VARIABLES Y FUNCIONES BUILT-IN

Introducción a la Programación (11071)
Departamento de Ciencias Básicas
Universidad Nacional de Luján

MATERIAL RELACIONADO

TEORIA

👩‍💻 **Teórica 2 (A, B, y C)** - Conceptos, partes, y depuración de programas.

📘 **Apunte Cap. 2** - Programas sencillos.

PRÁCTICA

📋 **TP III** - Variables y funciones built-in.

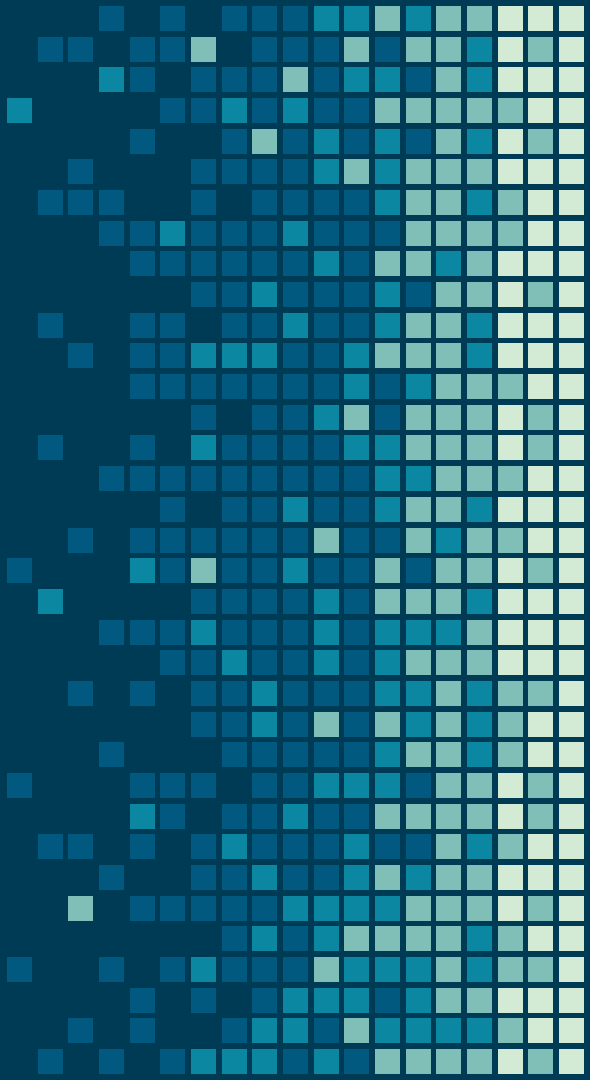
📋 **TP IV** - Interacción con el usuario.

💻 **Notebook 1** - Datos y operaciones.

ENTREGABLES

💾 **HomeWork 0** - Hola mundo.

REPASO DE CONCEPTOS



VARIABLES

DEFINICIÓN

Una **variable** es un espacio de memoria al que le damos un nombre (***identificador***), y en donde almacenamos un dato.

TIPO

En Python, toda variable tiene asociado implícitamente un ***tipo***, que describe la naturaleza del dato almacenado, y restringe las operaciones que se pueden aplicar a él.

TIPADO EN PYTHON

DINÁMICO

Una variable puede tomar valores de tipos diferentes a lo largo de la ejecución del programa. Los chequeos de tipo en las operaciones se hacen en runtime. Ejemplo:

```
x = 'Hola!'
x = 10
m = 'Chau!'
print(x + m)    # el error se detectará en runtime
```

FUERTE

...

TIPADO EN PYTHON

DINÁMICO

...

FUERTE

Las operaciones válidas para una variable se limitan a su tipo; Python no realiza conversiones implícitas de los tipos primitivos. Ejemplo: el string *'360'* siempre será considerado un string, aunque posea sólo números. Las operaciones válidas serán las correspondientes al tipo *string*, no a enteros. Work-around: utilizar *casting*.

```
x = 10
m = 'Chau!'
print(m + str(x))
```

CASTING

DEFINICIÓN

Acción de “transformar” o “traducir” el tipo de un valor.

UTILIDAD

Utilizamos *casting* cuando necesitamos “interpretar” una variable como si fuera de un tipo diferente. Ejemplo:

```
x = ‘10’
print(5 + x)
```

CASTING

DEFINICIÓN

Acción de “transformar” o “traducir” el tipo de un valor.

UTILIDAD

Utilizamos *casting* cuando necesitamos “interpretar” una variable como si fuera de un tipo diferente. Ejemplo:

```
x = ‘10’
print(5 + x)  <<< Error de tipos!
```

CASTING

DEFINICIÓN

Acción de “transformar” o “traducir” el tipo de un valor.

UTILIDAD

Utilizamos *casting* cuando necesitamos “interpretar” una variable como si fuera de un tipo diferente. Ejemplo:

```
x = ‘10’
print(5 + int(x))
```

CASTING

DEFINICIÓN

Acción de “transformar” o “traducir” el tipo de un valor.

UTILIDAD

Utilizamos *casting* cuando necesitamos “interpretar” una variable como si fuera de un tipo diferente. Ejemplo:

```
x = '10'
print(5 + int(x))     # casteo de x para interpretarlo como un entero
```

CASTING

FUNCIONES DE CASTING

- **int(***variable***)** : intenta convertir la variable a entero.
- **str(***variable***)** : intenta convertir la variable a string.
- **float(***variable***)** : intenta convertir la variable a flotante.
- **hex(***variable***)** : intenta convertir la variable a hexadecimal.
- **oct(***variable***)** : intenta convertir la variable a octal.

... etcétera.

Hacer un casting (convertir explícitamente el tipo de una variable) puede producir un error en runtime si la conversión no es posible.

FUNCIONES BUILT-IN

FUNCIÓN

Algoritmo codificado de manera independiente a la secuencia de instrucciones del programa principal, que se puede invocar mediante un *identificador*.

FUNCIONES BUILT-IN

Funciones provistas por el lenguaje para resolver tareas comunes. Ejemplos en Python:

- **print()**: muestra algo en la salida estándar.
- **input()**: lee un dato desde la entrada estándar.
- **int()**: transforma el tipo de una variable a entero.

Listado de funciones built-in:
https://docs.python.org/3/library/functions.html

PARÁMETROS

DEFINICIÓN

Los datos que reciben las funciones como entrada se llaman ***parámetros*** o ***argumentos***. Una función puede recibir cero, uno, o más parámetros.

SINTAXIS

Toda función se invoca mediante su identificador, seguido de un par de paréntesis. Si la función recibe parámetros, los mismos se listan entre los paréntesis, y separados por coma en caso de que necesite más de uno. Ejemplos:

```
print('Estoy programando!')
input()
int(x)
range(1, 11)
```

EJERCICIOS GUIADOS

Para resolución de TP III y TP IV.

EJERCICIO MODELO (TP III)

OBJETIVO DEL EJERCICIO

Comprender y ejercitar los conceptos de **variables** y **tipos**.

ENUNCIADO

Codifique un programa en Python que almacene dos números enteros; asigne dos valores cualesquiera a dichas variables. Finalmente, el programa debe mostrar en pantalla el resultado de multiplicar ambos valores.

CONTINUACIÓN

Luego de resolver la primera parte del ejercicio, cambie el valor de una de las variables para que sea de tipo literal (el número encerrado entre comillas). Vuelva a ejecutar el programa. ¿Qué ha sucedido? ¿Cómo puede solucionarse el problema (manteniendo a uno de los valores como literal)?

EJERCICIO MODELO (TP IV)

OBJETIVO DEL EJERCICIO

Comprender y ejercitar los conceptos de **funciones built-in** e **interacción con el usuario**.

ENUNCIADO

Tomando como base el enunciado del ejercicio anterior, modifique el programa para que ahora los valores de las variables sean ingresados por el usuario en lugar de ser un valor hardcodeado (arbitrario). El programa debe seguir mostrando el resultado de multiplicar ambos valores.

TIPS

La función built-in ***input()*** siempre asume que el valor leído desde teclado es un literal. Utilice las técnicas aprendidas para solucionar este inconveniente a la hora de realizar operaciones aritméticas.


31
HTML
C++
CSS
JAVA