

Вступ у Python.

Базові поняття мови



1. Введення в Python

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування високого рівня із строгою динамічною типізацією.

Офіційний сайт мови
програмування Python <https://www.python.org/>

Історія Python

- Гвідо ван Россум (нідерланд. Guido van Rossum) – автор мови Python
- Початок розробки в кінці 1980-х років – спроба виправити мову **ABC**
- Перший реліз Python відбувся у 1991 році
- Назва мови походить від британського комедійного телешоу 1970-х «Літаючий цирк Монті Пайтона»
- Філософія - **import this**
- У 2000 році була видана версія 2.0, а в 2008 році - версія 3.0. Python 3 вийшов 27 червня 2008 року
- Актуальна версія Python 3.7

Примітка! Після версії 2.7 відразу слідує версія 3.0

Примітка! Основна команда розробників мови Python припиняє підтримку версії Python 2.x з 1 січня 2020 року. Остання версія мови Python 3.7 вийшла 27 червня 2018 року (див. <https://docs.python.org/3/whatsnew/>).

Переваги (особливості мови Python):

- Мова високого рівня.
- Синтаксис мови Python мінімалістичний і гнучкий, що дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові.
- Підтримує декілька парадигм програмування, зокрема, *структурне, об'єктно-орієнтоване й функціональне*.
- Інтерпретація програм.
- Портативна і платформонезалежна (Linux/Unix/Windows/Mac/Android/iOS, mcu + micropython).
- Динамічна типізація (*тип змінної визначається при призначенні їй змінної*).
- Автоматичне управління пам'яті.
- Багата стандартна бібліотека (os, sys, threading, urllib, json, csv, argparse, hashlib, subprocess, datetime тощо), що встановлюється разом з Python.
- Велика кількість модулів
- Python абсолютно безкоштовний.

Один «недолік»

Швидкість виконання коду.

Де використовується Python?

- розробка сценаріїв для роботи з Web та Internet-програмами, мережеве програмування, засоби підтримки технологій HTML і XML (django, flask, pyramid, bottle тощо);
- програми для роботи з електронною поштою й підтримки Internet-протоколів;
- програми для обслуговування найрізноманітніших баз даних (SQLite, Postgrade тощо);
- програми для роботи з даними, машинного навчання, штучного інтелекту (pandas, keras, pytorch);
- програми для наукових розрахунків (numpy, scipy, matplotlib);
- програми з графічним інтерфейсом GUI (tkinter, PyQy);
- написання інтерфейсу командного рядка – cli (argparse, os, sys, subprocess, json);
- створення ігор і комп'ютерної графіки (pygame, py3d);
- розробка мобільних додатків (kivy) та багато іншого.

Популярні проекти на Python

- *BitTorrent* – протокол для обміну даними.
- *Ubuntu Software Center* – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
- *Blender, Maya* – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
- GIMP – растровий графічний редактор, із підтримкою векторної графіки.
- Вільна енциклопедія *Вікіпедія* .
- Пошукова система *Google* .
- *DropBox* – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
- *Instagram* (python/django).
- *Google/youtube* та засіб для завантаження відео *youtube-dl*.
- *Reddit* (новинний онлайн сервіс).
- *World of Tanks* (веб-портал на python/Django).
- *Cura, Octoprint, Klipper* (3d printing).
- *Ansible, Salt* (devops) – системи віддаленого автоматизованого керування в ОС Linux.
- *Deluge* (torrent client).
- *Yum, Portage* (linux package managers).

Порівняння коду програми на C++, Java і Python

Лістинг 1.1. Програма мовою C++

```
#include <iostream>
using namespace std;
int main() {
    cout<<"Hello world!"<<endl;
    return 0;
}
```

Лістинг 1.2. Програма мовою Java

```
Class MyClass{
    public static void main(String[] arg){
        System.out.println("Hello, world!")
    }
}
```

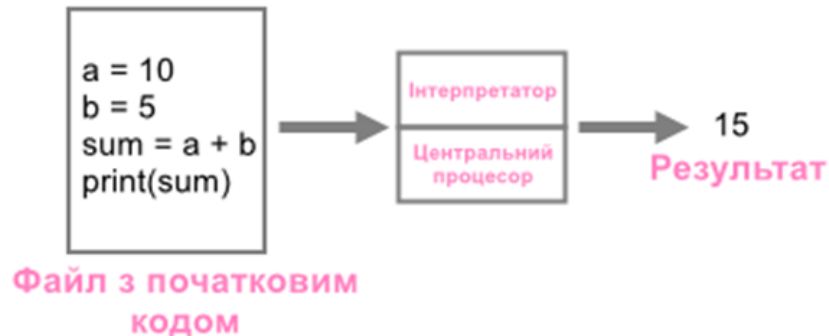
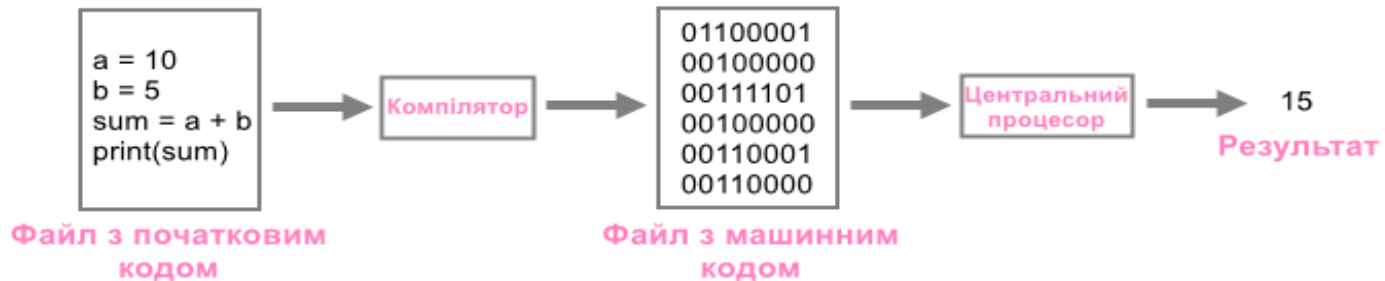
Лістинг 1.3. Програма мовою Python

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

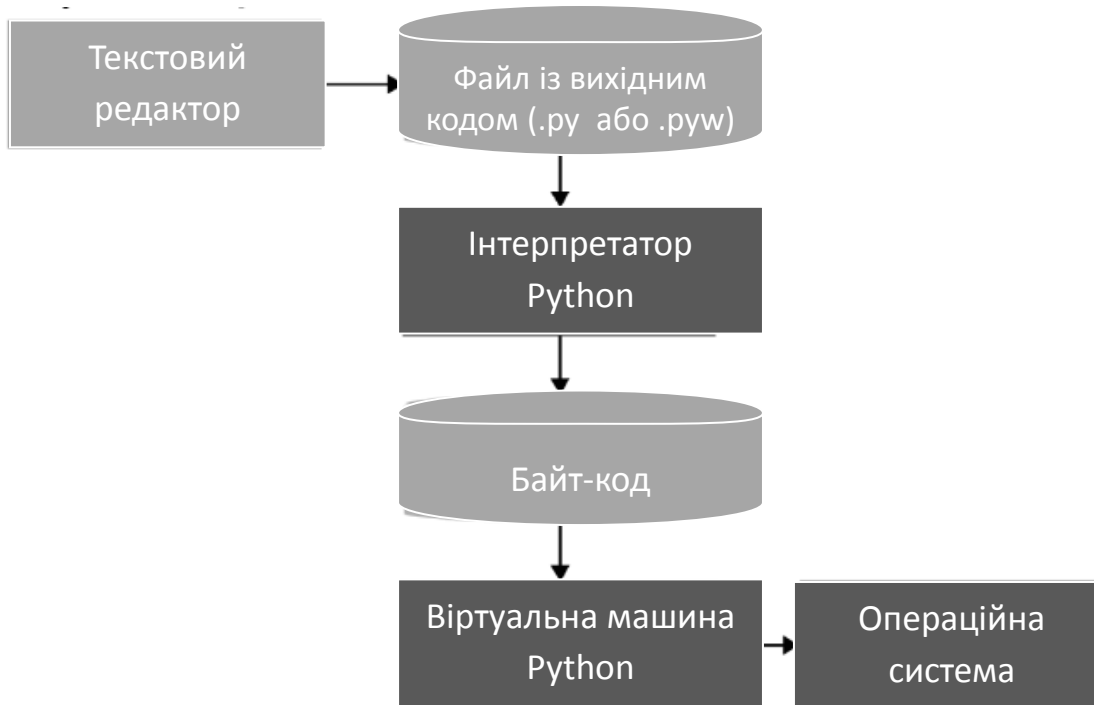

2. Початок роботи в Python

Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування.

Для перекладу мови високого рівня на машинну мову доступні два типи програм: компілятор, інтерпретатор.



Виконання програми в Python



Спочатку пишуть в текстовому редакторі програму (скрипт) із набором виразів на даній мові програмування та зберігають у файлі із розширенням **.py** чи **.pyw** (для програм з графічним інтерфейсом). Передають файл програми на виконання інтерпретатора. Під час першого запуску (або внесення змін у програмний код) створюється проміжний код (байт-код), який записується у файл із розширенням **.pyc**, а потім віртуальна машина переводить отриманий байт-код в набір інструкцій, які виконуються операційною системою.

Програмна реалізація інтерпретатора

1. За замовчуванням використовується інтерпретатор байт-коду написаний на мові C для Python, а саме CPython. CPython компілює вихідний код в байт код, який потім запускається на віртуальній машині CPython. Крім CPython, є ще наступні реалізації Python – Jython, IronPython, PyPy, які реалізовані відповідно на мовах Java, C# и RPython.
2. Запускати підготовлені файли можна в IDLE та в консолі за допомогою команди:
python адреса\ім'я_файлу.py
3. Для створення .exe файлу з коду Python потрібно скористатися зовнішніми модулями, наприклад cxFreeze (<https://cx-freeze.readthedocs.io/en/latest/>, <https://pypi.org/project/cx-Freeze/>)

Середовище програмування для Python

IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.

Notepad++ – безкоштовний текстовий редактор вихідного коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.

PyScripter – інтегроване середовище розробки для мови програмування Python, працює під Windows. Поширюється безкоштовно.

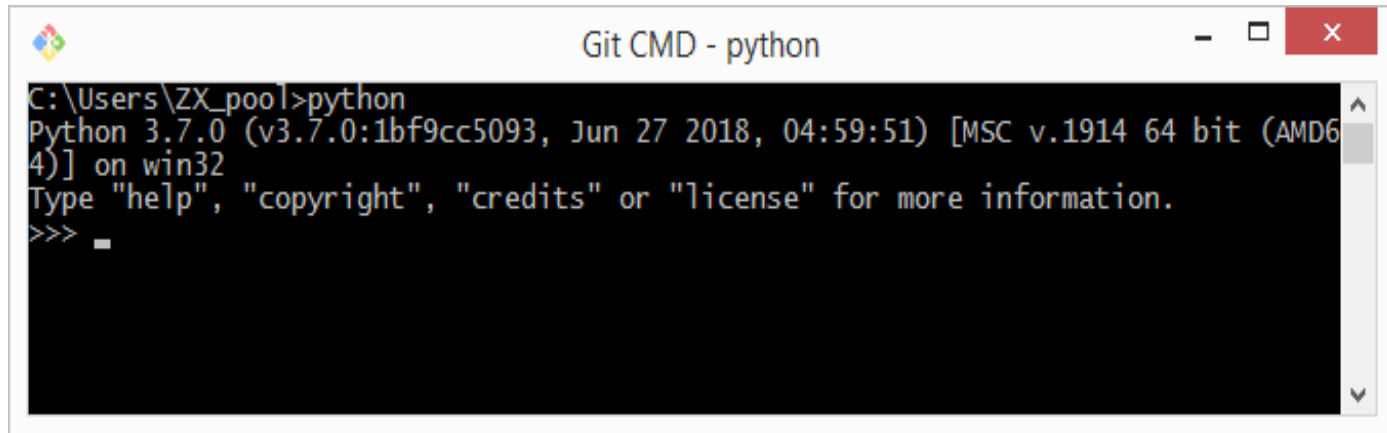
Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і MacOS. Поширюється безкоштовно.

Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, MacOS і Windows.

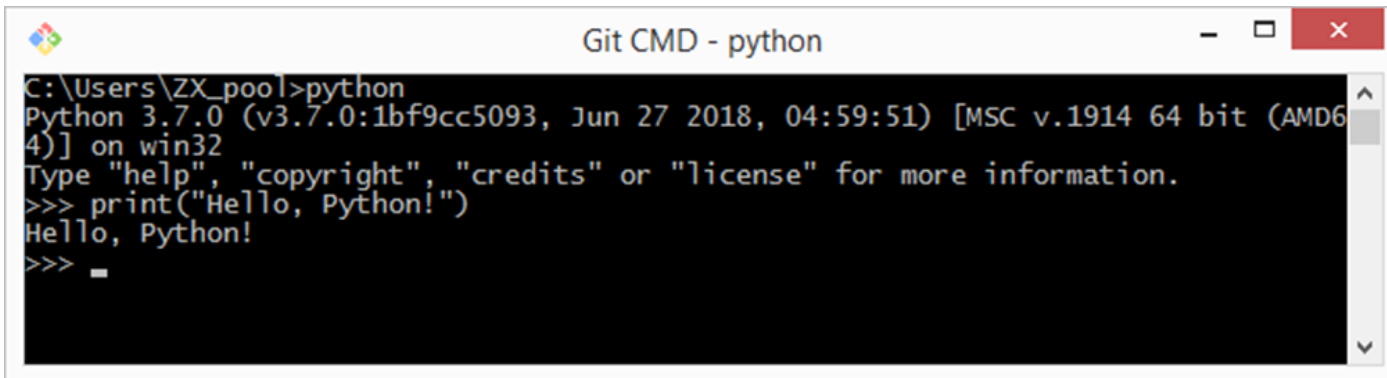
Sublime Text 3 – кросплатформовий текстовий редактор вихідних текстів програм та інтегроване середовище розробки. Підтримує плагіни, розроблені за допомогою мови програмування Python. Sublime Text не є вільним чи відкритим програмним забезпеченням, але деякі його плагіни розповсюджуються з вільною ліцензією, розробляються і підтримуються спільнотою розробників.

PyCharm – інтегроване середовище розробки для мови програмування Python. Підтримує веб-розробку на Django. Присутні безкоштовна версія Community з усіченим набором можливостей і безкоштовна версія Edu для навчальних закладів. PyCharm працює під операційними системами Windows, MacOS і Linux.

Запуск Python: інтерактивний інтерпретатор



```
C:\Users\ZX_pool>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```



```
C:\Users\ZX_pool>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, Python!")
Hello, Python!
>>> _
```

Повідомлення про помилку в режимі інтерпретатора:

```
>>> '19' + 81
Traceback (most recent call last):
  File "<interactive input>", line 1, in <module>
TypeError: Can't convert 'int' object to str implicitly
```

Коментарі:

```
>>> # Привіт, світ!
... print("Hello, world!")
Hello, world!
```

Продовження рядків:

```
>>> alphabet = 'abcdefg' + \
... 'hijklmnop' + \
... 'qrstuv' + \
... 'wxyz'
>>> alphabet
'abcdefghijklmnopqrstuvwxyz'
>>> 1 + 2 + \
... 3
6
```

Стиль мови програмування Python

- Розробники мови є прихильниками певної філософії програмування, яку називають «The Zen of Python» («Дзен Пайтона»). Її текст можна отримати у інтерпретаторі Python за допомогою команди:

>> import this

- У Python для побудови структури програми (виокремлення блоків) використовуються відступи від лівого краю, які створюються за допомогою пропусків (пробілів).
- Для Python визначені рекомендації по стилю написання коду PEP8 .

PEP (Python Enhancement Proposal) – це проектний документ, що надає інформацію спільноті Python або описує нову функцію для Python. PEP повинен забезпечити стислу технічну характеристику ознаки та обґрунтування її

Ключові слова мови Python

Таблиця 1

"Зарезервовані слова Python"

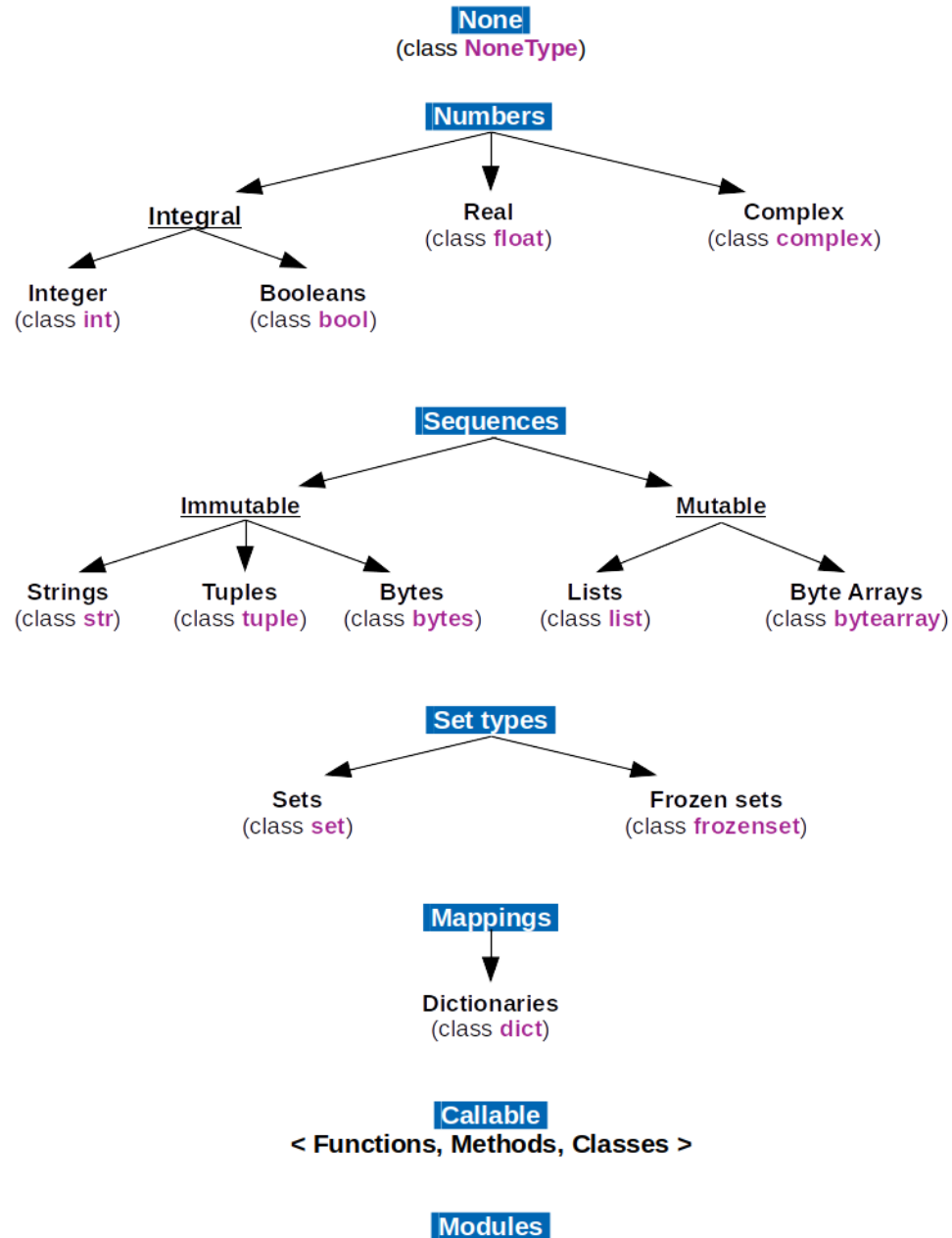
| | | | | |
|--------------|--------------|----------------|-----------|---------------|
| False | class | finally | is | return |
| None | continue | for | lambda | try |
| True | def | from | nonlocal | while |
| and | del | global | not | with |
| as | elif | if | or | yield |
| assert | else | import | pass | break |

Типова структура програми

```
n = int(input())
out = []
for i in range(n):
    ....k = 0
    ....while k < i + 1:
    .....out.append(i+1)
    .....k += 1
    ....if len(out) > n:
    .....break
out = out[0:n]
for i in out:
    ....print(i, end = " ")
```

Python 3

The standard type hierarchy



Базові поняття мови Python

В Python змінні – це просто імена. Присвоювання не копіює значення, воно прикріплює ім'я до об'єкта, який містить дані. Ім'я – це посилання на якийсь об'єкт, а не сам об'єкт. Ім'я можна розглядати як стікер-наклейку.

У математиці символ $=$ означає «дорівнює». У багатьох мовах програмування, включаючи Python, цей символ використовується для позначення «присвоювання».



Функція **id()** повертає ідентифікатор об'єкта (адрес об'єкта в пам'яті комп'ютера).

При зв'язуванні змінної з існуючою, обидві змінні вказують на один об'єкт. Змінна при цьому не копіюється!

Нехай змінна посилається на рядок 'Goodbay!':

```
>>> b='Goodbay!'
```



Коли змінній присвоюється нове значення, попереднє значення втрачається і створюється новий об'єкт (скринька):

```
>>>b='Hello!'
```

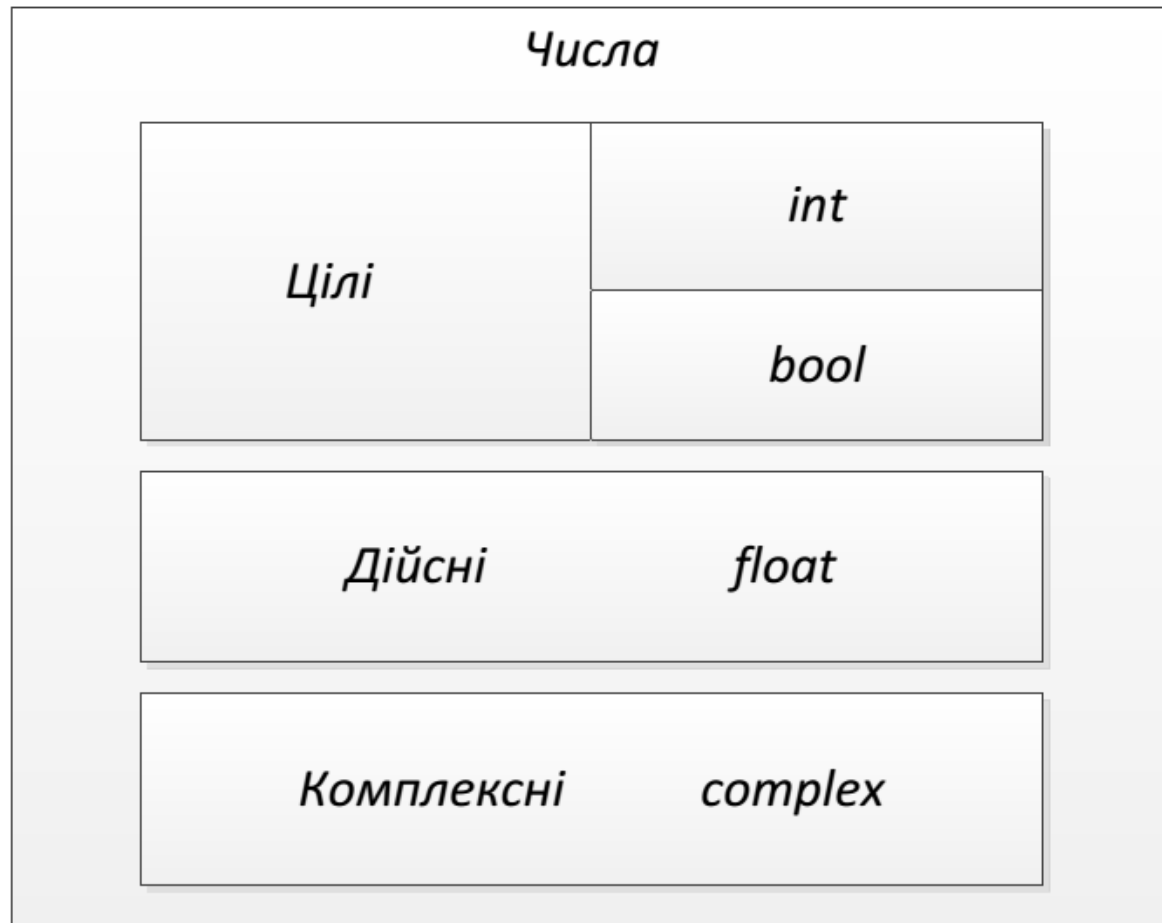
Тепер значення змінної b дорівнює 'Hello!'

Для змінної тип неважливий у Python, типом володіє об'єкт, а не змінна

Для визначення типу значення, на яке посилається змінна, слід використовувати стандартну (вбудовану) функцію **type()**.

Клас – це визначення об'єкта. В Python значення термінів «клас» і «тип» приблизно однакові.

Прості типи даних. Числа



Числа бувають різними: *цілими, дробовими, комплексними.*

Вони можуть мати величезне значення або дуже довгу дробову частину:

- **цілі числа (int)** – додатні і від'ємні цілі числа, а також 0 (наприклад, 4, 687, -45, 0).
- **числа з плаваючою точкою (float)** – дробові числа (наприклад, 1.45, -3.789654, 0.00453). Роздільником цілої і дробової частини служить точка.
- **комплексні числа (complex)** – зберігає пару значень типу float, одне з яких представляє дійсну частину комплексного числа, а інше – уявну (наприклад, $1+2j$, $-5+10j$, $0.44+0.08j$)

Приклад роботи із комплексним числом:

```
>>> x = complex ( 1 , 2 ) # задання комплексного числа – еквівалентно x=complex ('1+2j')
>>> print ( x . conjugate () ) # Комплексно-спряжене число
(1-2j)
>>> print ( x . imag ) # Уявна частина
2.0
>>> print ( x . real ) # Дійсна частина
```

Перетворення типу int в інші типи

Таблиця 2

| int(x) | long(x) | float(x) | complex(x) | bool(x) |
|--------|---------|----------|------------|---------|
| 10 | 10L | 10.0 | (10+0j) | True |
| -25 | -25L | -25.0 | (-25+0j) | True |
| 0 | 0L | 0.0 | 0j | False |

ЗАУВАЖЕННЯ! Тип long прихований в Python 3

Python 2.x

```
>>> type(2147483649)
<type 'int'>
>>> type(-2147483649)
<type 'long'>
```

Python 3.x

```
>>> type(2147483649)
<class 'int'>
>>> type(-2147483649)
<class 'int'>
```


Оператори та вирази в Python

Відомо, що **операція** – це виконання якихось дій над даними (операндами). Для виконання конкретних дій потрібні спеціальні інструменти – операнди.

Основні оператори

Зазвичай виділяють чотири групи операторів:

- арифметичні;
- побітові;
- логічні оператори;
- оператори порівняння.

Арифметичні оператори призначені, насамперед, для виконання арифметичних розрахунків.

Таблиця 3

Арифметичні (математичні) оператори та їх використання

| Оператор | Опис | Прикла д | Результат |
|----------|-----------------------|-------------|-----------|
| $x + y$ | Додавання | $5 + 7$ | 12 |
| $x - y$ | Віднімання | $55 - 10$ | 45 |
| $x * y$ | Множення | $4 * 6$ | 24 |
| x / y | Ділення | $7 / 2$ | 3.5 |
| $x // y$ | Цілочисельне ділення | $7 // 2$ | 3 |
| $x \% y$ | Остача від ділення | $7 \% 3$ | 1 |
| $x ** y$ | Піднесення до степеня | $3 ** 4$ | 81 |
| $-x$ | Зміна знаку | -3 | -3 |

ПРИМІТКИ:

- Допустиме використання скорочених форм операторів: +=, -=, *=, /=, %=, **=.
- В Python дозволене багатократне присвоєння ($x=y=10$) та множинне присвоєння ($a,b=1,2$ ліворуч від оператора присвоєння зазначено одразу декілька змінних)
- Деякі з перерахованих вище операторів можуть застосовуватися не тільки до числових даних, а й наприклад до тексту чи до списків.

Таблиця 4

Побітові оператори та їх використання

| Операція | Результат | Примітки |
|--------------|--|----------|
| $x \mid y$ | побітове або з x і y | |
| $x \wedge y$ | побітове виключаюче або з x і y | |
| $x \& y$ | побітове і з x і y | |
| $x \ll n$ | x зсувається вліво на n біт | (1) (2) |
| $x \gg n$ | x зсувається вправо на n біт | (1) (3) |
| $\sim x$ | біти x інвертуються | |

ПРИМІТКИ:

1. Від'ємні обчислення зсуву є незаконними і викликають [ValueError](#).
2. Зсув вліво на n біт еквівалентно множенню на $\text{row}(2, n)$ без перевірки переповнення.
3. Зсув вправо на n біт еквівалентний діленню на $\text{row}(2, n)$ без перевірки переповнення.

Додаткові методи

int.bit_length () - кількість біт, необхідних для представлення числа в двійковому вигляді, без урахування знака і лідируючих нулів.

int.to_bytes (length, byteorder, *, signed = False) - повертає рядок байтів , що представляють це число.

int.from_bytes (bytes, byteorder, *, signed = False) - повертає число з цього рядка байтів.

Системи численні:

bin (x) - перетворення цілого числа в двійкове число у вигляді рядка (наприклад, 12 → '0b1100').

hex (x) - перетворення цілого числа в шістнадцяткове число у вигляді рядка (наприклад, 12 → '0xC').

oct (x) - перетворення цілого числа в вісімкове число у вигляді рядка (наприклад, 12 → '0o75').

Для задання складених логічних умов використовуються логічні (булеві) операції - **or, and , not**

Таблиця 5

Операції порівняння

| Операція | Значення |
|----------|----------------------------------|
| < | строго менше, ніж |
| <= | менше або рівне |
| > | строго більше, ніж |
| >= | більший або рівний |
| == | рівний |
| != | не рівні |
| is | ідентичність об'єкта |
| is not | заперечення ідентичності об'єкта |

Стандартні функції мови Python (вбудована бібліотека)

| | | | | |
|--------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|---------------------------------------|
| <u>abs()</u> | <u>delattr()</u> | <u>hash()</u> | <u>memoryview()</u> | <u>set()</u> |
| <u>all()</u> | <u>dict()</u> | <u>help()</u> | <u>min()</u> | <u>setattr()</u> |
| <u>any()</u> | <u>dir()</u> | <u>hex()</u> | <u>next()</u> | <u>slice()</u> |
| <u>ascii()</u> | <u>divmod()</u> | <u>id()</u> | <u>object()</u> | <u>sorted()</u> |
| <u>bin()</u> | <u>enumerate()</u> | <u>input()</u> | <u>oct()</u> | <u>staticmethod()</u> |
| <u>bool()</u> | <u>eval()</u> | <u>int()</u> | <u>open()</u> | <u>str()</u> |
| <u>breakpoint()</u> | <u>exec()</u> | <u>isinstance()</u> | <u>ord()</u> | <u>sum()</u> |
| <u>bytearray()</u> | <u>filter()</u> | <u>issubclass()</u> | <u>pow()</u> | <u>super()</u> |
| <u>bytes()</u> | <u>float()</u> | <u>iter()</u> | <u>print()</u> | <u>tuple()</u> |
| <u>callable()</u> | <u>format()</u> | <u>len()</u> | <u>property()</u> | <u>type()</u> |
| <u>chr()</u> | <u>frozenset()</u> | <u>list()</u> | <u>range()</u> | <u>vars()</u> |
| <u>classmethod()</u> | <u>getattr()</u> | <u>locals()</u> | <u>repr()</u> | <u>zip()</u> |
| <u>compile()</u> | <u>globals()</u> | <u>map()</u> | <u>reversed()</u> | <u>__import__()</u> |
| <u>complex()</u> | <u>hasattr()</u> | <u>max()</u> | <u>round()</u> | |

Таблиця 6

Математичні функції і константи (модуль math)

| Виклик функції/константи | Опис/значення |
|--------------------------------|---|
| <code>math.sin(x)</code> | Синус x (представлений у радіанах) |
| <code>math.cos(x)</code> | Косинус x (представлений у радіанах) |
| <code>math.radians(x)</code> | Перетворення x (в градусах) у радіани |
| <code>math.degrees(x)</code> | Перетворення x (в радіанах) у градуси |
| <code>math.exp(x)</code> | Експонентна функція x (e^x) |
| <code>math.sqrt(x)</code> | Квадратний корінь x |
| <code>math.factorial(x)</code> | Факторіал цілого числа x |
| <code>math.e</code> | 2.718281828459045 |
| <code>math.pi</code> | 3.141592653589793 |

Інші функції модуля math:

`acos(x)`, `cosh(x)`, `ldexp(x,y)`, `sqrt(x)`, `asin(x)`, `log(x)`, `tan(x)`, `atan(x)`, `fabs(x)`, `sinh(x)`, `frexp(x)`, `atan2(x,y)`, `floor(x)`, `pow(x,y)`, `modf(x)`, `ceil(x)`, `fmod(x,y)`, `sin(x)`, `log10(x)`, `tanh(x)`