



INPUT, OUTPUT Y STRINGS

10145 - FUNDAMENTOS DE PROGRAMACIÓN PARA
INGENIERÍA



RESUMEN DE CONTENIDOS



FUNCIÓN `print()`

- `print()` es una **función nativa** de Python que tiene las siguientes características:
 - **Entrada:** Recibe distintos elementos separados por comas
 - **Proceso:** Muestra el texto en la pantalla del intérprete
 - **Salida:** Ninguna, si intento asignar algo a `print()`, Python entregará **None**
- Adicionalmente se pueden invocar los parámetros `sep` y `end`, para seleccionar los caracteres que separan a los elementos de un `print()` o el carácter final de este, respectivamente



FUNCIÓN `input()`

- `input()` es una **función nativa** de Python que tiene las siguientes características:
 - **Entrada:** Recibe un texto, que será mostrado en el intérprete de Python al usuario
 - **Proceso:** Muestra el texto de entrada en pantalla y espera una respuesta que el usuario escribe a través del teclado, hasta que recibe un ENTER
 - **Salida:** Datos ingresados por el usuario como tipo de dato `string`



EJEMPLOS

```
>>> x = input('Ingrese un valor: ')
```

```
Ingrese un valor: 5
```

```
>>> print(x)
```

```
5
```

```
>>> y = print('El valor de x es', x, sep=': ', end='')
```

```
El valor de x es: 5
```

```
>>> print(y)
```

```
None
```

- Nótese que la asignación de `y = print()` no entregará ningún valor de utilidad



STRING

- En la segunda clase del curso conocimos el tipo de dato `string`, y aprendimos que servían para almacenar símbolos o caracteres
- Es posible declararlos usando comillas simples (`'`) o dobles (`"`) y siempre se debe ser consistentes a la hora de **cerrar con el mismo tipo de comilla con el que se abrió** el string

Strings válidos:

`"Hola mundo"`
`'Hola mundo'`
`"abc123"`

Strings no válidos:

`"Hola mundo'`
`'Hola mundo"`
`"Hola
mundo"`



OPERACIONES CON STRINGS

- Algunos operadores aritméticos permiten realizar nuevas operaciones al tratar con strings
- En particular, si existen strings involucrados, los operadores suma (+) y multiplicación (*) se vuelven el operador de **concatenación** y **repetición** respectivamente

```
>>> 'HOLA' + 'MUNDO'  
'HOLAMUNDO'  
>>> 'HOLA' * 3  
'HOLAHOLAHOLA'
```

INDEXACIÓN

0	1	2	3	4	5	6	7	8	9
H	O	L	A		M	U	N	D	O

- Al igual que en las listas, es posible acceder a un caracter en particular del string utilizando **indexación**

```
>>> texto = "HOLA MUNDO"
```

```
>>> print(texto[2])  
L
```

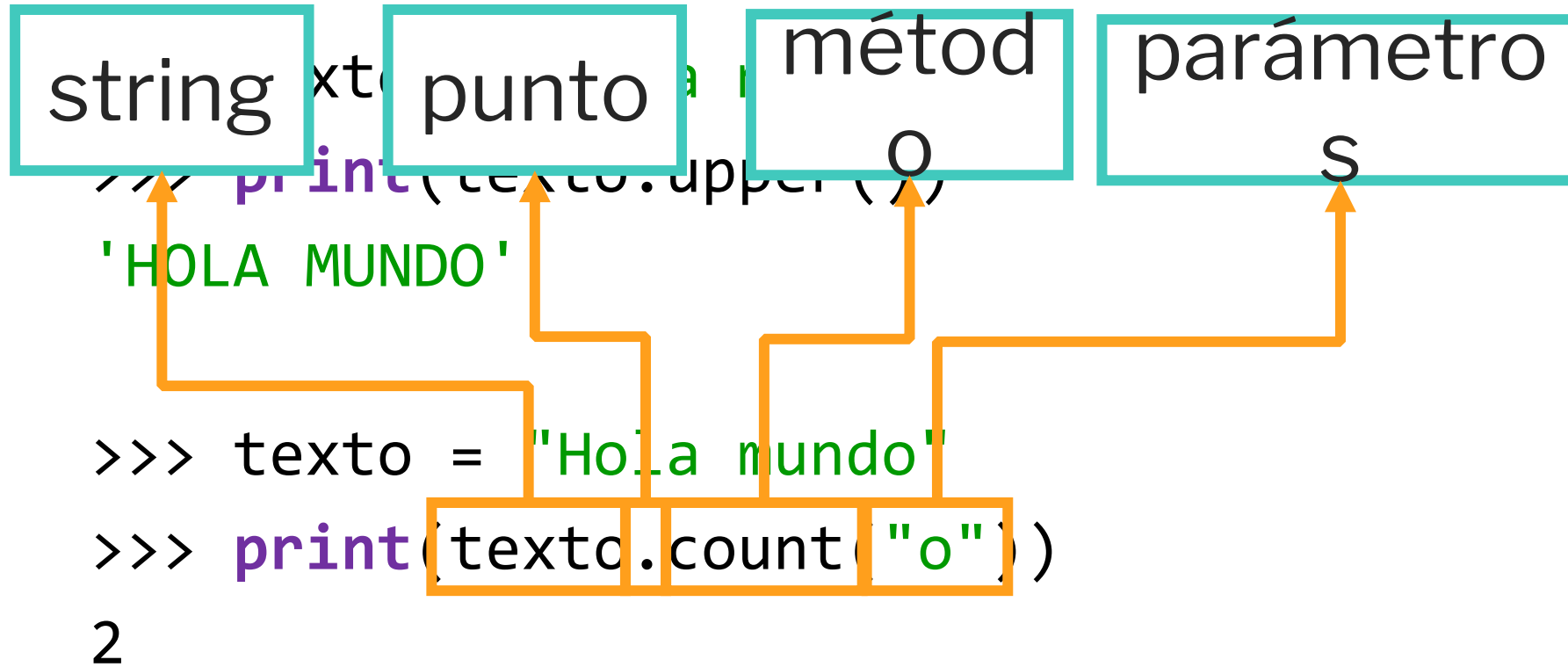



MÉTODOS DE LOS STRINGS

- Para utilizar un método de los strings, se debe utilizar un dato o variable que contenga un string, es decir, un **objeto string**
- A continuación **un punto** (.)
- Luego el **nombre del método**
- Finalmente **los parámetros que el método requiera**, entre paréntesis

`<string>.<método>(<parámetros>)`

MÉTODOS DE LOS STRINGS





MÉTODOS DE LOS STRING

- Algunos métodos de los string son:
 - `<string>.lower()` devuelve el mismo string pero en minúsculas
 - `<string>.upper()` devuelve el mismo string pero en mayúsculas
 - `<string>.islower()` devuelve **True** si todas las letras del string son minúsculas
 - `<string>.isupper()` devuelve **True** si todas las letras del string son mayúsculas
 - `<string>.isdigit()` devuelve **True** si el string sólo esta compuesto por números
- Entre otras



EJERCICIOS



EJERCICIO PROPUESTO 1

- Ahora, con todo lo que hemos aprendido hasta el momento, vamos a resolver un ejercicio:

Samuel quiere hacer una fiesta y para tener algo comestible en ella quiere hacer un queque, usando la receta de su abuelita, sabe que para hacer un queque de 10 porciones necesita 4 huevos, 8 tazas de harina y media taza de leche. Como Samuel todavía está viendo cuántas personas vendrán a la fiesta no quiere estar constantemente calculando las porciones cada vez que alguien le confirma que viene o no. Para ayudarlo construya un programa en Python que permita a Samuel calcular los ingredientes necesarios para que todos los asistentes puedan comer dos porciones de queque



¿CUÁLES SON LAS ENTRADAS DEL PROBLEMA?

CONSIDEREMOS QUE EL PROBLEMA PRESENTA ENTRADAS
VARIABLES Y NOS ALIMENTA ADEMÁS DE INFORMACIÓN
CONSTANTE



ENTRADAS

- En este caso, **la entrada es una sola**, el número de asistentes, por lo que podríamos nombrarla como: **numero_de_asistentes**, **asistentes**, **cantidad_invitados**, etc.
- Cómo la entrada es variable, idealmente deberíamos solicitarla al usuario siempre, por lo que **input()** resulta ideal
- Así, nuestra entrada quedaría:
 - `asistentes = input('Ingrese la cantidad de asistentes: ')`



¿EXISTEN
VALORES
CONSTANTES?



CONSTANTES

- Los valores que **no cambian en el programa**

son:

- Cantidad de huevos que se requieren para un queque
 - `huevos_por_queque = 4`
- Cantidad de harina que se requiere para un queque
 - `tazas_harina_por_queque = 8`
- Cantidad de leche que se requiere para un queque
 - `tazas_leche_por_queque = 1/2`
- Porciones por persona
 - `porciones_por_persona = 2`
- Porciones por queque
 - `porciones_por_queque = 10`

Si bien nuestros años de preparación matemática hacen tentador el realizar los cálculos de huevos, tazas de harina y tazas de leche que se requieren por persona, la idea es aprovechar la **capacidad de cálculo del computador**



¿QUÉ CÁLCULOS
DEBO REALIZAR?



PROCESAMIENTO

- Con las transformaciones que hay que realizar identificadas estas se traducen a **líneas de código en Python** y asignamos los cálculos intermedios a nuevas variables
- Primero a los cálculos de **ingredientes por persona**:

```
huevos_por_persona = huevos_por_queque / porciones_por_queque  
tazas_harina_por_persona = tazas_harina_por_queque / porciones_por_queque  
tazas_leche_por_persona = tazas_leche_por_queque / porciones_por_queque
```



PROCESAMIENTO

- Luego el cálculos de los ingredientes necesarios en **función de la cantidad de asistentes**

```
huevos_requeridos = huevos_por_persona * numero_de_asistentes * porciones_por_persona  
tazas_de_harina_requeridas = tazas_harina_por_persona * numero_de_asistentes*porciones_por_persona  
tazas_leche_por_persona = tazas_leche_por_persona * numero_de_asistentes * porciones_por_persona
```



**FINALMENTE:
¿CUÁLES SON LAS
SALIDAS QUE DEBEMOS
ENTREGAR?**



SALIDAS

- En este caso, la salida debería ser un mensaje indicando al usuario la cantidad que requiere de cada ingrediente
- Para esto podemos usar `print()`
 - `print('Se necesitan:', huevos_requeridos, 'huevos')`
 - `print('Se necesitan:', tazas_de_harina_requeridas, 'tazas de harina')`
 - `print('Se necesitan:', tazas_leche_por_requeridas, 'tazas de leche')`
- Es importante usar mensajes que sean **entendibles para el usuario**.



ABSTRACCIÓN

- Lo que acabamos de hacer corresponde a la **abstracción** de un enunciado de un problema, lo que constituye **la habilidad más importante** a la hora de programar.
- Con la capacidad de abstracción podemos reducir un enunciado de un problema complejo a **sub-problemas manejables** con solución conocida, cuya combinación resuelve el problema completo.
- Esta capacidad **sólo se adquiere** a través de la **ejercitación constante**.



SOLUCIÓN

- En este punto el código que tenemos representa una solución al problema
- Sin embargo, si ejecutáramos el código, tendríamos **un error**
- Esto es porque a pesar de que el **algoritmo es correcto**, este no considera los **elementos de programación en Python** que vimos la clase anterior
- Debemos realizar los **cambios de tipo necesarios** para que la solución entregada sea correcta



CONSTANTES

```
huevos_por_queque = 4  
tazas_harina_por_queque = 8  
tazas_leche_por_queque = 1/2  
porciones_por_persona = 2  
porciones_por_queque = 10
```

ENTRADAS

```
numero_de_asistentes = input('Ingrese la cantidad de asistentes: ')
```

PROCESAMIENTO

Cambio la cantidad de asistentes de string a entero

```
numero_de_asistentes = int(numero_de_asistentes)
```

Hago los cálculos por persona

```
huevos_por_persona = huevos_por_queque / porciones_por_queque  
tazas_harina_por_persona = tazas_harina_por_queque / porciones_por_queque  
tazas_leche_por_persona = tazas_leche_por_queque / porciones_por_queque
```

Hago los cálculos por asistente

```
huevos_requeridos = huevos_por_persona * numero_de_asistentes * porciones_por_persona  
tazas_de_harina_requeridas = tazas_harina_por_persona * numero_de_asistentes * porciones_por_persona  
tazas_leche_por_requeridas = tazas_leche_por_persona * numero_de_asistentes * porciones_por_persona
```

SALIDA

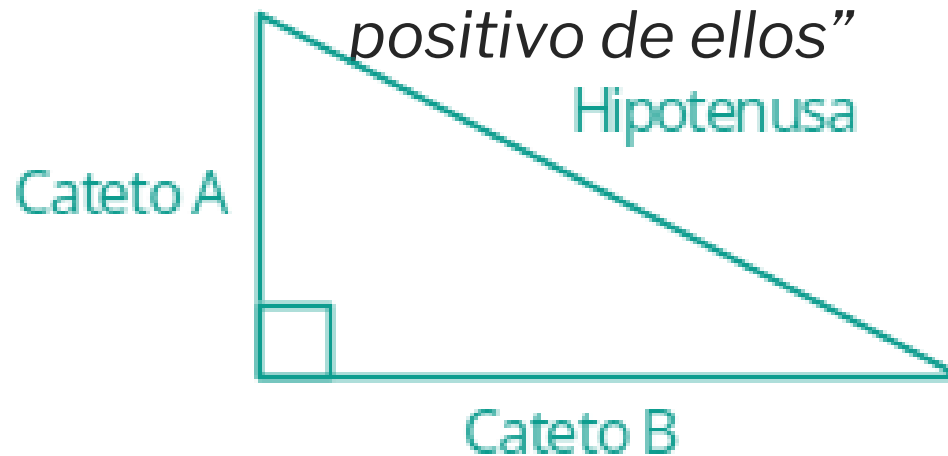
```
print('Se necesitan:', huevos_requeridos, 'huevos')  
print('Se necesitan:', tazas_de_harina_requeridas, 'tazas de harina')  
print('Se necesitan:', tazas_leche_por_requeridas, 'tazas de leche')
```



EJERCICIO PROPUESTO 2

- Utilicemos los conceptos vistos para resolver el siguiente problema

“Usando el Teorema de Pitágoras, cree un programa en Python que permita el cálculo de la hipotenusa de un triángulo rectángulo a partir del valor de sus catetos, catetoA y catetoB, para cualquier valor positivo de ellos”





EJERCICIO PROPUESTO 2

- **ENTRADAS:**
 - Valores de los catetos
- **SALIDAS:**
 - Valor de la hipotenusa
- **ALGORITMO:**
 1. Solicitar los valores de los catetos
 2. Calcular la hipotenusa
 3. Mostrar por consola el valor de la hipotenusa



EJERCICIO PROPUESTO 2

Calcula la hipotenusa de un triángulo rectángulo dados sus catetos

ENTRADAS

cateto_uno = float(input("Ingrese el valor de un cateto: "))

cateto_dos = float(input("Ingrese el valor del otro cateto: "))

PROCESAMIENTO

hipotenusa = (cateto_uno ** 2 + cateto_dos ** 2) ** 0.5

SALIDAS

print("El valor de la hipotenusa del triángulo rectángulo con
catetos",cateto_uno,"y",cateto_dos,"es:",hipotenusa)



EJERCICIO PROPUESTO 3

- Implemente un programa en Python que calcule la ecuación de la recta que relaciona la escala de temperatura Fahrenheit con la escala de grados Celsius, sabiendo que:
- 32 grados Fahrenheit equivalen a 0 grados Celsius
- 50 grados Fahrenheit equivalen a 10 grados Celsius
- $y = m x + b;$ $m = \frac{\Delta y}{\Delta x}$



EJERCICIO PROPUESTO 3

- **CONSTANTES:**
 - Dos puntos de la recta
- **SALIDAS:**
 - Pendiente, ordenada al origen
- **ALGORITMO:**
 1. Asignar las abscisas y ordenadas de los puntos
 2. Calcular la pendiente (m)
 3. Calcular la ordenada al origen (b)
 4. Mostrar por consola la ecuación



Calcula la relación lineal entre grados Fahrenheit y grados Celsius

CONSTANTES

Abscisa del punto uno

PUNTO_UNO_X = 32

Ordenada del punto dos

PUNTO_UNO_Y = 0

Abscisa del punto uno

PUNTO_DOS_X = 50

Ordenada del punto dos

PUNTO_DOS_Y = 10

PROCESAMIENTO

Calcula la pendiente

$m = (PUNTO_DOS_Y - PUNTO_UNO_Y) / (PUNTO_DOS_X - PUNTO_UNO_X)$

Calcula la ordenada al origen

$b = PUNTO_UNO_Y - m * PUNTO_UNO_X$

SALIDAS

Entrega el valor de la pendiente

print("La pendiente de la ecuación es: ",m)

Entrega el valor de la ordenada al origen

print("La ordenada al origen es: ",b)



TAREAS PARA TRABAJO AUTÓNOMO

1. Revisar el apunte:
 - Estructuras de control: Decisiones en Google Colab (Disponible en: https://github.com/PROGRA-FING-USACH/Material/blob/main/Lecturas/03_Control_de_flujo.ipynb)
2. Revisar la guía 1 y 2.



¿CONSULTAS?