

CÓNQUER BLOCKS

BASES DE LA INFORMÁTICA

**CREACIÓN DE ALGORITMOS
(CONTINUACIÓN)**

CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



CUALQUIER NÚMERO SE PUEDE EXPRESAR CON UNAS SERIE DE 0 Y 1



PODEMOS REDUCIR TODO A CÓDIGO BINARIO



SISTEMA BINARIO

32	16	8	4	2	1
1	0	0	1	0	1

$$1 + 4 + 32 = 37$$

CÓDIGO BINARIO

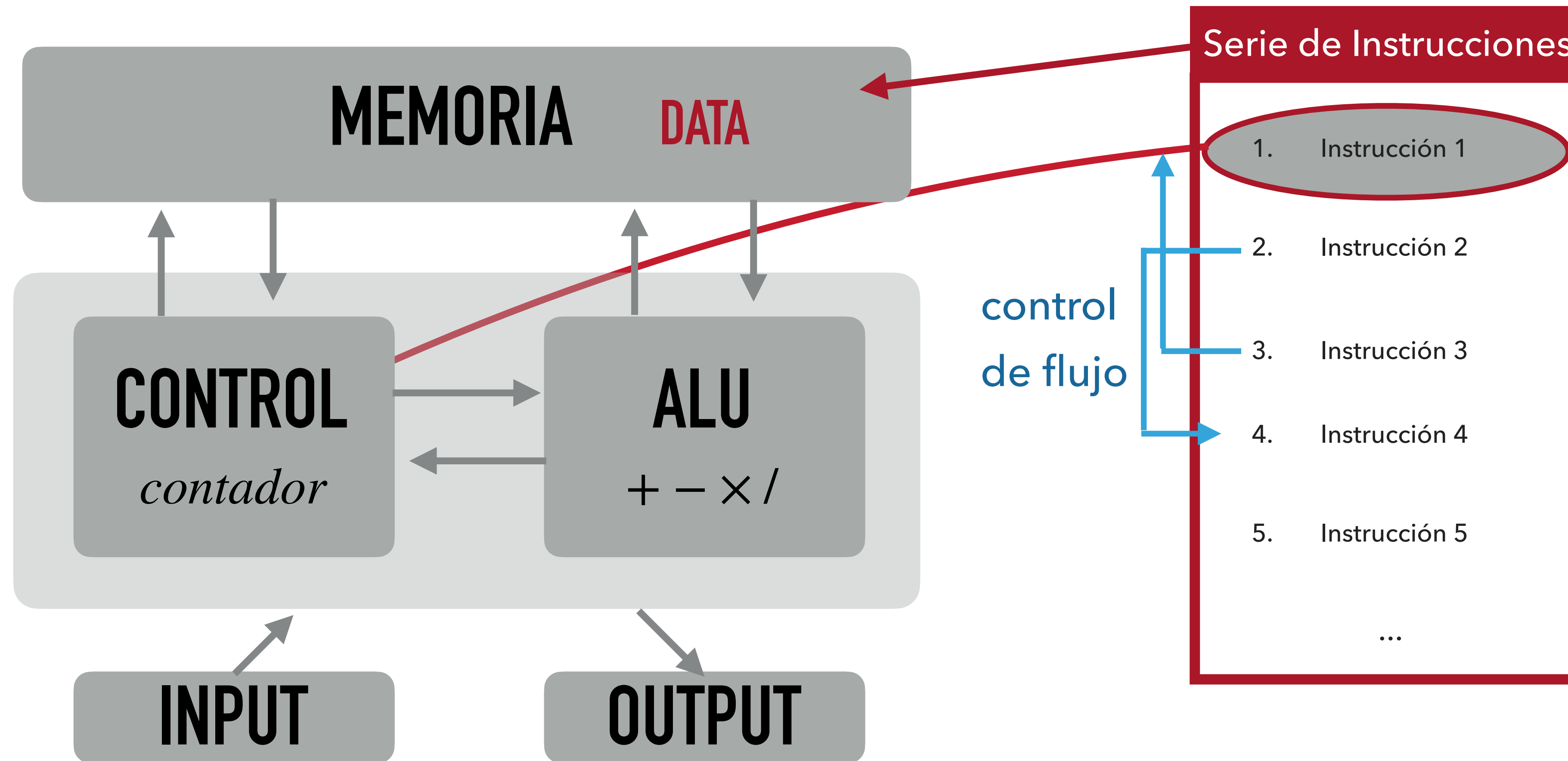
PODEMOS REDUCIR TODO A CÓDIGO BINARIO



Tiempo
Posiciones
Colores
Sonidos
...

**Podemos transmitir todo tipo
de información usando este
método de codificación**

COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: ESTRUCTURA

Datos:

1. Números

2, 7, 12 ...

INTEGER - ENTERO

2.7, 12.6...

FLOAT - DECIMAL

2. Caracteres

"abc"

STRING - TEXTO

3. Lógicos

True / False

BOOLEAN

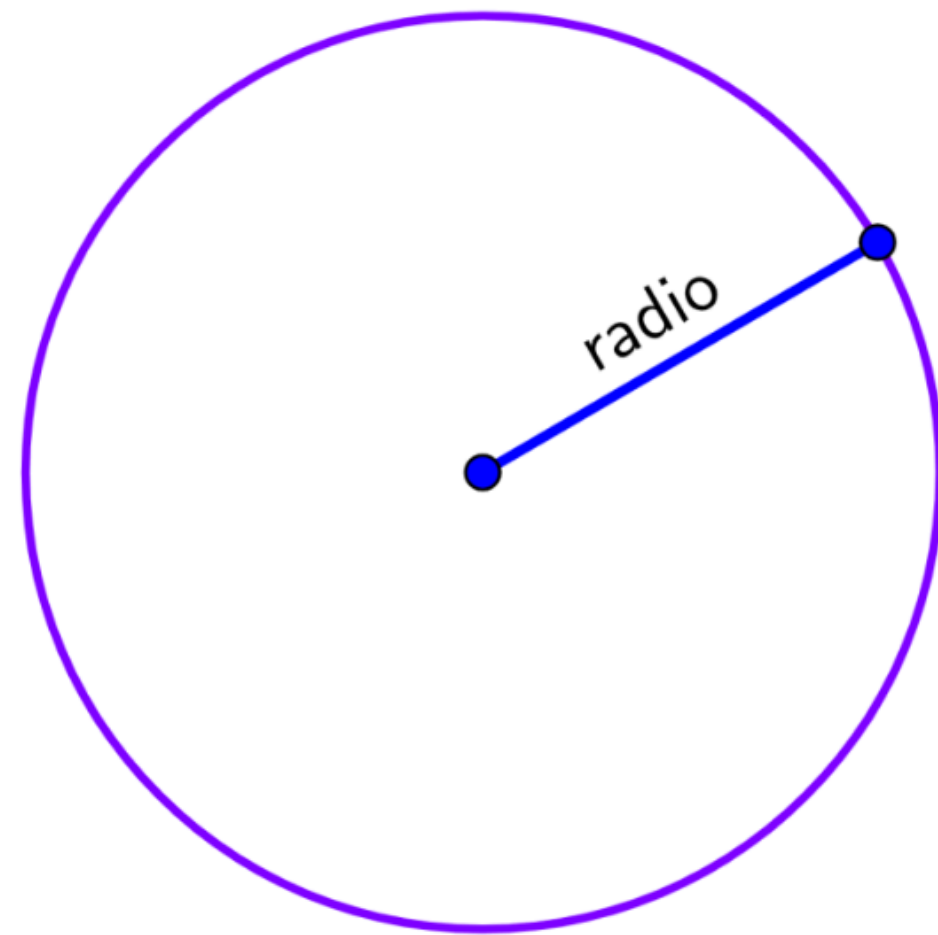
Operaciones primitivas:

- Suma
- Resta
- Multiplicación
- División
- Exponenciación
- Resto (de una división)

Asignaciones:

- Variables
- Constantes
- Expresiones

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141 → Creamos un vínculo entre la palabra Pi y el valor 3.141

radio = 1.5

area del círculo → Comentario

Area = Pi * (radio**2) Area = 3.14 * (1.5**2)

Pi → Constante

r, Area → Variable

Area = Pi * (radio**2) → Expresión

EJEMPLO: AREA DE UN CÍRCULO

```
areaCirculo.py x
areaCirculo.py > ...
1  Pi = 3.141
2  radio = 1.5
3
4  # Area del círculo
5  Area = Pi * (radio**2)
6
7  print(Area)
8

PROBLEMAS  SALIDA  TERMINAL  JUPYTER  COM
/MODULOS/Bases de la informatica/areaCirculo.py"
7.06725
```

Pi = 3.141 → Creamos un vínculo entre la palabra Pi y el valor 3.141

radio = 1.5

area del círculo → Comentario

Area = Pi * (radio**2) Area = 3.14 * (1.5**2)

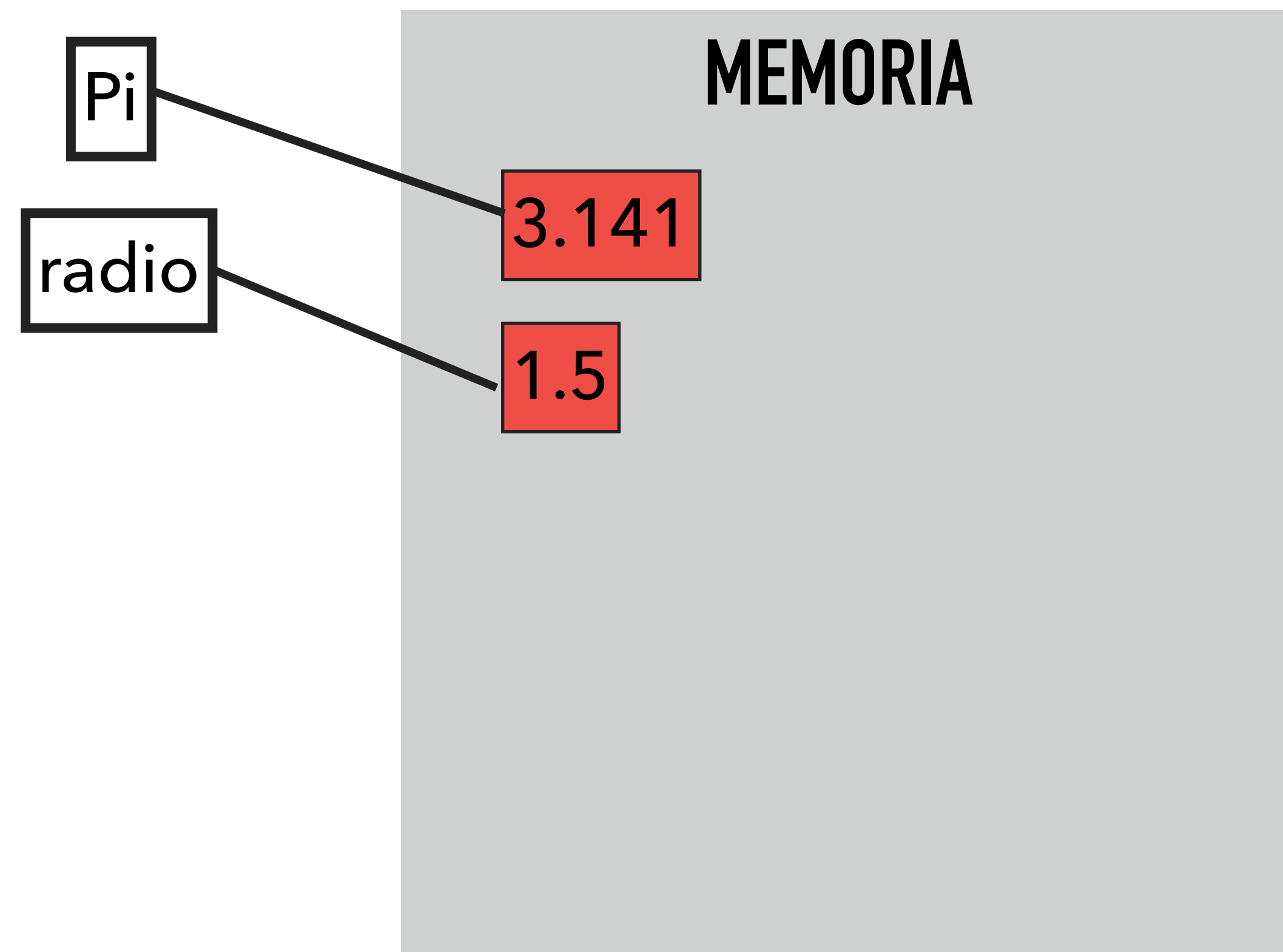
Pi → Constante

r, Area → Variable

Area = Pi * (radio**2) → Expresión

USO DE VARIABLES — EJEMPLO: AREA DE UN CÍRCULO

Las variables se pueden **re-vincular**

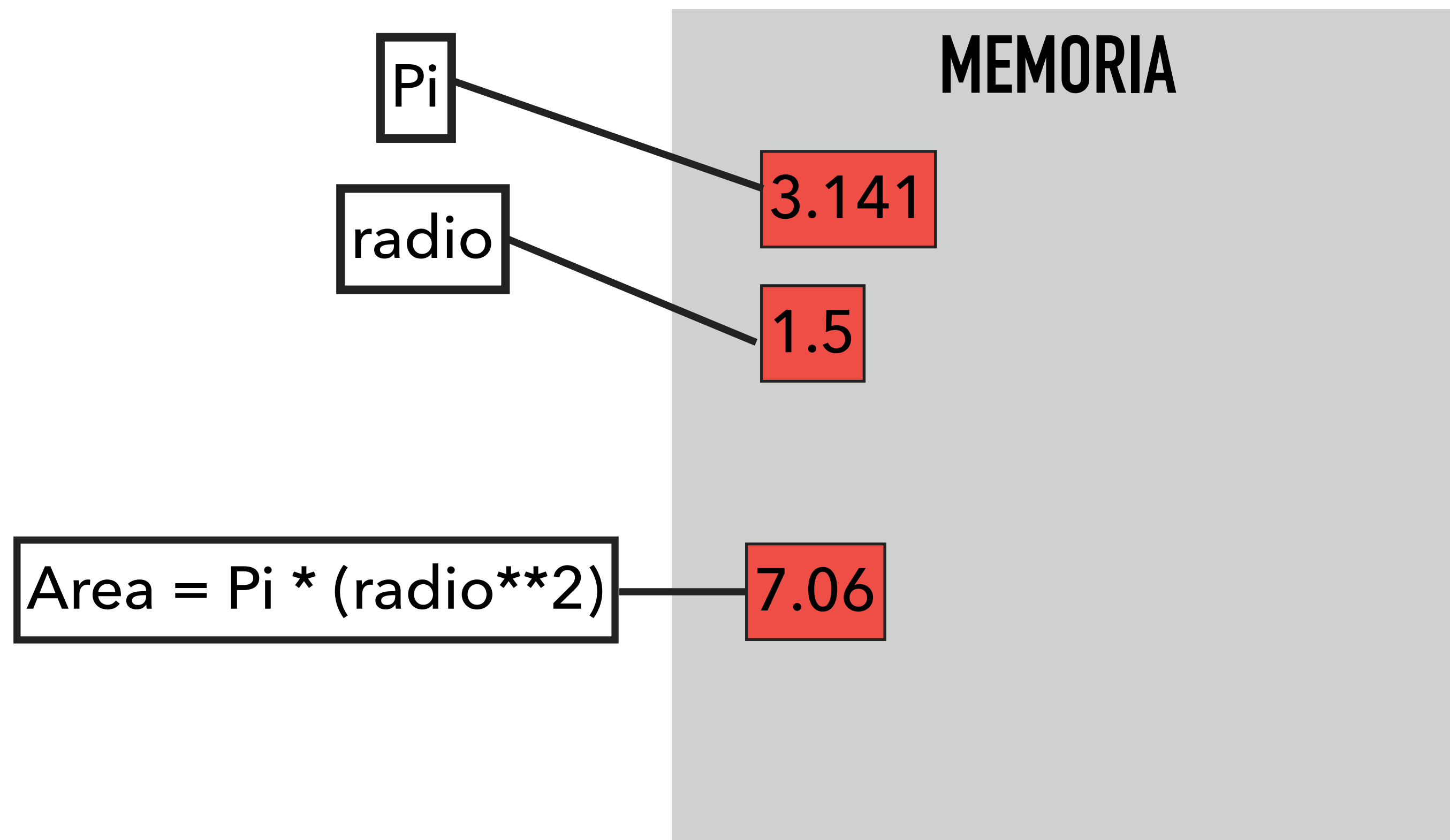


$Pi = 3.141$

$radio = 1.5$

USO DE VARIABLES — EJEMPLO: AREA DE UN CÍRCULO

Las variables se pueden **re-vincular**



`Pi = 3.141`

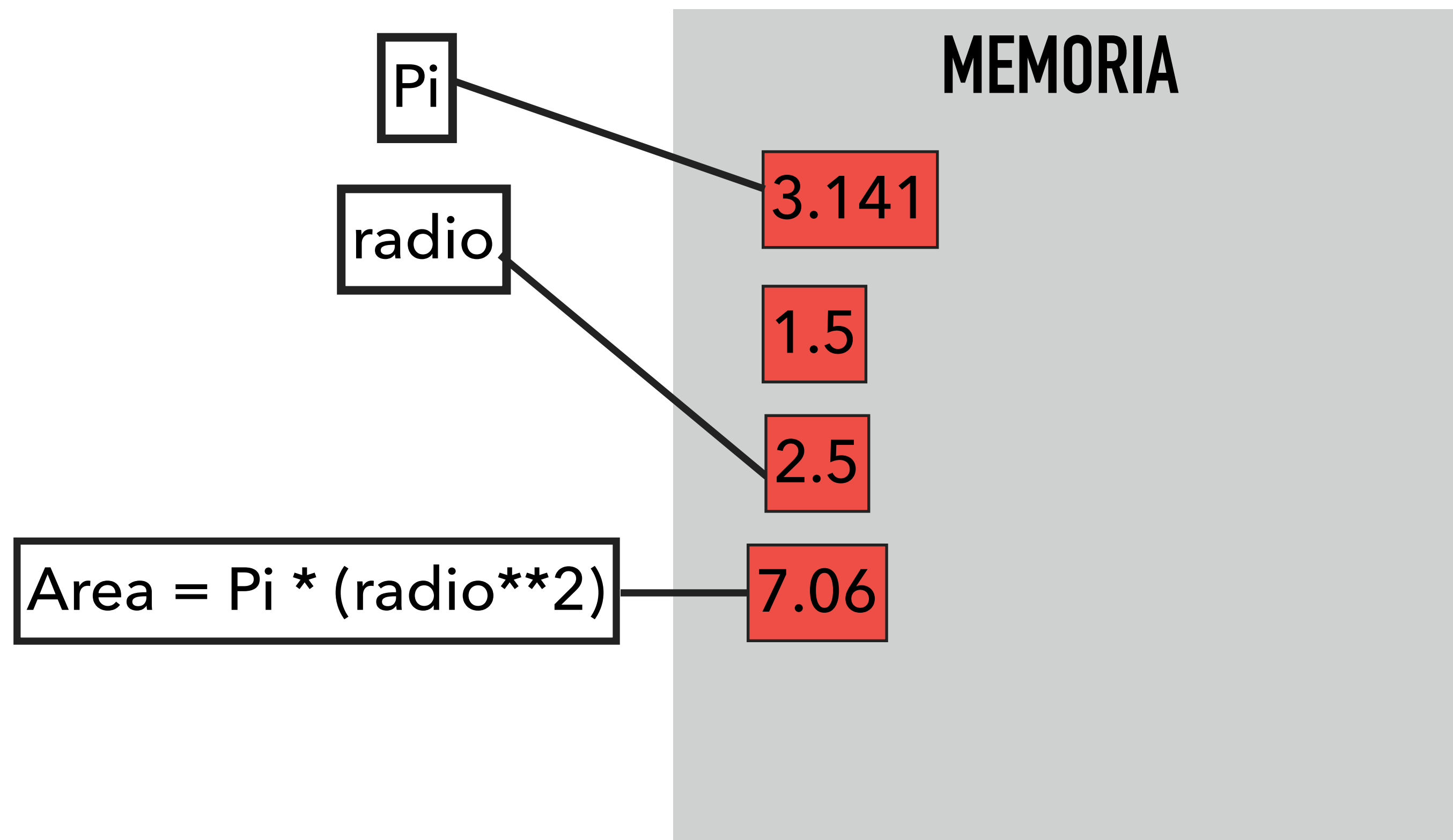
`radio = 1.5`

`# area del círculo`

`Area = Pi * (radio**2)`

USO DE VARIABLES — EJEMPLO: AREA DE UN CÍRCULO

Las variables se pueden **re-vincular**



`Pi = 3.141`

`radio = 1.5`

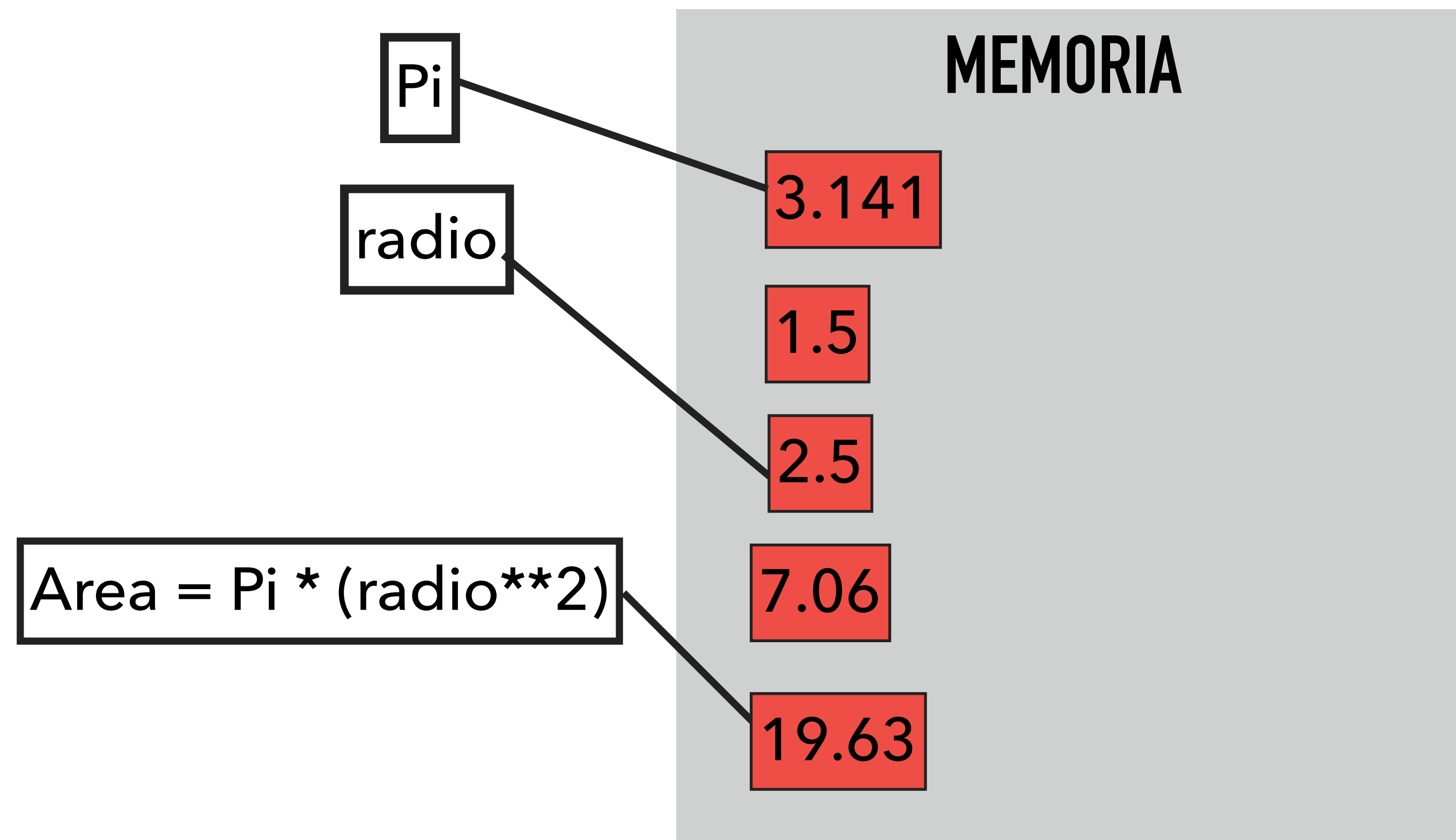
`# area del círculo`

`Area = Pi * (radio**2)`

`radio = 2.5`

USO DE VARIABLES — EJEMPLO: AREA DE UN CÍRCULO

Las variables se pueden **re-vincular**



Pi = 3.141

radio = 1.5

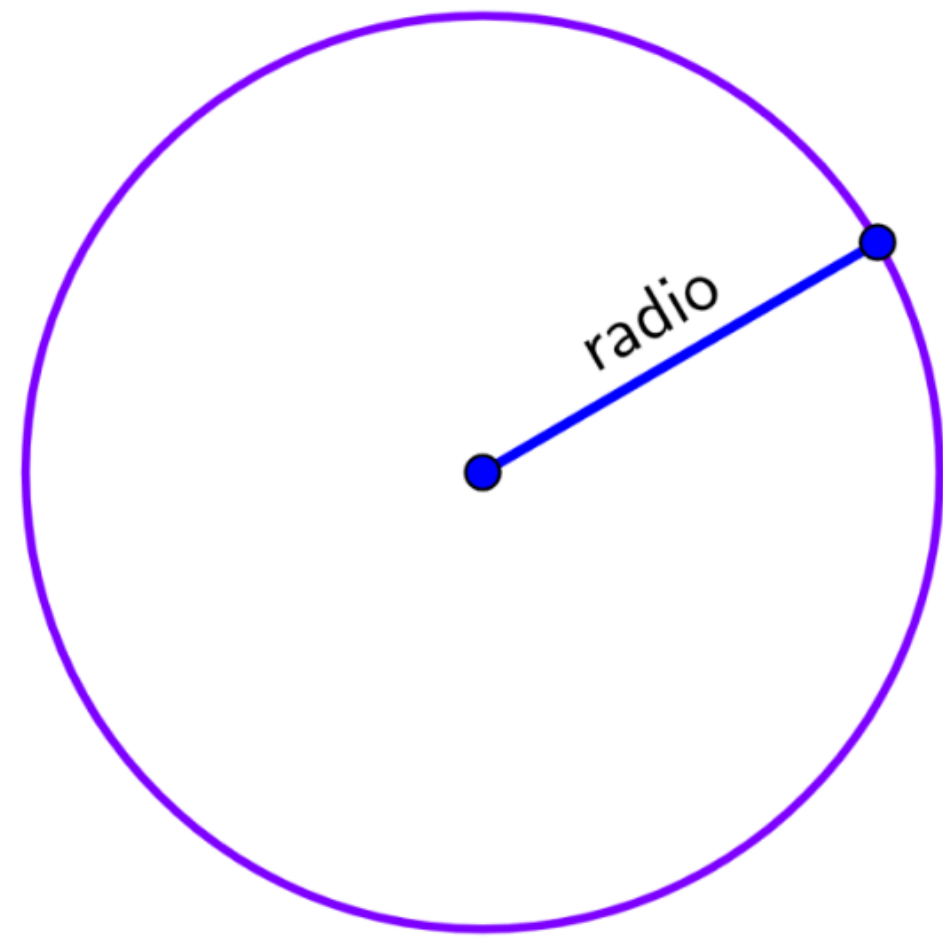
area del círculo

Area = $Pi * (radio^{**2})$

radio = 2.5

Area = $Pi * (radio^{**2})$

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141

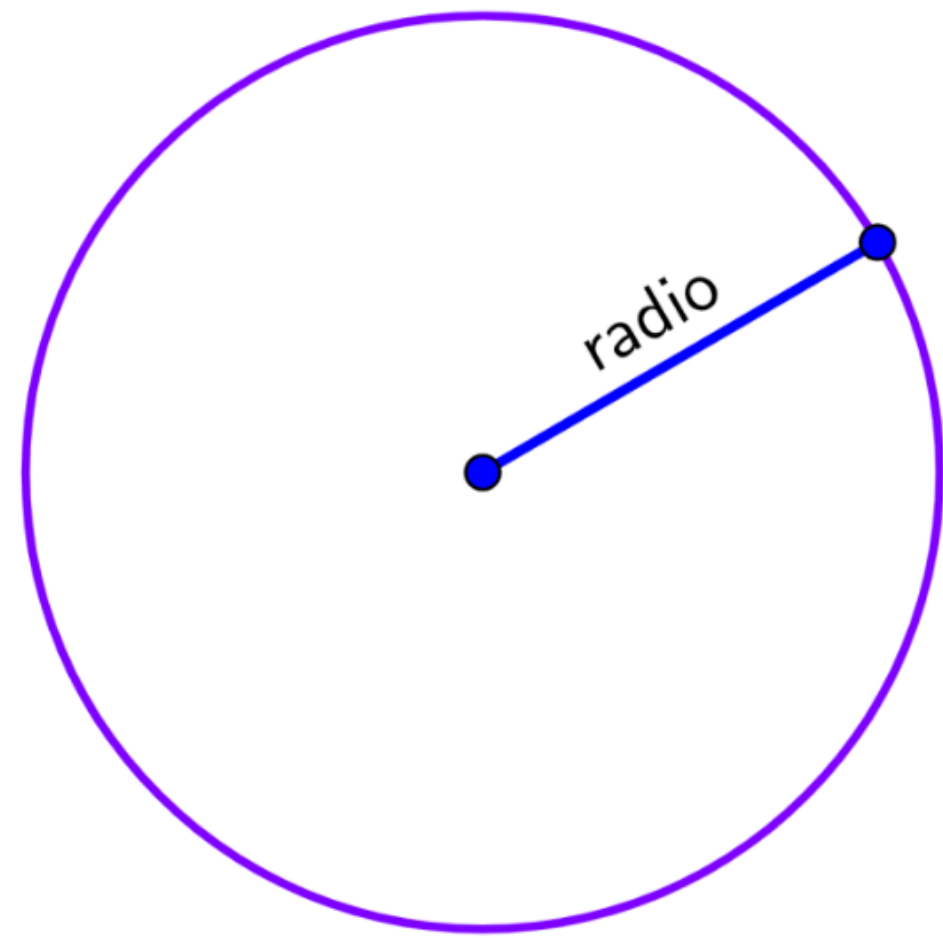
lista de radios: 1.5, 1.1, 3.4 , 8.2, ...

radio = 1.5

area del círculo

Area = Pi * (radio**2)

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141

lista de radios: 1.5, 1.1, 3.4 , 8.2, ...

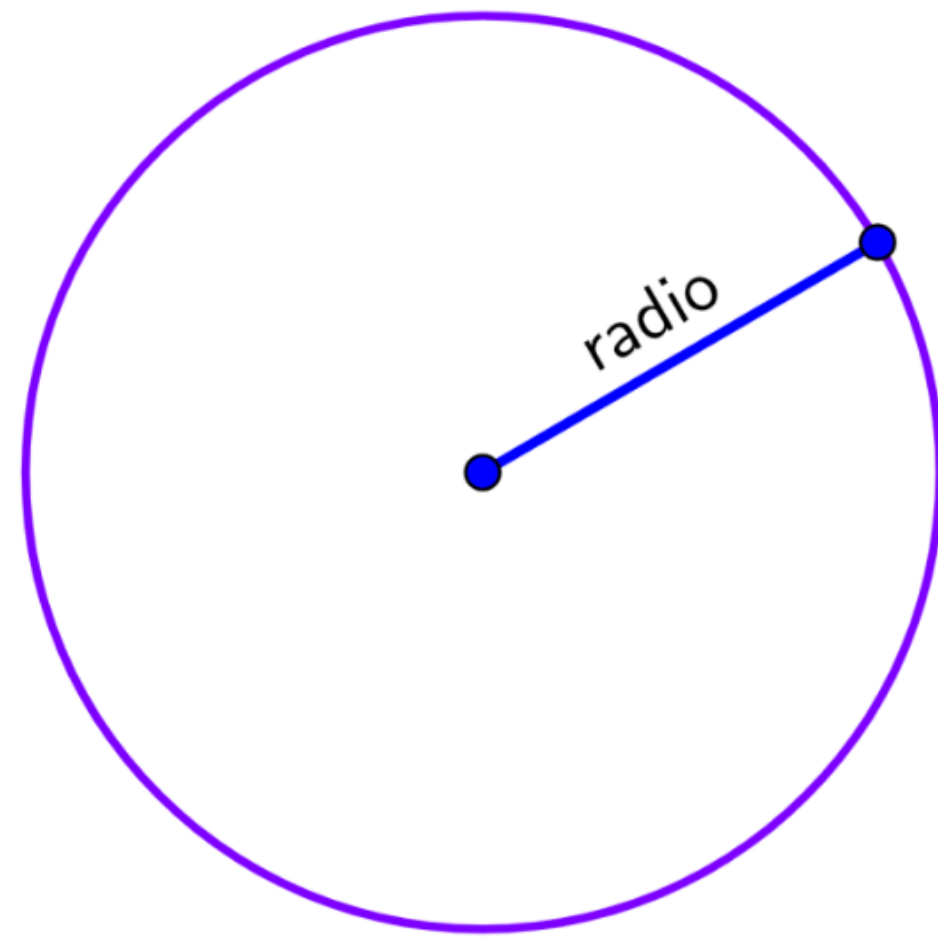
radios = [1.5, 1.1, 3.4 , 8.2] → total de 4 elementos

area del círculo

0:3

Area = Pi * (radio**2)

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141

lista de radios: 1.5, 1.1, 3.4 , 8.2, ...

radios = [1.5, 1.1, 3.4 , 8.2] → total de 4 elementos

area de los círculos

0:3

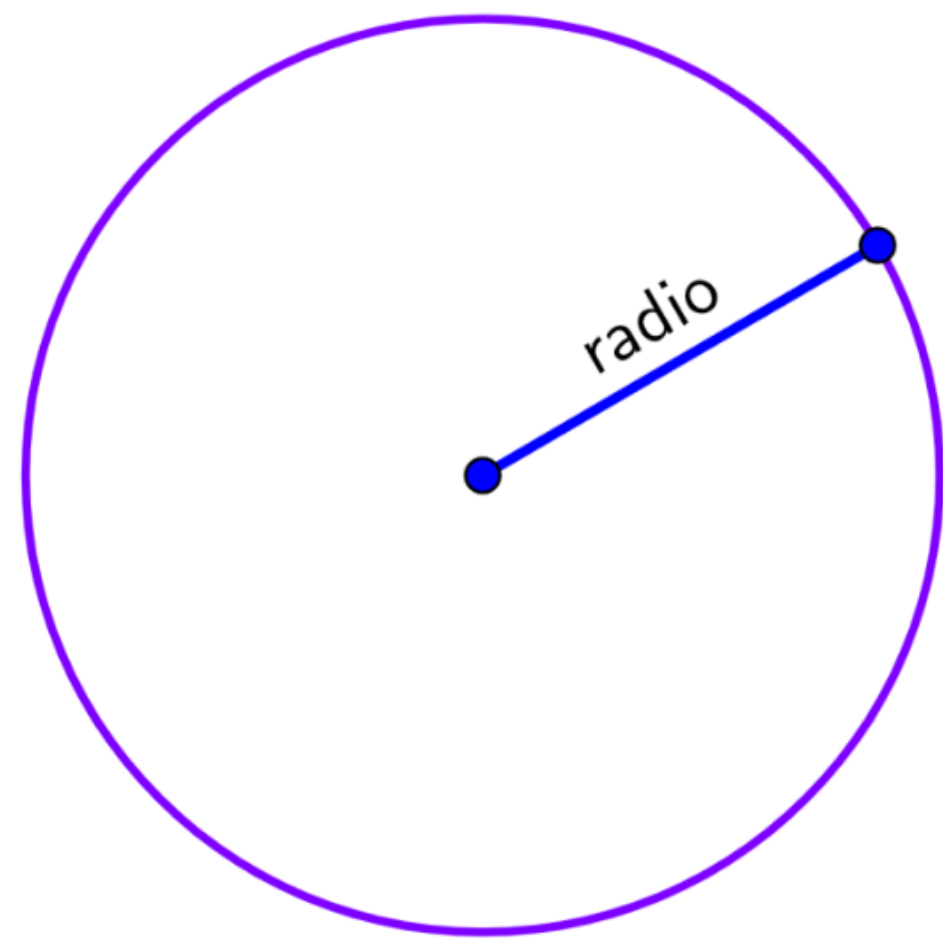
for radio in radios:

Para cada valor de radio en la lista radios

Area = Pi * (radio**2)

...

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141

lista de radios: 1.5, 1.1, 3.4 , 8.2, ...

radios = [1.5, 1.1, 3.4 , 8.2] → total de 4 elementos

area de los círculos

0:3

for radio in radios:

Para cada valor de radio en la lista radios

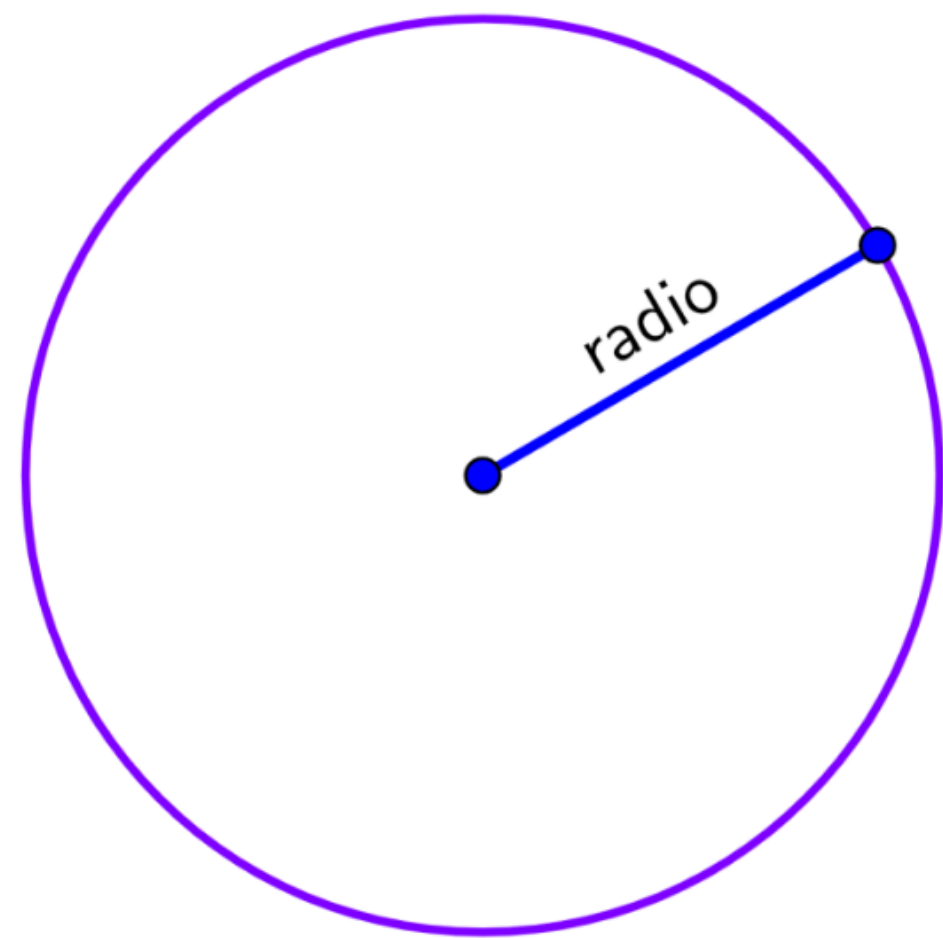
Area = Pi * (radio**2)

...

print(Area)

imprime el valor Area por pantalla

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141

lista de radios: 1.5, 1.1, 3.4 , 8.2, ...

radios = [1.5, 1.1, 3.4 , 8.2] → total de 4 elementos

area de los círculos

0:3

for radio in radios:

Para cada valor de radio en la lista radios

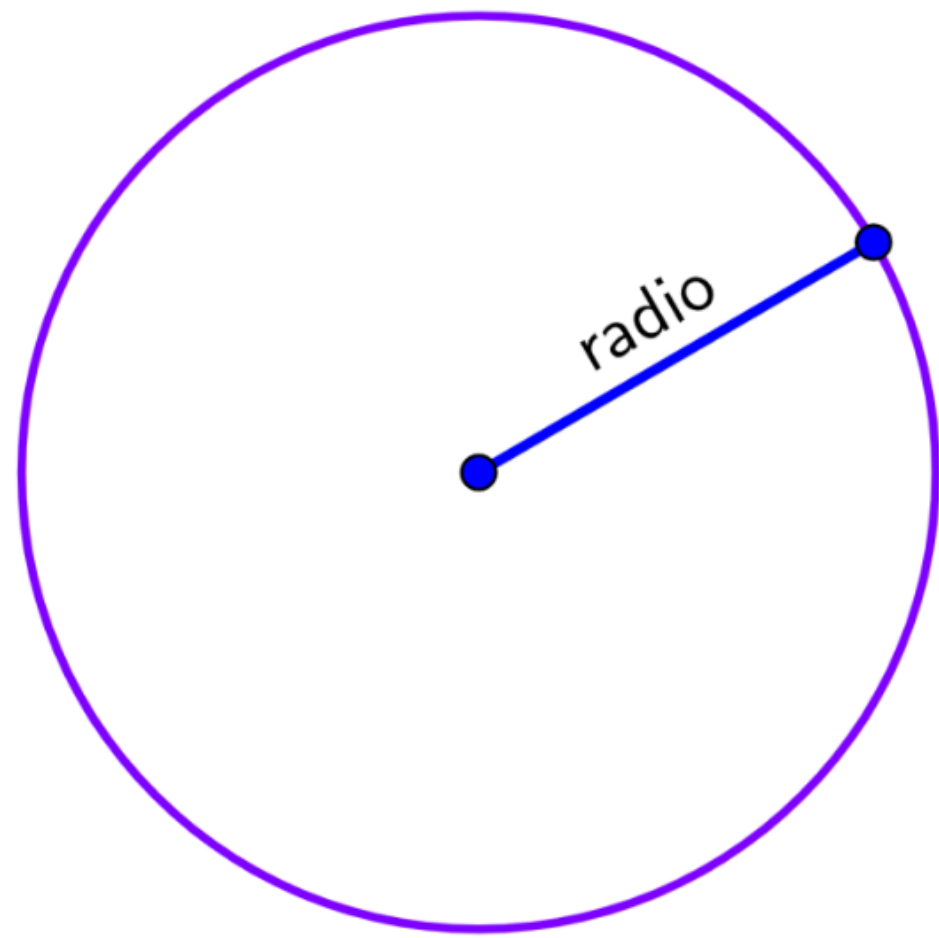
Area = Pi * (radio**2)

...

print(Area)

imprime el valor Area por pantalla

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141

lista de radios: 1.5, 1.1, 3.4 , 8.2, ...

radios = [1.5, 1.1, 3.4 , 8.2] → total de 4 elementos

area de los círculos

0:3

for radio in radios:

Para cada valor de radio en la lista radios

Area = Pi * (radio**2)

...

print(Area)

imprime el valor Area por pantalla

EJEMPLO: AREA DE UN CÍRCULO

```

areaCriculo.py ×
areaCriculo.py > ...
1  Pi = 3.141
2  radios = [1.5, 1.1, 3.4 , 8.2]
3
4  # area de los círculos
5  for radio in radios:
6      Area = Pi * (radio**2)
7
8      print(Area)
9
PROBLEMAS  SALIDA  TERMINAL  JUPYTER

7.06725
3.8006100000000007
36.30996
211.20083999999997
    
```

Pi = 3.141

lista de radios: 1.5, 1.1, 3.4 , 8.2, ...

radios = [1.5, 1.1, 3.4 , 8.2] → total de 4 elementos

area de los círculos

0:3

for radio in radios:

Para cada valor de radio en la lista radios

Area = Pi * (radio**2)

...

print(Area)

imprime el valor Area por pantalla

EJEMPLO: AREA DE UN CÍRCULO

Pi = 3.141

radio = 1.5

area del círculo

Area = Pi * (radio**2)

Pi = 3.141

radios = [1.5, 1.1, 3.4 , 8.2] →

area de los círculos

for radio in radios: →

Area = Pi * (radio**2)

print(Area)

Estructuras de datos

Estructuras de control

Datos

Asignaciones
sencillas

Operaciones
primitivas

EJEMPLO: AREA DE UN CÍRCULO

Pi = 3.141

radio = 1.5

area del círculo

Area = Pi * (radio**2)

Datos

Asignaciones
sencillas

Operaciones
primitivas

Pi = 3.141

radios = [1.5, 1.1, 3.4 , 8.2]

area de los círculos

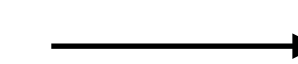
for radio in radios:

Area = Pi * (radio**2)

print(Area)



Estructuras de datos



Estructuras de control

EJEMPLO: AREA DE UN CÍRCULO

Pi = 3.141

radio = 1.5

area del círculo

Area = Pi * (radio**2)

Datos

Asignaciones
sencillas

Operaciones
primitivas

Pi = 3.141

radios = [1.5, 1.1, 3.4 , 8.2]

area de los círculos

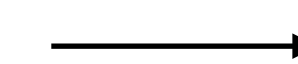
for radio in radios:

Area = Pi * (radio**2)

print(Area)



Estructuras de datos



Estructuras de control

EJEMPLO: AREA DE UN CÍRCULO

Pi = 3.141

radio = 1.5

area del círculo

Area = Pi * (radio**2)

Datos

Asignaciones
sencillas

Operaciones
primitivas

Pi = 3.141

radios = [1.5, 1.1, 3.4 , 8.2]

area de los círculos

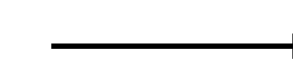
for radio in radios:

Area = Pi * (radio**2)

print(Area)



Estructuras de datos



Estructuras de control

EJEMPLO: AREA DE UN CÍRCULO

Pi = 3.141

radio = 1.5

area del círculo

Area = Pi * (radio**2)

Datos

Asignaciones
sencillas

Operaciones
primitivas

Pi = 3.141

radios = [1.5, 1.1, 3.4 , 8.2]

area de los círculos

for radio in radios:

Area = Pi * (radio**2)

print(Area)

control
de flujo

Serie de Instrucciones

1. Instrucción 1

2. Instrucción 2

3. Instrucción 3

4. Instrucción 4

5. Instrucción 5

...

EJEMPLO: AREA DE UN CÍRCULO

Pi = 3.141

radio = 1.5

area del círculo

Area = Pi * (radio**2)

Datos

Asignaciones
sencillas

Operaciones
primitivas

Pi = 3.141

radios = [1.5, 1.1, 3.4 , 8.2] →

area de los círculos

for radio in radios: →

Area = Pi * (radio**2)

print(Area)

Estructuras de datos

Estructuras de control

Asignaciones complejas

ESTRUCTURAS DE CONTROL

Estructuras condicionales

Comprobamos si algo es cierto o falso y actuamos en función de eso

ESTRUCTURAS DE CONTROL

Estructuras condicionales

Comprobamos si algo es cierto o falso y actuamos en función de eso

```
Si condición entonces  
    set de instrucciones 1  
si no entonces  
    set de instrucciones 2  
fin si
```

ESTRUCTURAS DE CONTROL

Estructuras condicionales

Comprobamos si algo es cierto o falso y actuamos en función de eso

```
Si condición entonces  
    set de instrucciones 1  
si no entonces  
    set de instrucciones 2  
fin si
```

Estructuras iterativas

Comprobamos si algo es cierto y mientras eso se cumple realizamos un set de instrucciones de manera recurrente

ESTRUCTURAS DE CONTROL

Estructuras condicionales

Comprobamos si algo es cierto o falso y actuamos en función de eso

```
Si condición entonces  
    set de instrucciones 1  
si no entonces  
    set de instrucciones 2  
fin si
```

Estructuras iterativas

Comprobamos si algo es cierto y mientras eso se cumple realizamos un set de instrucciones de manera recurrente

```
mientras condición hacer  
    set de instrucciones  
fin mientras
```

```
para  $i \leftarrow x$  hasta  $n$  hacer  
    set de instrucciones  
     $i = i + 1$   
fin mientras
```

ESTRUCTURAS DE CONTROL

Estructuras condicionales

Comprobamos si algo es cierto o falso y actuamos en función de eso

```
Si condición entonces  
    set de instrucciones 1  
si no entonces  
    set de instrucciones 2  
fin si
```

Estructuras iterativas

```
radios = [1.5, 1.1, 3.4 , 8.2]
```

```
# area de los círculos
```

```
for radio in radios:
```

```
    Area = Pi * (radio**2)
```

```
    print(Area)
```

```
mientras condición hacer  
    set de instrucciones  
fin mientras
```

```
para cada i en L hacer  
    set de instrucciones  
fin mientras
```

ESTRUCTURAS DE CONTROL

Estructuras condicionales

- *Encontrar el maximo de dos numeros* -

Pseudocódigo:

```
ESCRIBIR "Introduce el primer número"
LEER x
ESCRIBIR "Introduce el segundo número"
LEER y

SI x > y ENTONCES
    max ← x
SINO
    max ← y
FINSI
ESCRIBIR "Máximo: ", max
```

Si **condición** entonces
 set de instrucciones 1
si no entonces
 set de instrucciones 2
fin si

Python:

```
x = int(input("Escribe un numero"))
y = int(input("Escribe un numero"))

if x > y:
    max = x
else:
    max = y
print("El numero mayor es: ", max)
```

ESTRUCTURAS DE CONTROL

Estructuras condicionales

- Encontrar el maximo de dos numeros -

Pseudocódigo:

```

ESCRIBIR "Introduce el primer número"
LEER x
ESCRIBIR "Introduce el segundo número"
LEER y

SI x>y ENTONCES
    max ← x
SINO
    max ← y
FINSI
ESCRIBIR "Máximo: ", max
    
```

Si *condición* entonces
 set de instrucciones 1
 si no entonces
 set de instrucciones 2
 fin si

Python:

```

maxDosNumeros.py ×
maxDosNumeros.py > ...
1  x = int(input("Escribe un numero "))
2  y = int(input("Escribe un numero "))
3
4  if x > y:
5      max = x
6  else:
7      max = y
8  print("El numero mayor es: ", max)
    
```

PROBLEMAS SALIDA TERMINAL JUPYTER CONSOLA DE

```

/MODULOS/Bases de la informatica/maxDosNumeros.py"
Escribe un numero 35
Escribe un numero 47
El numero mayor es: 47
    
```


CÔNQUER BLOCKS