

# Programa

Un **programa** es una secuencia finita de **instrucciones**.

## **Ejemplo:**

- 1.- Moje el cabello.
- 2.- Coloque shampoo.
- 3.- Masajee suavemente y deje actuar por 2 min.
- 4.- Enjuague.
- 5.- Repita el procedimiento (desde 1.-).

# Programa

## Otro ejemplo:

**Ingredientes:** 15 huevos, 600 gramos de harina, 600 gramos de azúcar

- 1.- Mientras no estén espumosos, batir los huevos junto con el azúcar,
- 2.- agregar la harina en forma envolvente sin batir,
- 3.- batir suavemente,
- 4.- colocar en el horno a 180 grados,
- 5.- si le clavo un cuchillo y sale húmedo, entonces ir a 4.-
- 6.- retirar del horno,
- 7.- mientras no esté frío, esperar
- 8.- desmoldar y servir

# Instrucción

Una **instrucción** es una operación que:

- transforma los datos (el *estado*), o bien
- modifica el flujo de ejecución.

# Instrucción

Una **instrucción** es una operación que:

- transforma los datos (el *estado*), o bien
- modifica el flujo de ejecución.

- 1.- Moje el cabello.
- 2.- Coloque shampoo.
- 3.- Masajee suavemente y deje actuar por 2 min.
- 4.- Enjuague.
- 5.- Repita el procedimiento (desde 1.-)

# Instrucción

Una **instrucción** es una operación que:

- transforma los datos (el *estado*), o bien
- modifica el flujo de ejecución.

- 1.- Mientras no estén espumosos, batir los huevos junto con el azúcar,
- 2.- agregar la harina en forma envolvente sin batir,
- 3.- batir suavemente,
- 4.- colocar en el horno a 180 grados,
- 5.- si le clavo un cuchillo y sale húmedo, entonces ir a 4.-
- 6.- retirar del horno,
- 7.- mientras no esté frío, esperar
- 8.- desmoldar y servir

# Datos

Los programas manipulan **valores** de diferentes **tipos**.

Ejemplos:

- 2.5                      es un valor de tipo **real**.
- "Hola"                  es un valor de tipo **string/char**.
- false                    es un valor de tipo **bool/logical**.
- $f = @(x) x^2+1$       es un valor de tipo **function-handle**.

# Tipos de datos

## Matrices (**double** en Octave)

Un real  $f$  representado en punto flotante es un par  $(m, e)$  tal que:

$$f \approx \pm m * 10^e \quad \text{donde } 0,1 \leq m < 1$$

( $m$ : mantisa;  $e$ : exponente)

El tipo de dato “double” es básicamente una matriz de  $m \times n$ .

$[3 \ 5 \ 7] \rightarrow (3 \ 5 \ 7)$

$[3;5;7] \rightarrow (3 \ 5 \ 7)^t$

$[1 \ 2 \ 3;4 \ 5 \ 6;7 \ 8 \ 9] \rightarrow$

1	2	3
4	5	6
7	8	9

# Tipos de datos

## Valores de verdad (bool, logical):

Hay dos valores de verdad posibles: “verdadero” (*true*) y “falso” (*false*).

## Operaciones de booleanos:

Operador Octave	Operación
! / not	Negación
&& / and	Conjunción
/ or	Disyunción

## Tablas de verdad:

p	!p
true	false
false	true

p	q	p && q	p   q
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false



# Tipos de datos

## Cadena de caracteres (**string**):

Un **character** (**char**) es un símbolo válido en la computadora:

abcdefghijklmnopqrstuvwxyz

ABCDEFGHIJKLMNOPQRSTUVWXYZ

1234567890

!@#\$%\*()-\_+=~`' : ; , . " < > ? /

etc.

En Octave se escriben entre comillas simples: `'a'`.

Un **string** es una cadena o secuencia de caracteres.

# Tipos de datos

## Función inline (**function-handle**):

Una **función inline** (**function-handle**) es una función matemática que podemos evaluar: por ejemplo,  $x^2+1$ .

En Octave se escriben así: `@(x) x^2+1`

`@(s,t) sin(t)*s*pi`

`@(X) X(1)*X(2)+X(length(X))`

**Atención:** Octave no nos deja sumar, restar, multiplicar, dividir o componer funciones.

# Tipos de datos - Resumen

Tipo de datos	Ejemplos
Bool, logical	true, false
float, double	3.0, [10 20; 30 40],- 3.141592
String, char	"pepe", "coco"
Function handle	(x) sin(x)

# Variable

Una **variable** es un **nombre** que denota la **dirección** de una celda en la memoria, en la cual se almacena un **valor**.

En esa celda de memoria es posible:

- **leer** el valor almacenado, y
- **escribir** un valor nuevo, que reemplace al anterior.

# Expresión

Una **expresión** es una combinación de literales, variables y operadores.

La **evaluación** de una expresión arroja como resultado un valor.

Ejemplos:

¿Qué valores resultan de evaluar estas expresiones (suponiendo que s es un string con valor “hola”)?

1

`length(s) + 6`

`(1>0) | !('a'<'b')`

`(5.6 > 2.0) && (length(s) < 2)`

Un **literal** es un valor particular utilizado directamente en el código del programa. En los ejemplos de arriba: 1 6 1 0 'a' 'b' 5.6 2.0 2

# Asignación

***VARIABLE = EXPRESIÓN ;***

Almacena el valor de la *EXPRESIÓN* en la dirección en memoria denotada por *VARIABLE*.

Ejemplos:

<b>x</b> = 1000;	// <b>Bien.</b>
1000 = <b>x</b> ;	// <b>Mal.</b> 1000 no es una variable.
<b>x</b> = <b>y</b> ;	// <b>Bien.</b>
<b>x</b> = <b>x</b> ;	// <b>Bien.</b> Aunque no tiene efecto.
<b>x</b> = <b>x</b> + <b>y</b> * 22;	// <b>Bien.</b>
<b>x</b> + 1 = <b>y</b> ;	// <b>Mal.</b> x + 1 no es una variable.

# Secuencialización

Un **programa** es una secuencia finita de **instrucciones**.

Si *PROG1* y *PROG2* son programas, entonces

***PROG1***

***PROG2***

también es un programa.

Se ejecuta primero *PROG1*. Al terminar, se ejecuta *PROG2*.

**Ejemplo:**

```
a = 10;
```

```
'La variable a tiene valor'
```

```
a
```

# Estado

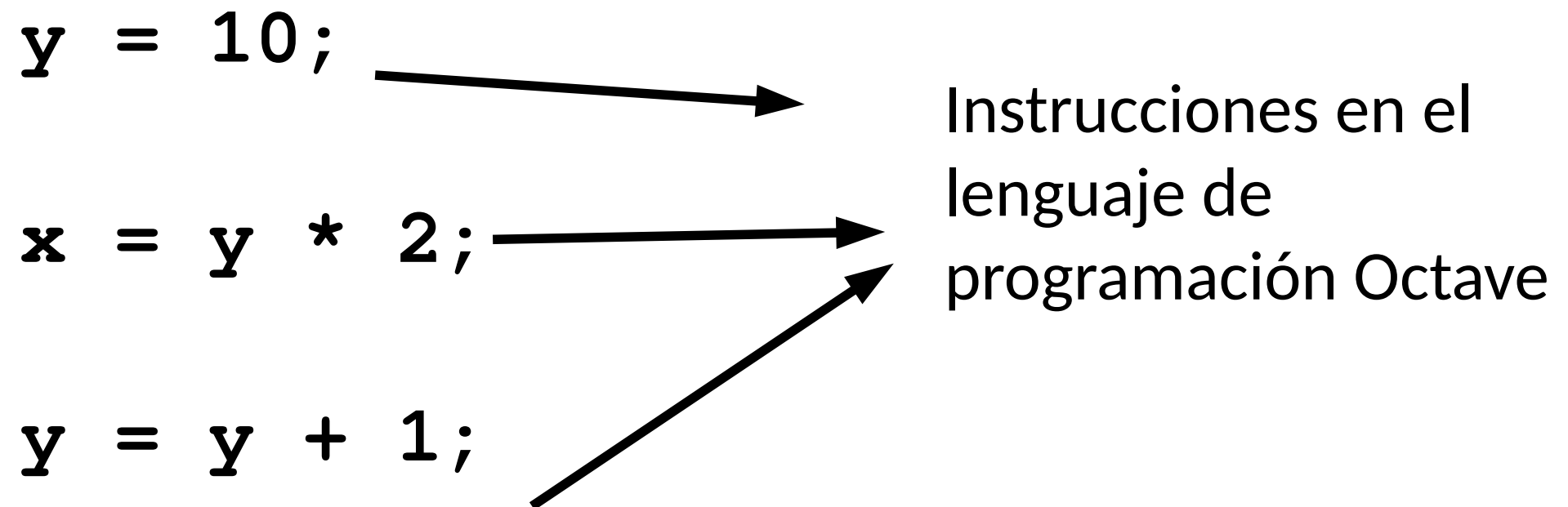
Se denomina **estado** al valor de todas las variables de un programa en un punto de su ejecución.

**Es una “foto” de la memoria  
en un momento determinado.**



# Estado

Ejemplo:



# Estado

Ejemplo:

$\{ x=\uparrow \wedge y=\uparrow \}$

**y** = 10;

$\{ x=\uparrow \wedge y=10 \}$

**x** = **y** \* 2;

$\{ x=20 \wedge y=10 \}$

**y** = **y** + 1;

$\{ x=20 \wedge y=11 \}$

Descripción de los  
estados del programa  
en lógica de predicados

↑ significa “valor indefinido”

# Repaso de la clase de hoy

- Valor. Tipos de datos: bool, string, double.
- Expresiones, variables, literales.
- Memoria, estado.
- Programa, instrucción, asignación, secuencialización.

## Próximos temas

- Condicionales, ciclos, funciones.