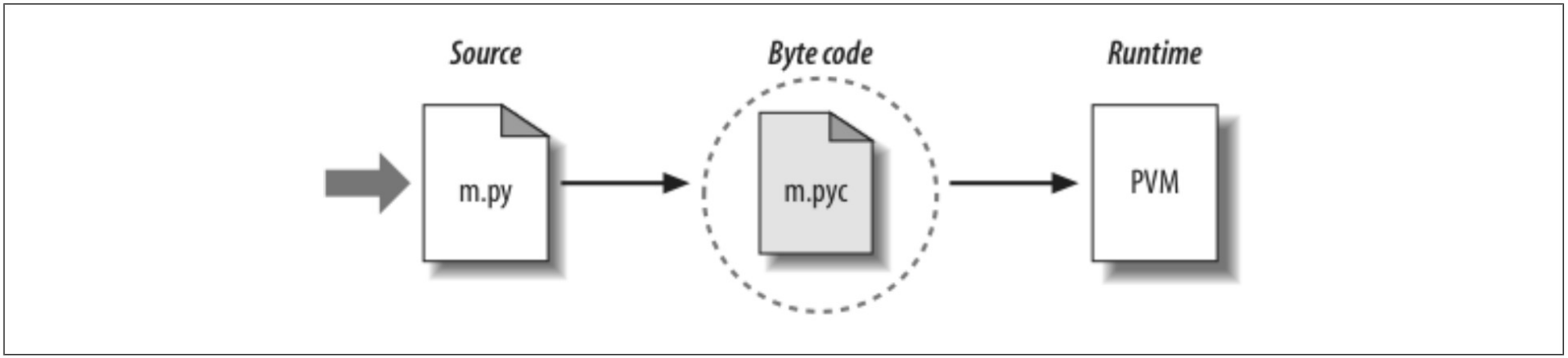
Laboratorio de Elementos Programables II

Tipos de Datos

Runtime execution model



UNIVERSIDAD IBEROAMERICANA PUEBLA

¿Qué es un Identificador?

- Es la manera de identificar variables, funciones, constantes y objetos.

- Python es sensible al lenguaje (Distingue minúsculas y mayúsculas)

Identificadores

- Letras: de la A – Z o a – z.
- Espacios: “ _ ”
- Números: 0 al 9

- Validos

  age_of_dog taxRateY2K move_name a_123 nombre
  PrintHeading ageOfHorse _temp j a23b9 noMbre

- No Validos (¿Por qué?)

  age# 2000TaxRate Age-Of-Cat sqrt

Variables en Python

- Una variable es un espacio de memoria al que podemos referirnos con un identificador, en el cual es posible cambiar el valor alojado.

- Una declaración consta de su nombre y tipo de dato.

```cpp
#include <iostream>
using namespace std;

int main()
{
  int num;
  num=30;

  int num1=30;
}
```

```
num = 30
num1 = 31

print(num)
print(num1)
```

```
30
31
```

UNIVERSIDAD IBEROAMERICANA PUEBLA

Tipos de Datos

Datos numéricos:

- Integer
- Floating Point
- Double Floating Point
- Usigned int
- Short int
- Long int

Caracteres:

- Char
- signed char
- usigned char
- wchar_t
- char32_t

Object type	Example literals/creation
Numbers	`1234, 3.1415, 3+4j, Decimal, Fraction`
Strings	`'spam', "guido's", b'a\x01c'`
Lists	`[1, [2, 'three'], 4]`
Dictionaries	`{'food': 'spam', 'taste': 'yum'}`
Tuples	`(1, 'spam', 4, 'U')`
Files	`myfile = open('eggs', 'r')`
Sets	`set('abc'), {'a', 'b', 'c'}`
Other core types	Booleans, types, None
Program unit types	Functions, modules, classes (Part IV, Part V, Part VI)
Implementation-related types	Compiled code, stack tracebacks (Part IV, Part VII)

Tipos de Datos

Text Type:	`str`
Numeric Types:	`int`, `float`, `complex`
Sequence Types:	`list`, `tuple`, `range`
Mapping Type:	`dict`
Set Types:	`set`, `frozenset`
Boolean Type:	`bool`
Binary Types:	`bytes`, `bytearray`, `memoryview`

Example	Data Type
`x = "Hello World"`	str
`x = 20`	int
`x = 20.5`	float
`x = 1j`	complex
`x = ["apple", "banana", "cherry"]`	list
`x = ("apple", "banana", "cherry")`	tuple
`x = range(6)`	range
`x = {"name" : "John", "age" : 36}`	dict
`x = {"apple", "banana", "cherry"}`	set
`x = frozenset({"apple", "banana", "cherry"})`	frozenset
`x = True`	bool
`x = b"Hello"`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview

Tipos de Datos. Declaración especifica.

Example	Data Type
`x = str("Hello World")`	str
`x = int(20)`	int
`x = float(20.5)`	float
`x = complex(1j)`	complex
`x = list(("apple", "banana", "cherry"))`	list
`x = tuple(("apple", "banana", "cherry"))`	tuple
`x = range(6)`	range
`x = dict(name="John", age=36)`	dict
`x = set(("apple", "banana", "cherry"))`	set
`x = frozenset(("apple", "banana", "cherry"))`	frozenset
`x = bool(5)`	bool
`x = bytes(5)`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview

Tipos de Datos Enteros. Generalización

1 BYTE = 8 BITS

Tipo	Tamaño en Bytes	Valor Mínimo	Valor Máximo
char	1	-128	127
short	2	-32,768	32,767
int	2-4	-32,768	32,767
long	4	-2,147,483,648	2,147,483,647

NOTA: Los valores y tamaños son dependientes de la maquina, compilador, procesador, controlador, etc..

Tipos de Datos Decimales. Generalización

Tipo	Tamaño en Bytes	Valor Mínimo	Valor Máximo
float	4	3.4E-38	3.4E+38
double	8	1.7E-308	1.7E+308
Long double	10	3.4E-4932	1.1E+4932

NOTA: Los valores y tamaños son dependientes de la maquina, compilador, procesador, controlador, etc..

Tipos de Variables

Variables Globales: Son aquellas que pueden llamarse desde cualquier punto del código, funciones, main, clases, etc..

Se declaran fuera de cualquier función, incluida la función main.

```
[36] Age = 25

[37] Mario = 0
     if (Mario == 1):
       print(Age)
       Test = 0
     else:
       print('NOP')

     NOP

[38] print(Test)
```

Tipos de Variables

Variables Locales: Son aquellas que pueden llamarse unicamente desde su punto de declaración, los cuales están limitados por la estructura de control usada.

Pueden estar dentro de un loop, función, clase, etc.

```
Mario = 0
if (Mario == 1):
  print(Age)
  Test = 0
else:
  print('NOP')
```

Operadores Aritméticos

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

Operadores de Asignación

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

Operadores de Comparación

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Operación Lógica, sin contexto

OPERATOR	DESCRIPTION	SYNTAX
and	Logical AND: True if both the operands are true	x and y
or	Logical OR: True if either of the operands is true	x or y
not	Logical NOT: True if operand is false	not x

Ejercicio

Comencemos aplicando operaciones básicas en python.

Actividad en clase: 2022_Actividad-1_180233

- Declaración de numeros y cadenas.
- Conversión de tipo de dato.
- Especificación de tipos de variables.
- Operaciones con variables.
- Función Print.
- Operaciones Lógicas con variables.
- Inputs

¿Inputs y Outputs?

Salida básica:

Print(mensaje o variable)
Extra info

Entrada básica:

Valiable = input(“mensaje”) #El mensaje es opcional
Extra info

Gracias


LA VERDAD
NOS HARÁ LIBRES
UNIVERSIDAD
IBEROAMERICANA
PUEBLA
®