*tCelsius* como valor de entrada) a grados Fahrenheit:

$$tFahrenheit = \frac{9}{5} tCelsius + 32$$

(d) Calcular el área de una esfera (*radio* como entrada):

$$área = 4 * \pi * radio^2$$

(e) Calcular el volumen de la esfera (*radio* como entrada):

$$volumen = 4/3 * \pi * radio^3$$

(f) Calcular la distancia euclídea entre dos puntos. Las dos coordenadas de cada punto se piden como entrada. Dados dos puntos (x1, y1), (x2, y2), la distancia entre dichos puntos se define como:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

(g) Calcular la siguiente expresión (x, y son datos de entrada):

$$res = 5 * x^3 + \sqrt{x^2 + y^2} + e^{\ln(x)}$$

(h) Calcular la superficie y volumen de un cilindro en función de la altura y del radio del cilindro, que se suministran como entrada. Para resolver este ejercicio, hay que tener en cuenta que:

- el área de un círculo es:

$$area = \pi * radius^2$$

- el perímetro de un círculo es:

$$perimeter = 2 * \pi * radius$$

- el área de un rectángulo es:

$$area = side1_{length} * side2_{length}$$

(i) Convierte un ángulo de radianes a grados sexagesimales y de grados a radianes (donde *d* y *r* son datos de entrada).

Teniendo en cuenta que:

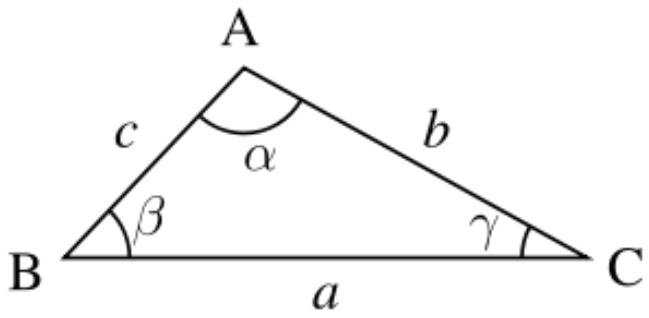
$$2\pi[radianes] = 360[grados]$$

Se debe obtener lo siguiente:

$$d[grados] = r[radianes] * \left(\frac{180}{\pi}\right)$$

$$r[radianes] = d[grados] * \left(\frac{\pi}{180}\right)$$

(j) Dado el siguiente triángulo:



Calcula *c* según la ley de cosenos:

$$c^2 = a^2 + b^2 - 2ab \cdot \cos(\gamma)$$

Para calcular el tercer lado de un triángulo (c), se darán los siguientes datos de entrada: los otros dos lados (a and b) and su ángulo en radianes (γ).

Nota:
El ángulo γ debe expresarse en radianes, ya que la función *cos* de Matlab lo requiere:
https://es.mathworks.com/help/matlab/ref/cos.html

Nota:
Graba todos los comandos en un script. Puedes usar las flechas de cursor para encontrar los comandos si los has borrado de la ventana de comandos.
Puede ejecutarse un script pulsando en el botón *Run*.

ANEXO

A continuación, se describen algunas funciones integradas en Matlab más habituales:

Función	Descripción	Ejemplo
help	MatLab visualiza la información de ayuda disponible.	>>help
help <function>	Visualiza la ayuda sobre una función específica.	>>help plot
who, whos	who lista en orden alfabético todas las variables en el espacio de trabajo actual y que está activo.	>>who >>whos
clear	Elimina las variables y las funciones de la memoria.	>>clear >>clear x
size	Tamaño de arrays, matrices.	>>x=[1 2 ; 3 4]; >>size(A)
length	Longitud de un vector.	>>x=[1:1:10]; >>length(x)
format	Fijar un formato en la visualización de datos.	
disp	Visualiza texto o array.	>>A=[1 2;3 4]; >>disp(A)
plot	Esta función se emplea para obtener gráficos.	>>x=[1:1:10]; >>plot(x) >>y=sin(x); >>plot(x,y)
clc	Limpia la ventana de comandos.	>>clc
rand	Genera un número random, o un vector o matriz con números al azar.	>>rand >>rand(2,1)
max	Encuentra el número mayor en un vector.	>>x=[1:1:10] >>max(x)
min	Encuentra el número menor en un vector.	>>x=[1:1:10] >>min(x)
mean	Media o valor medio.	>>x=[1:1:10] >>mean(x)
std	Desviación típica o estándar.	>>x=[1:1:10] >>std(x)