

ММП 2025/2026

Викладач Канцедаль Георгій Олегович

Python

Як каже нам вікіпедія, Python - інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією. Розроблена на початку 1990-х років Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, аспектно-орієнтована та функціональна.

Можливості мови програмування Python

Що таке Python? Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Пайтоні оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
гнучкість та масштабованість Пайтона дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;
розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
Пайтон – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
у Python — колосальна спільнота односторонців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек.

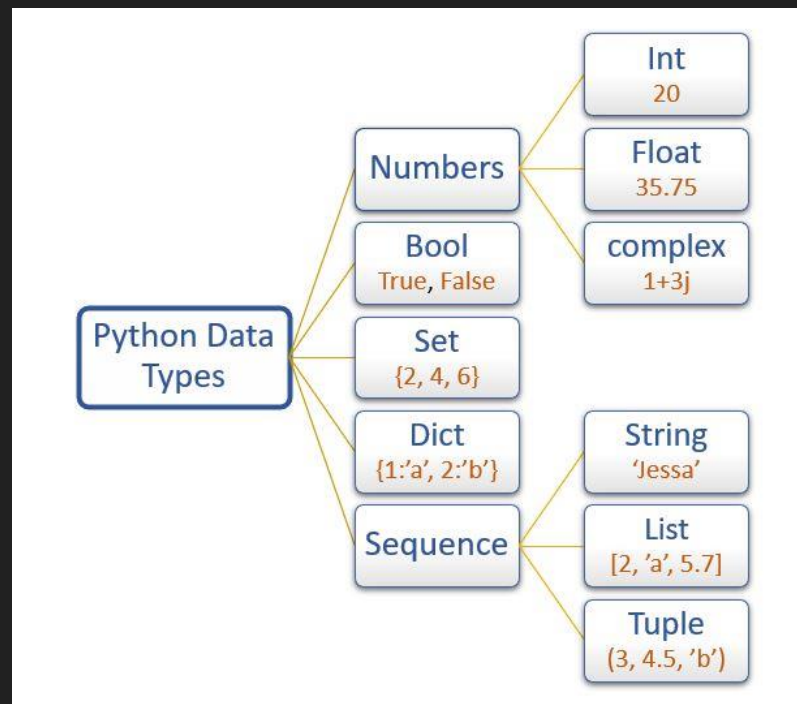
Бібліотеки Python

Різні бібліотеки Пайтон використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Мова універсальна

Бажаєте зробити веб сайт не проблема, існує бажання зробити програму під ПК також можна, хочете писати нейронні мережі без проблем, парсинг даних, ігри не проблема. Такий вибір зумовлений великою кількістю бібліотек, та фреймворків.

Основні дата типи



Name	Syntax	Example	Conversion
Boolean	<code>True, False</code>		<code>bool()</code>
Integer	<code>0, 13, -3</code>		<code>int()</code>
Float	<code>1.5, 1e-4</code>		<code>float()</code>
Complex	<code>1j, 2+0.5j</code>		<code>complex()</code>
String	<code>'abc', "abc", """abc"""</code>		<code>str()</code>
List	<code>[1, 1, 2, 3]</code>		<code>list()</code>
Tuple	<code>(1, 1, 2, 3)</code>		<code>tuple()</code>
Set	<code>{1, 2, 3}</code>		<code>set()</code>
Dictionary	<code>{"a": 1, "c": 5}</code>		<code>dict()</code>

Операції над даними

Arithmetic Operators

Operator	Meaning	Example
+	Addition	$4 + 7 \longrightarrow 11$
-	Subtraction	$12 - 5 \longrightarrow 7$
*	Multiplication	$6 * 6 \longrightarrow 36$
/	Division	$30 / 5 \longrightarrow 6$
%	Modulus	$10 \% 4 \longrightarrow 2$
//	Quotient	$18 // 5 \longrightarrow 3$
**	Exponent	$3 ** 5 \longrightarrow 243$

Операції над даними

OPERATOR | SYNTAX |

and | a and b |

#Example-----
if condition1 and condition2 and condition3:
 all_conditions_met == true
else:
 all_conditions_met == false

OPERATOR | SYNTAX |

or | a or b |

#Example
if condition1 or condition2 or condition3:
 any_condition_met == true
else:
 any_condition_met == false

OPERATOR | SYNTAX |

not | not a |

if not equal_to_condition:
 equal_to_condition == false
else:
 equal_to_condition == true

Організація коду функції

```
#Letters, Nums,
#& underscores
#only, can't    =7
#start w/ num  |          _ 5 is the default value if no
#              |          | value is passed
#              v          v          v
def funcName(param1, param2=5): #The colon terminates func header
    str(param1) #Typecasting is temporary
    strVar = str(param1) # To make typecast perma, assign to var
    param1 = param2 # 5 is passed to param1
    return param1 # 5 is returned
#      ^      ^
#      |      |_ return value is optional, if no return value, func
#Optional      will end returning to func call
#return

x = funcName(7)
print(x) # prints 5
```

кінець

- Рекомендовано лаба 0, лаба 1 для проходження матеріалу.