

Taller de introducción a Python

M.ª Isabel Ruiz Martínez

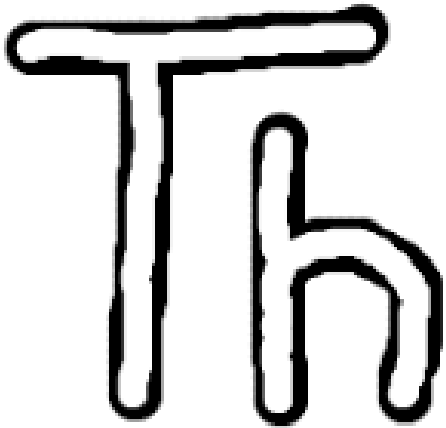
¿Qué es Python?

- Lenguaje de programación
- Interpretado
- Tipado dinámico
- Código legible
- Multiparadigma
- Multiplataforma
- Libre y gratis



IDE de Python

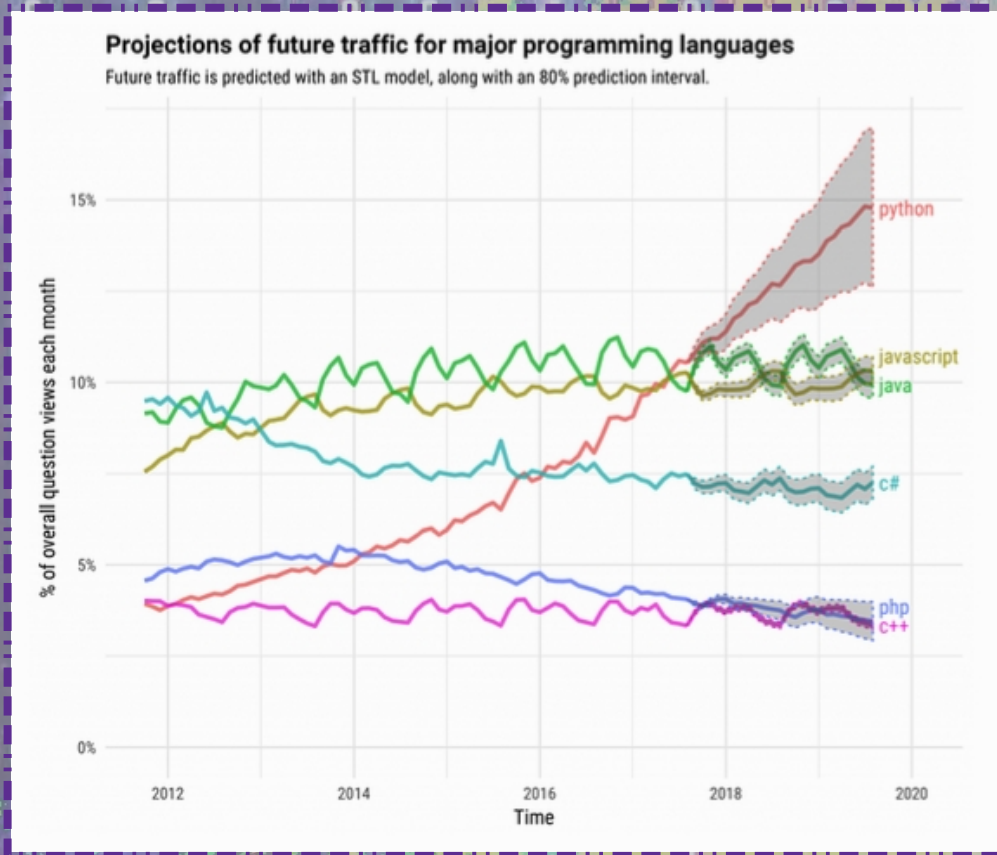
- Thonny



¿Para qué se usa Python?

- Python en la Inteligencia Artificial (AI)
- Python en Big Data
- Python en Data Science
- Python en Frameworks de Pruebas
- Python en Desarrollo Web
- Robótica

¿Para qué se usa Python?



M.^a Isabel Ruiz Martínez

Curso de iniciación a Python de PROGRAMA ERGO SUM

- <https://www.programoergosum.com/cursos-online/raspberry-pi/244-iniciacion-a-python-en-raspberry-pi/introduccion>
- <https://www.aprendeprogramando.es/cursos-online/python>

Programas de ejemplo

- <https://github.com/mruiz54/>

Código pythonic o unpythonic

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.

M.^a Isabel Ruiz Martínez

Código pythonic o unpythonic

- Frente a la ambigüedad, rechaza la tentación de adivinar.
 - Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
 - Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.
 - Ahora es mejor que nunca.
 - Aunque nunca es a menudo mejor que ya mismo.
 - Si la implementación es difícil de explicar, es una mala idea.
 - Si la implementación es fácil de explicar, puede que sea una buena idea.
 - Los espacios de nombres (namespaces) son una gran idea ¡Hagamos más de esas cosas!
- (hace referencia a Guido van Rossum, el autor del lenguaje de programación Python)

“Hello World” en Python

- `print(“Hello World”)`

Operadores

- Operadores aritméticos
- Operadores lógicos
- Expresiones compuestas

Operadores aritméticos

Suma	+
Resta	-
División	/
Multiplicación	*
Potencia	**
Resto	%
Cociente	//

Operadores lógicos

Igualdad	==
Distinto	!=
Mayor que	>
Mayor o igual que	>=
Menor que	<
Menor o igual que	<=

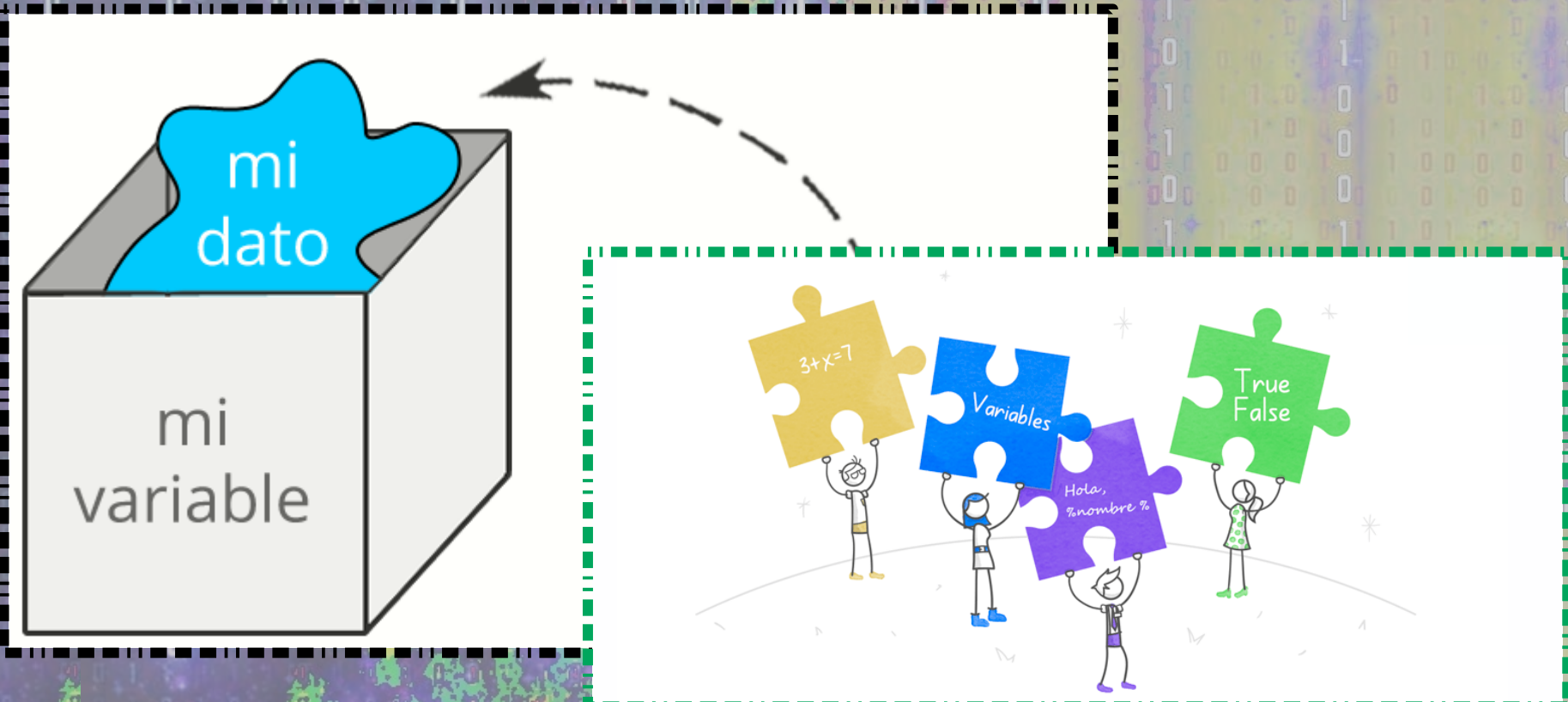
Expresiones compuestas

Negación	not
Y	and
O	or

Tipos de datos

- Numéricos
- Booleanos
- Cadenas
- Listas
- Tuplas
- Diccionario

¿Qué es una variable?



¿Cómo solicitar datos por el teclado?

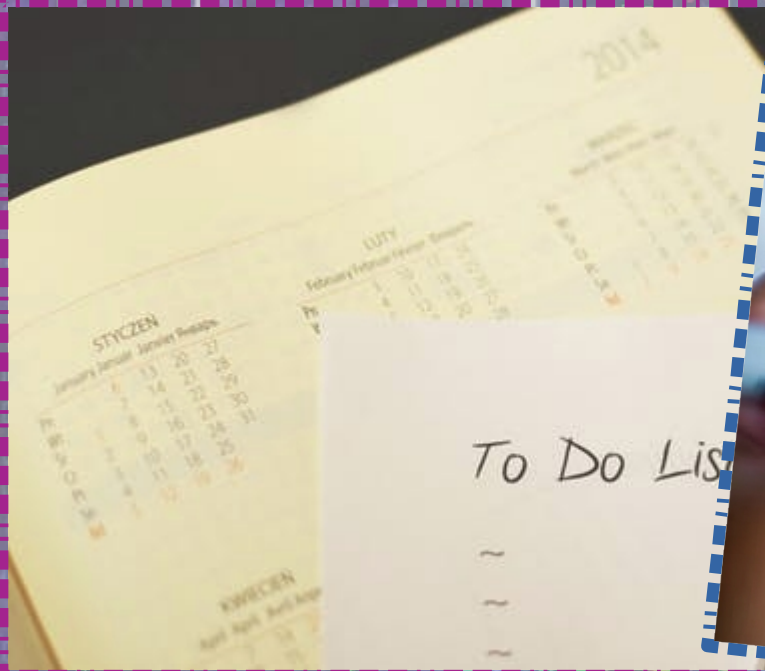
- `input("Introduce una cadena: ")`
- `int(input("Introduce un número entero: "))`
- `float(input("Introduce un número decimal: "))`

¿Cómo imprimir datos por pantalla?

- `print("Hola Mundo")`
- `print("El número entero es: " + str(entero))`
- `print("El número decimal es: " + str(decimal))`

¿Qué es una lista?

- `nombre_lista = [elemento1, ..., elementon]`



Condiciones

```
if (codicion):  
    Algo  
else:  
    Otra cosa
```

```
if (codicion1):  
    Algo  
elif (condicion2):  
    Otro Algo  
else:  
    Otra cosa
```


Bucles

- Bucle for
for i in range(valor_inicial,menor_que_este_valor,paso):
 Cuerpo
- Bucle while
while condicion:
 Cuerpo

Funciones

- Función sin parámetros

```
def nombre_funcion ():
```

```
    Cuerpo
```

- Función con parámetros

```
def nombre_funcion (parametro1,...,parametron):
```

```
    Cuerpo
```


Reto

- Crea una función que reciba un número y calcule si es primo o no.



M.ª Isabel Ruiz Martínez



FIN

Espero que os haya gustado

M.^a Isabel Ruiz Martínez