

# Python

Все, що потрібно

[DAN.İT]  
EDUCATION

# Що таке Python

**високорівнева мова \***  
програмування загального  
призначення, орієнтована на  
підвищення продуктивності  
розробника та читання коду



# Типи мов програмування

Є багато типів мов програмування. Це пов'язано з тим, що базову мову, якою говорять комп'ютери, що складається тільки з одиниць і нулів, занадто складно використовувати людині.

Потрібна якась мова, яку люди можуть використовувати, щоб віддавати комп'ютерам ефективні накази в розумні терміни.

# Високорівневі мови



import conio.h

100010110

# Низькорівневі мови

## Парадигми програмування

- Об'єктно-орієнтоване програмування.
- Функціональне програмування.
- Процедурне програмування.
- Логічне програмування.
- Програмування баз даних.

Prolog

C++

Pascal

SQL

Haskell

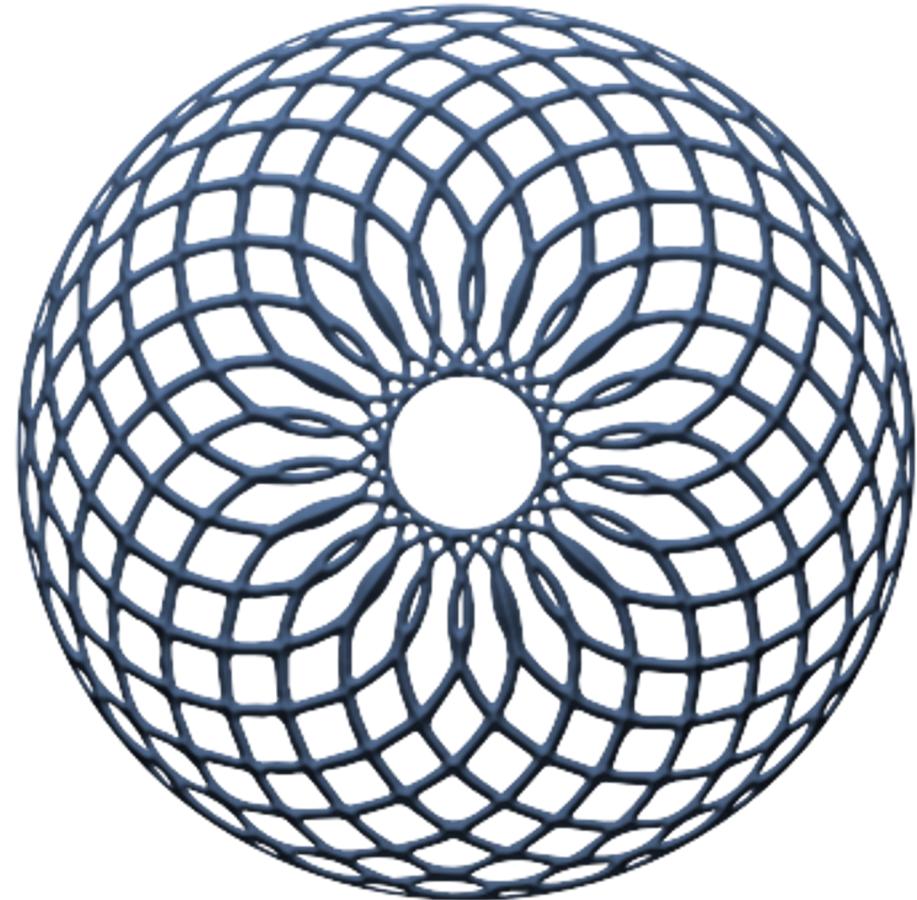
# Навіщо потрібен Python?

- ❑ Системне програмування
- ❑ Графічний інтерфейс
- ❑ WEB
- ❑ Інтеграція компонентів
- ❑ Компоненти баз даних
- ❑ Швидке створення прототипів
- ❑ Програмування математичних та наукових обчислень
- ❑ Штучний інтелект
- ❑ Робота з даними

- Об'єктно-орієнтований
- Free-of-charge
- Multiplatform
- Powerful
- Community and support
- Cozy

- Нестандартний синтаксис
- Повільність
- *Skyrocketing* складність мови

- Інтерпретація
- Динамічна типізація
- Мультипарадигменне
- програмування



# Звідки взяти Python?

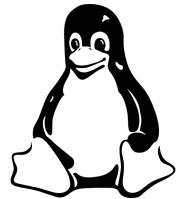
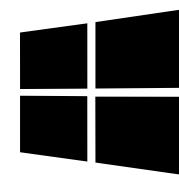
# Python IDLE

**Сама мова програмування.**

За допомогою цієї програми  
ми можемо починати писати та  
комп'ютер зрозуміє, що ми від  
нього хочемо.

Але тут це робити не зовсім  
зручно.

[www.python.org](http://www.python.org)



**150 Mb HDD/SSD**

**2 Gb RAM**

**1.0 GHz Single-core CPU**



# PyCharm CE



**Зручне середовище для розробки**

Тут буде зручно створювати наші проекти та редагувати код.

Можна вибрати іншу програму, але ця створена саме для Python

<https://www.jetbrains.com/pycharm>

**2 Gb HDD/SSD**  
**2 Gb RAM**  
**1.4 GHz Single-core CPU**



Hello world

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```



```
#include <iostream>
int main() {
    std::cout << "Hello World!";
    return 0;
}
```



```
print("Hello world!")
```



Зберігаємо дані

# Змінні

my\_var = 5

Іменоване місце в пам'яті  
пристрою, доступне при зверненні  
«за іменем»



namer, my\_name, \_my\_name, name1

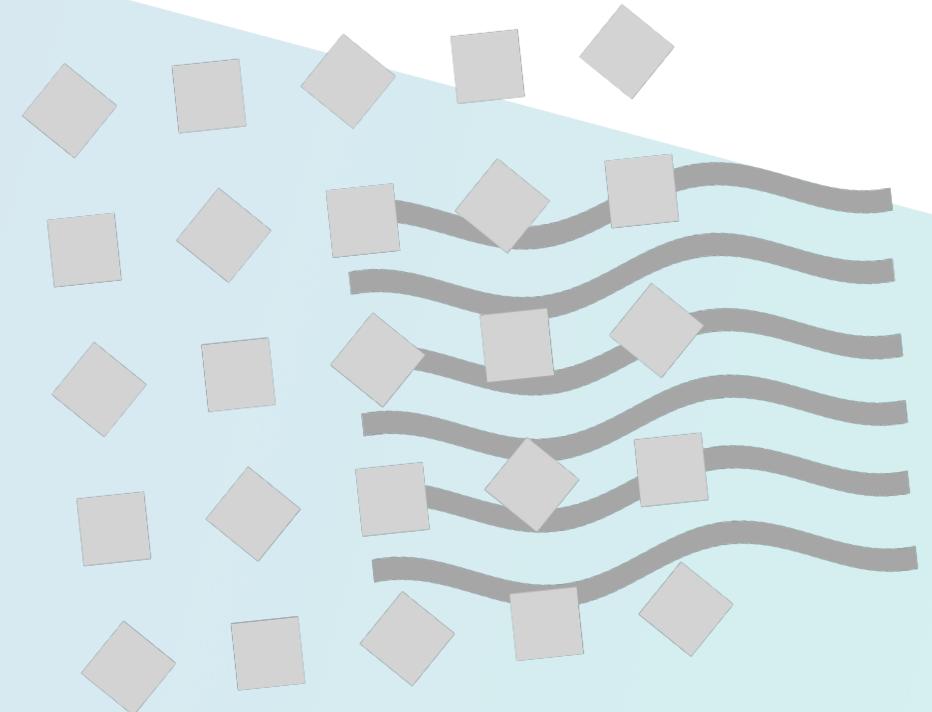
1, print, !heloo, 123qwerty

# Константи

CONST= 5

**“Змінна”, яку не можна змінити.**

Однак у Python їх просто немає.  
Якщо ви хочете убездечити змінну  
від зміни – напишіть її назву  
великими літерами (але її все одно  
можна буде поміняти)



# Введення та виведення (I/O)

**Введення з  
клавіатури**

`input(a)`



**Виведення  
на екран**

`print(a)`

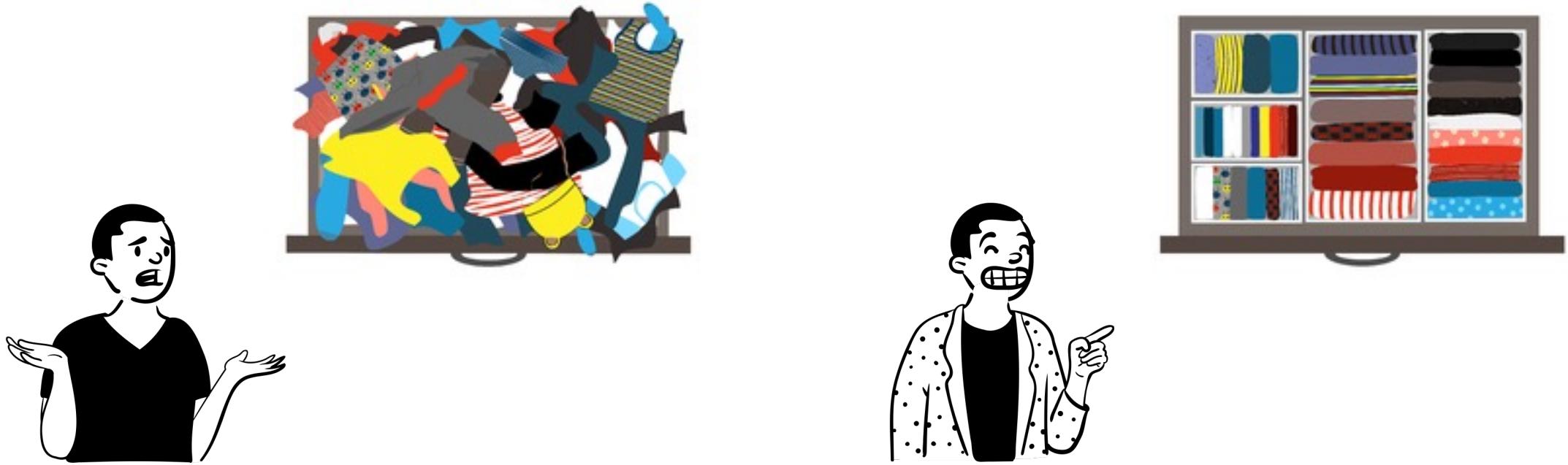
1. Run the interpreter. Print "Hello, world" and then print your name.
2. Write a script which prints "Hello, world" and then prints your name, that will be input inside code by keyboard Run the script from the terminal and from PyCharm.



# Типи даних

- 👍 Числа
- 👍 Рядки
- 👍 Списки
- 👍 Словники
- 👍 Кортежі
- 👍 Файли
- 👍 Множини
- 👍 Інші базові типи
- 👍 Типи структурних елементів програм
- 👍 Типи, що стосуються реалізації





## Порядок та швидкість роботи



## Принцип застосування



Резервування місця у пам'яті

# Числа

my\_var = 5

Числа потрібні для відображення  
подій під час роботи програми,  
математики та статистики

integer

Цілі

ост  
bin  
hex

float

decimal

complex

Дробові

fractions



# Цілі

# 5

Цілі числа відображають природну нумерацію і найчастіше служать як інструмент математики та керування потоком. Автоматично забезпечують необмежену точність великих значень

до 3.10 розмір значення був  $\pm\inf$ . Зараз до  $10^{400}$

Название(EN)	Название(UK)	Название(RU)	У системі	Приклад
Integer	Цілі Десяткові	Целые Десятичные	int	9000

<b>Название(EN)</b>	<b>Название(UK)</b>	<b>Название(RU)</b>	<b>У системі</b>	<b>Приклад</b>	<b>У 10-й (integer)</b>
Binary	Двійкова	Двоичные	bin	0b10010011	147
Octal	Восімкова	Восьмеричные	oct	0o223	147
Hexadecimal	Шістнадцяткова	Шестнадцатеричная	hex	0x93	147

# Дробові

# 3.14

є суто математичними значеннями і дозволяють зберігати інформацію з високою точністю

Название(EN)	Название(UK)	Название(RU)	В системе	Пример
Float	Число з рухомою комою	Число с плавающей точкой	float	4.20

Название(EN)	Название(UK)	Название(RU)	В системе	Пример	В 10-й (
Decimal fixed point	десяткові числа з фіксованою точкою	десятичные числа с фиксированной точкой	decimal.Decimal	Decimal('0.14')	0.14
Complex	Комплексні	Комплексные	complex	(5+0j)	5
Fractions	Раціональні числа	рациональные числа	fractions.Fraction	2/3	Fraction(2,3)

# Усічення

**A=3.0**

**B=3.9**

**A=3**

**B=3**

# Булеві

Логічний тип даних. Можливий двох видів – True та False

Потрібний для перевірки, що відбувається з умовою

# Арифметичні оператори

$x + y$	Додавання
$x - y$	Віднімання
$x * y$	Множення
$x / y$	Ділення
$x // y$	Отримання цілої частини від ділення
$x \% y$	Залишок від ділення
$-x$	Зміна знака числа
$x**y$	У у степені числа X

# Оператори порівняння

$x < y$	Менше
$x > y$	Більше
$x \leq y$	Менше або дорівнює
$x \geq y$	Більше або дорівнює
$x == y$	Дорівнює
$x != y$	Не дорівнює

3. Обчисліть, скільки секунд у добі
4. Обчисліть, скільки секунд у році
5. Відкрито нову планету! Введіть назву планети і кількість земних днів в одному році на цій планеті (на приклад один рік на новій планеті – 532 земних дні) Обчисліть і надрукуйте, скільки секунд становить рік планети.  
(«В одному році Zork 22118400 секунд»)
6. Візьміть як ціле число вік людини в місяцях і виведіть «Істина», якщо особа старше 18 років, інакше «False».

