

# Лекція 29

## Налагодження програм в Pycharm



Pycharm надає повний спектр послуг для налагодження початкового (вхідного) коду:

1. Breakpoints в Python.

2. Frames, variables, and watches views в налагоджувачі UI.




3. Runtime – оцінка виразів.

## Основні етапи налагодження

1. Налаштування параметрів налагоджувача. (debugger)
2. Визначення конфігурації запуску/налагодження для програми, яке необхідно налагодити.
3. Створення точок останову (breakpoints) у вхідному (початковому) коді.
4. Запуск сеансу налагодження.
5. Припинення або поновлення сеансу налагодження в міру необхідності.
6. Дії під час сеансу налагоджувача.
7. Режими покрокового виконання програми.

## Запуск сеансу налагодження (дебагера)

Сеанс налагодження починаємо з натискання значка 

1. Натискання  на панелі інструментів запускає сеанс налагодження для поточного вікна у покроковому режимі “Step Through <frame>”
2. Натискання  в меню Run-“Step Through <frame>” діє аналогічно пункту 1.ф
3. Натискання  в меню Python Console включає сеанс налагодження для консолі.

Індикатором активного режиму налагодження є

зміна значка на такий: 

## Debug tool window- вікно налагодження

Вікно Debug tool window містить такі вкладені вкладки:

**Консоль (Console)**: відображає системну інформацію і повідомлення про помилки, а також ввід та вивід даних для програми.

**Налагоджувач (Debugger)** розділений на наступні області:

**Frames** — містить список ниток (**Threads**), що використовуються в програмі.

Область змінних: (**Variables**)

Область стеження: (**Watches**)

## Панель змінних Variables

Панель Variables дозволяє досліджувати значення, що зберігаються в об'єктах програми.

Необхідно:

1. Вибрати фрейм на палі Frames
2. Змінні панелі Variables будуть відображати всі внутрішні дані (параметри методу, локальні і глобальні змінні).

На панелі Variables можливо:

1. Інспектувати об'єкти,
2. Обчислювати вирази,
3. Додати змінні на спостереження і багато іншого.

# Налаштування параметрів програми Debugger

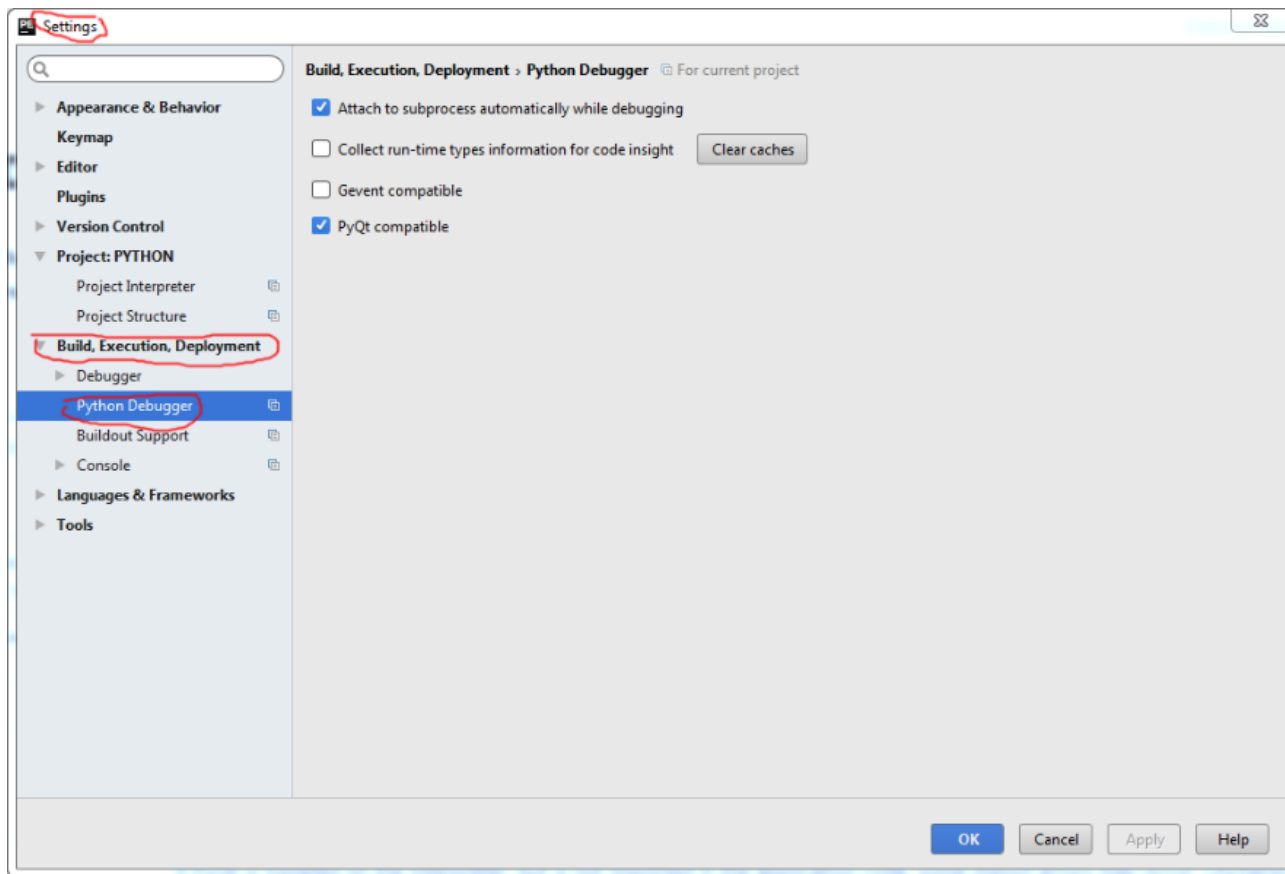
Виберіть у головному меню:

**File** ⇒ **Settings** ⇒ **Build,Execution,Deployment**

“**Build,Execution,Deployment**” вибираємо в колонці ліворуч у вікні “**Settings**”

Розкриваємо “**Build,Execution,Deployment**” і вибираємо пункт “**Python Debugger**”.

У вікні, що відкрилося, налагоджуємо параметри налагоджувача Python:





Опція	Опис
Attach to subprocess automatically while debugging	Якщо обраний цей прапорець, Pycharm автоматично приєднає усі підпроцеси процесу, який налагоджує. Таким чином, якщо батьківський процес має підпроцеси, їхні точки останову будуть завжди працювати.
Collect run-time types information for code insight	Якщо обраний цей прапорець, типи викликів функцій зберігаються під час налагодження, і передаються в Debugger типів. Зверніться до розділу «Використання рядків документації для вказівки типів» « <u>Using Docstrings to Specify Types</u> », якщо потрібні деталі.
Clear caches	Натисніть цю кнопку, щоб вилучити інформацію про типи аргументів, зібраних під час виконання.

# Визначення конфігурації запуску/налагодження для програми, яку необхідно налагодити

Дії тільки для PyCharm Edu

Необхідно розгорнути повне головне меню:  
Виберіть у головному меню:

**File ⇒ Settings ⇒**

**Appearance & Behavior ⇒ Menus and Toolbars ⇒**

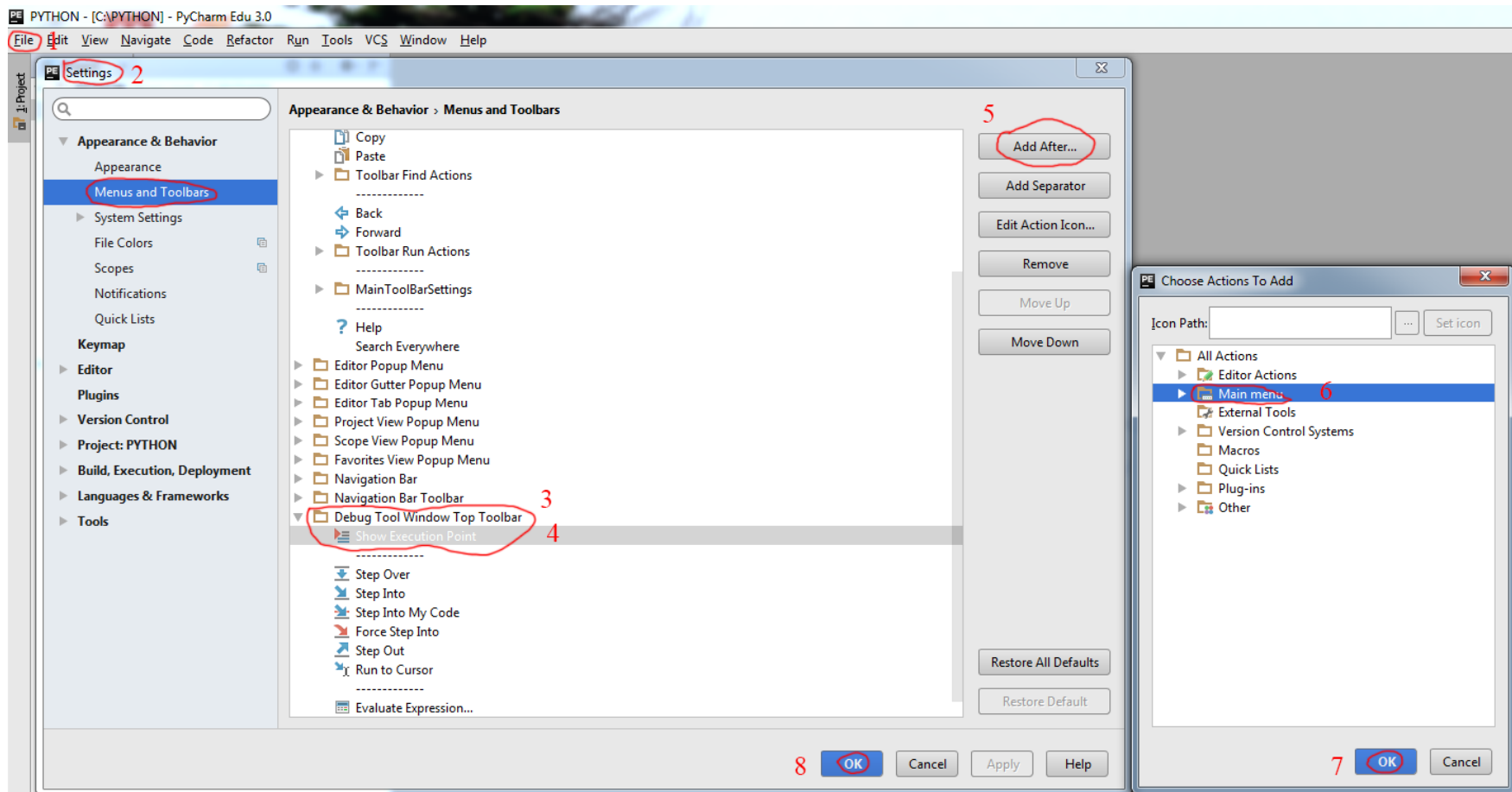
**⇒ Debug Tool Window Top Toolbar**

**⇒ ShowExecutionPoint**


Натиснути кнопку **Add After**

# Вибрати Main Menu.

## Порядок дій зазначений на рисунку.

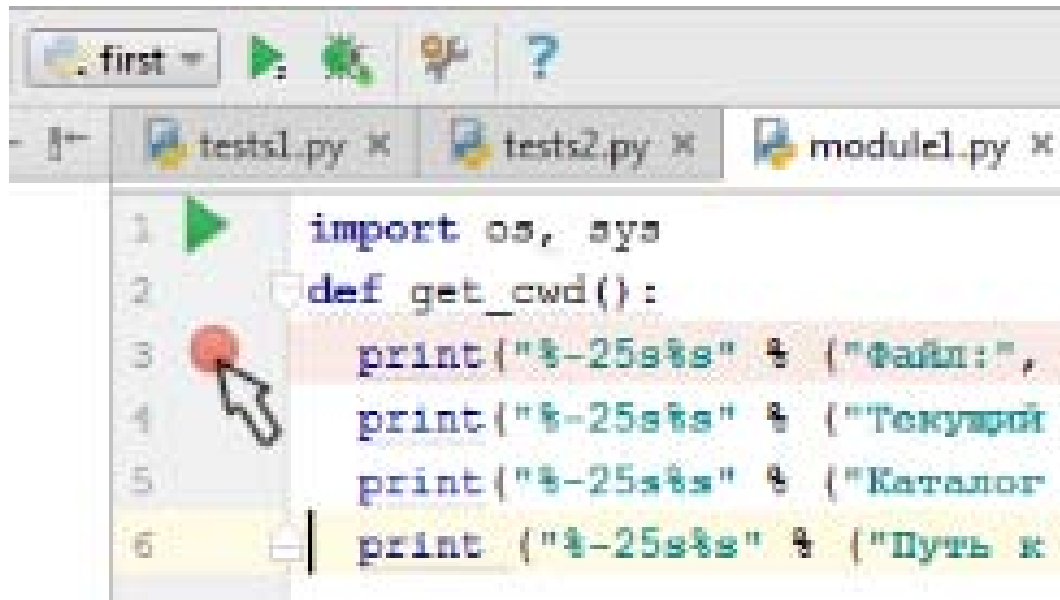


### 3. Створення точок останову (breakpoints) у вхідному коді

Рядок точки останова  є контрольною точкою (breakpoint), призначається на визначений рядок у вхідному коді.

Така точка останову може бути встановлена на рядок, що **виконується**. Коментарі, оголошення й порожні рядки не є припустимими місцями для **точок останову**. Для створення **breakpoint** у редакторі помістіть курсор на потрібний рядок вхідного коду й виконайте одну з наступних дій:

1. Натисніть на ліву зону на лівій смузі проти рядка, де ви бажаєте встановити **breakpoint**



2. Другий спосіб полягає в тому, що в головному меню необхідно вибрати **Run | Toggle Line Breakpoint**.

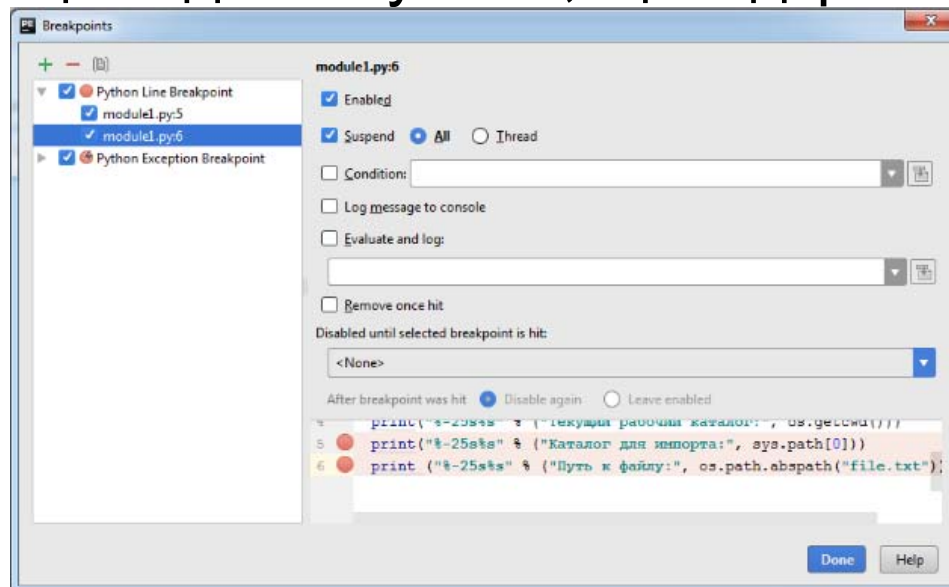
3. Третій спосіб: натиснути комбінацію клавіш **Ctrl + F8**, попередньо активізувавши потрібний рядок тексту.

## Опції точок останову

1. Встановлюємо Breakpoint

2. Задаємо опції для breakpoints: вибрати в головному меню **Run | View Breakpoints...**

Опції задають у вікні, що відкрилося



Елементи керування цієї частини діалогу Breakpoints залежать від типу обраної контрольної точки.

Опції	Опис	Types of breakpoints
Suspend	<p>Установіть цей прапорець, щоб призупинити програму, коли дійде до <b>точки останова</b>. Якщо прапорець не встановлений, програма не буде <b>призупинятися</b>.</p>	Усі типи
Condition	<p>Установіть цей прапорець і вкажіть умову для спрацьовування точки останову в текстовому полі. Умовою є логічний вираз.</p> <p>Цей вираз повинен бути правильним для того рядка, де встановлюється точка останову, і обчислюватися щораз, коли досягається рядок точки останову. Якщо результат оцінки True, обрані користувачем дії виконуються.</p> <p>Якщо результат False, точка останову не виконує ніякого ефекту. Якщо налагоджувач не може оцінити вираз, він виводить повідомлення про помилку оцінки стану. Ви можете вибрати, чи хочете ви зупинитися на цій точці останову або ігнорувати її.</p>	Усі типи

Log message to console	Установіть цей прапорець, якщо ви <b>хочете</b> <b>ввести</b> повідомлення журналу, <b>який</b> буде відображатися в консолі <b>виводу</b> , коли дійде до <b>точки</b> <b>останову</b> .	Усі типи
Log evaluated expression	Установіть цей прапорець, якщо ви <b>хочете</b> обчислити <b>певний вираз</b> в цій контрольній <b>точці</b> й експортувати результат на консоль. Якщо <b>вираз</b> для обчислення є некоректним, то коли дана <b>точка</b> <b>останову</b> буде <b>досягнута</b> , на консоль буде виведене повідомлення про помилку <i>Unable to evaluate expression &lt;your_expression&gt;</i> .	Line breakpoints



## Для видалення точок останову

1. У діалоговому вікні Breakpoints виберіть потрібний рядок з точкою останову, і натисніть на ній лівою кнопкою мишки.
2. У редакторі знайдіть рядок із точкою останову, яку необхідно вилучити, і натисніть на червоний кружок ліворуч.
3. Помістіть курсор на потрібний рядок і натисніть Ctrl+F8.

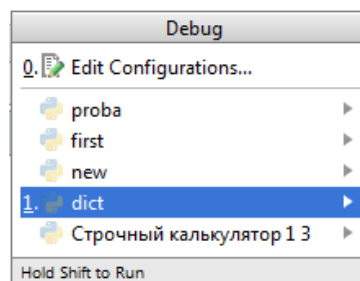
## Запуск сеансу налагодження

Сеанс налагодження починається з обраної конфігурації **Run/debug...**

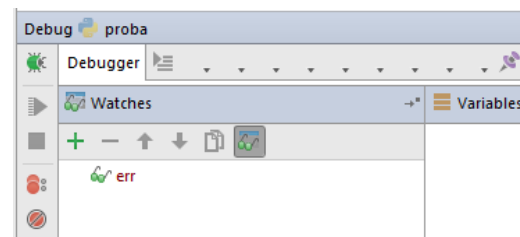
-Далі відкривається меню, у якому можна запуснути редагування конфігурації

**Run/debug.../Edit Configurations...**

-Можна запуснути відкритий фрейм з заданою конфігурацією.



В стані запуску необхідно натиснути кнопку  
“**Show watches in variables tab**”



## Налагодження програми

Для того, щоб почати налагодження скрипта Python:

1. Відкрийте потрібний скрипт Python у редакторі, або виберіть його у вікні інструмента **Project**.
2. У контекстному меню (права кнопка мишки в полі скрипта) або в головному меню **Run** виберіть **Step Through <ім'я сценарію>**:

## Припинення й поновлення сеансу налагоджувача

Коли програма дійде до точки останову, або коли вона поставлена на паузу вручну, робота програми зупиняється.


Для припинення роботи виконайте одну з наступних дій:

1. У головному меню виберіть **Run | Pause Program**

2. Натисніть значок паузи  на панелі інструментів налагодження.

Щоб відновлення роботи, виконайте одну з наступних дій у головному меню:

1. Виберіть **Run | Resume Program**.

2. Натисніть  на панелі інструментів налагодження.

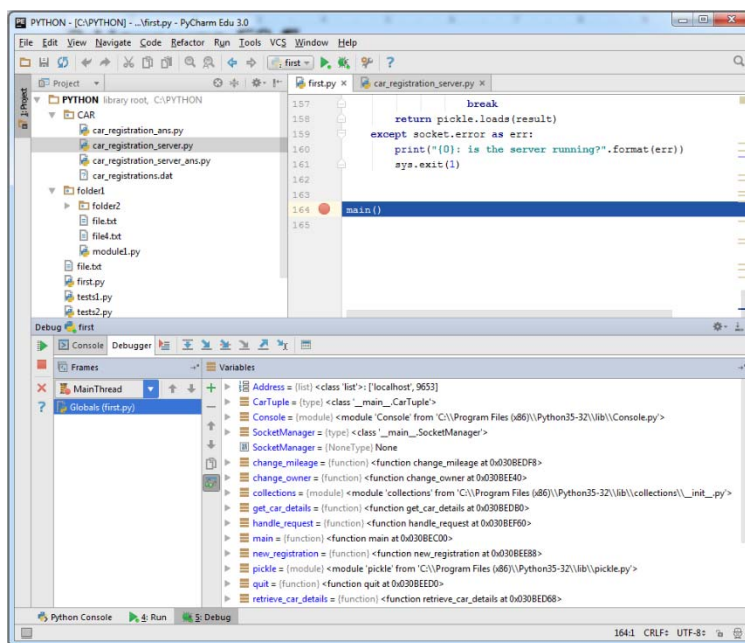
3. Натисніть F9.

## 6. Дії під час сеансу налагоджувача

### 6.1. Моніторинг інформації про налагодження

Інформація про сеанс налагодження відображається у виділених вкладках вікна інструментів налагодження обраної конфігурації Run/debug.

Для кожного сеансу слід використовувати вкладку Console для перегляду повідомлень налагоджувача й виводу даних для з програми.



## *Вивчення припиненої програми*

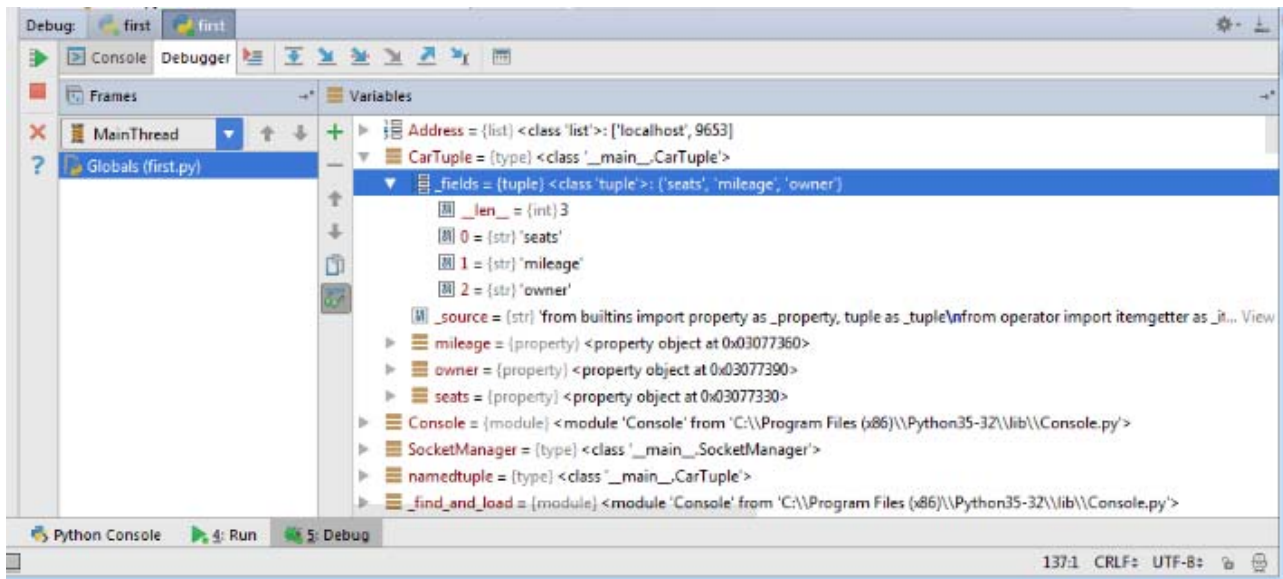
Коли програма призупиняється на точці останову, або виконання програми припинене вручну її можна перевірити шляхом аналізу кадрів (frames).

Фрейм зберігає локальні змінні викликуваного методу або функції, аргументи до нього, і контекст коду, який дозволяє виконати вираз.

Усі поточні активні фрейми відображаються на панелі **Frames** вікна інструментів налагодження, де ви можна перемикає їх й аналізувати інформацію, що зберігається в них.

1. Виберіть нитку зі списку, що розкривається, на верхній частині панелі Frames.

2. Виберіть фрейм із списку **Frames**. Панель **Variables** показує всі змінні, доступні для виклику методу в цьому фреймі, так що ви можете додатково вивчити їх:



Коли фрейм обраний у списку, усі значення, доступні для виклику методу цього фрейму відображаються в **Variables** панелі вікна інструментів налагодження **Debug**, так що ви можете додатково вивчити їх. Опишемо способи спрощення перегляду цих значень.

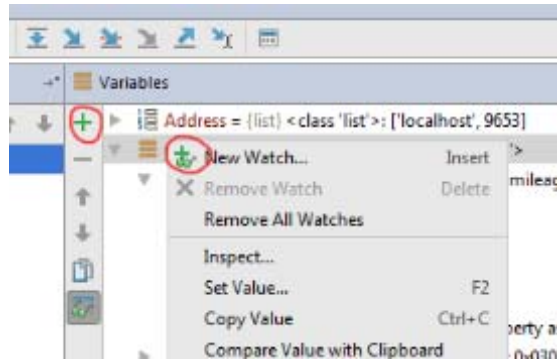
## Додавання, редагування й видалення *watches*

Якщо необхідно обчислити декілька змінних або виразів у контексті поточного фрейму, а також переглядати їх усі одночасно створемо для них *watches*. Значення виразів обновляються з кожним кроком по додаткові, але їх видно тільки коли додаток припинений. Ці вирази зберігаються як частина вашого проекту.

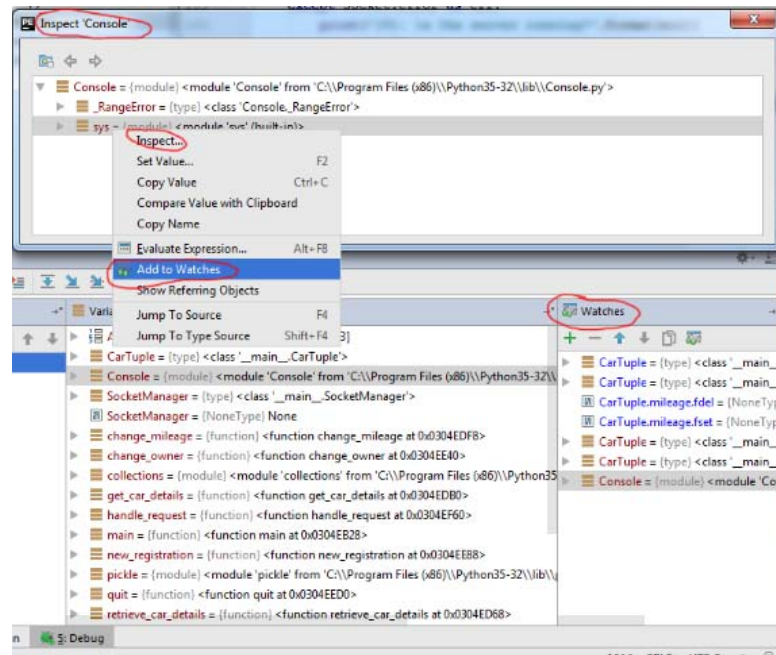
Додавання watch:

1. На панелі [Watches](#) натисніть кнопку +,
2. Виберіть пункт New Watch у контекстному меню
3. Просто натисніть кнопку **Insert** на клавіатурі.





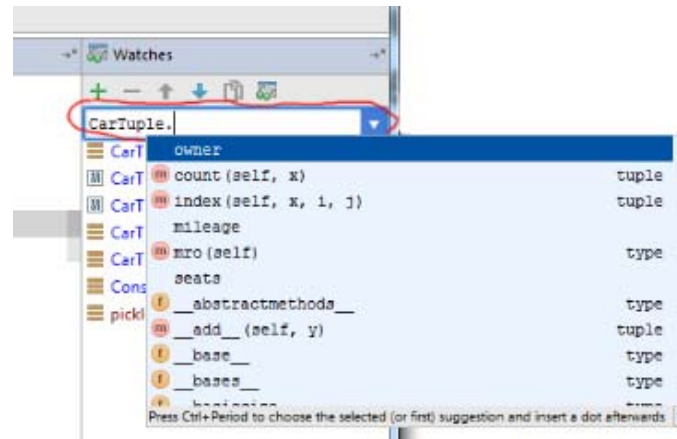
4. Вибрати контекстному меню елемент **Add to Watches:**
- на панелі **Variables**,
  - у вікні **Inspect** ,
  - удіалоговому вікні **Evaluate Expression**.



5. Виберіть потрібний елемент у панелі змінних і перетягніть його на панель **Watches**.

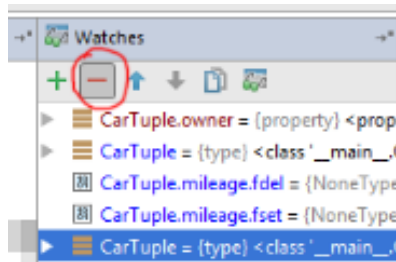
## Редагування watch

Для редагування **watch**, тобто зміни виразу, представленого за допомогою watch, клікніть правою кнопкою миші на бажаному watch і виберіть **Edit** у контекстному меню.



## Видалення watch

1. Щоб видалити watch на панелі **Watches**, виберіть watch, який бажаєте вилучити, й натисніть «-».



2. Можна у контекстному меню вибрати команду **“Remove Watch”**
3. Можна натиснути **Alt + Delete**.

## Обчислення виразів (*Evaluating Expressions*)

Рyсharm дозволяє обчислювати вирази й фрагменти коду в контексті стека фрейму, обраного в панелі Frames вікна інструментів налагодження ([Debug tool window](#)).

Передбачені наступні режими обчислення:

1. **Expression Mode** для обчислення однорядкових виразів.
2. **Code Fragment Mode** для обчислення коротких ділянок коду. Ми можемо обчислювати присвоєння, цикли та `if/else`.

Крім того, Рyсharm дає можливість швидко обчислити вираз в редакторі при наведенні курсора або при виборі рядка. (Наводимо на код, значення спливає, натискаємо+)

## Обмеження Expression Evaluation


При використанні **Expression Evaluation**, слід звернути увагу на наступне:

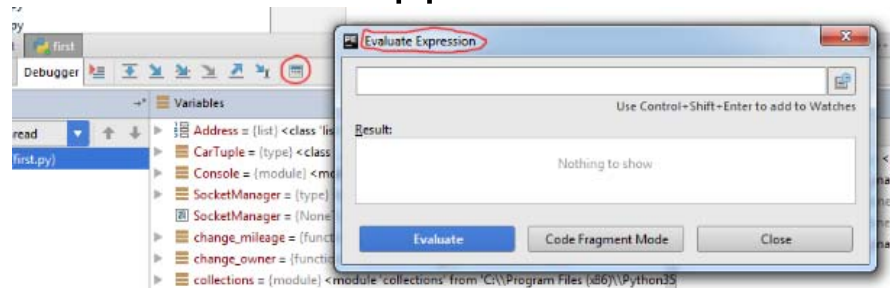
1. Метод може бути викликаний у діалоговому вікні Expression Evaluation **тільки якщо налагоджувач зупинився на контрольній точці**, а не був припинений на паузу.

2. Обчислення виразів може бути тільки "однорівневим". Інакше кажучи, якщо Pucharm зупиняється в точці останову в методі, викликаному з Expression Evaluation, ви не можете використовувати знову Expression Evaluation.

## Виклик панелі Evaluate Expression

Виклик команди Evaluate Expression здійснимо одним з наступних способів:

1. У головному меню виберемо **Run | Evaluate Expression**.
2. У контекстному меню редактора виберемо **Evaluate Expression**
3. Натискаємо **Alt+F8**.
4. Натискаємо  на Stepping-панелі інструментів у вікні інструмента налагодження.

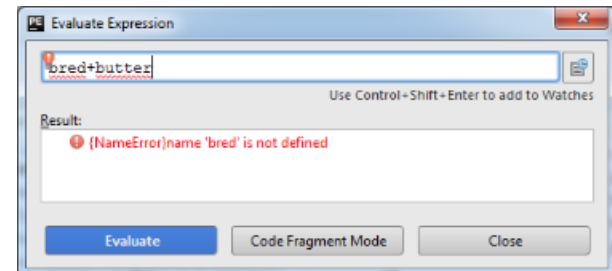
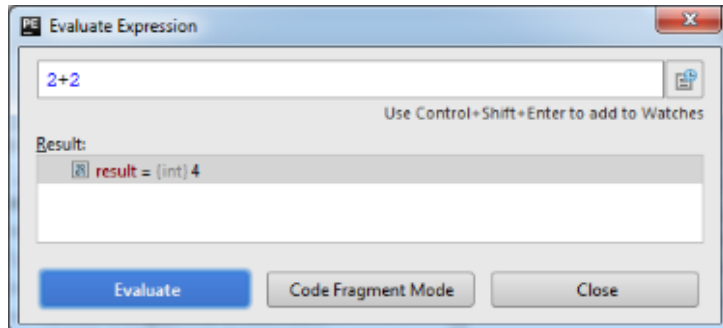


## Обчислення в панелі Evaluate Expression

### 1. Виберемо режим обчислення.

Якщо ми бажаємо обчислити фрагмент коду, натискаємо кнопку **Mode Code Fragment**.

2. Залежно від обраного режиму, введіть вираз або оператор у текстове поле й натисніть кнопку **Evaluate**.

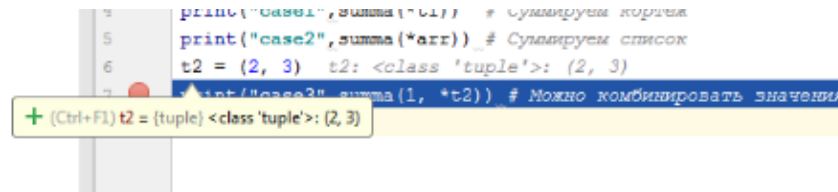


Якщо зазначений вираз неможливо обчислити, ймовірна причина буде коротко описана в панелі результатів діалогового вікна.

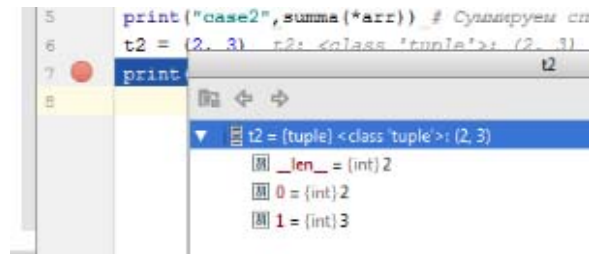


## Обчислення виразів у редакторі

Під час сеансу налагоджувача, значення будь-якого виразу відображається в підказці щоразу, коли ви наведете курсор миші над ним.



Якщо вираз **містить дітей**, натиснувши кнопку **+**, розширюємо вузол і відображаємо всіх дітей.

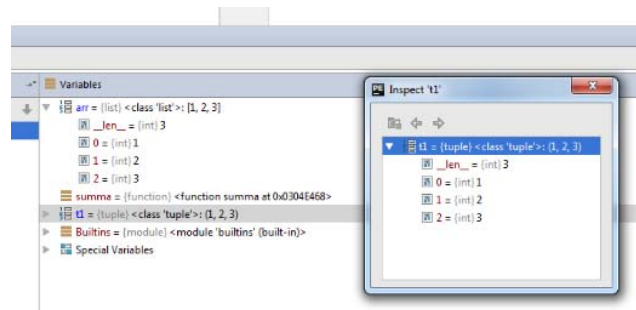


## Інспектування спостережуваних елементів

Pycharm допомагає перевіряти будь-які змінні або **watches** елементи у своєму власному вікні. Наприклад, якщо вам необхідно вивчити кілька посилань у деталях, ви можете відкрити оглядове вікно для кожного з них. Діючи таким чином, окреме вікно створюють для кожної змінної або watch посилання й усі їх дочірні посилання.

Оглядові вікна не модальні, і ви можете запустити стільки вікон, скільки потрібно. Усі зміни посилань негайно відбиваються у відповідних оглядових вікнах.

Для інспектування посилання виберіть елемент на панелі **Variables** або **Watches**. Інший варіант – вибрати **Inspect** у контекстному меню.

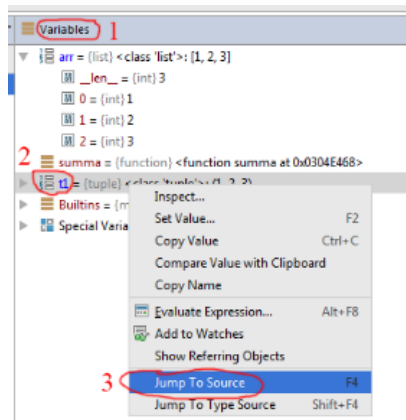


## Перехід з Debug Tool Window до вхідного коду

Для переходу до вхідного коду з вікна інструментів налагодження необхідно зробити наступне:

1. Вибрати бажаний елемент у закладці **Variables** і натиснути F4.

2. Клікнути елемент правою кнопкою мишки в закладці **Variables**, а потім вибрати **Jump to Source** з контекстного меню.



## Знаходження точки поточного виконання


Коли програма припинена, вхідний файл, пов'язаний з поточною точкою виконання, відкривається в редакторі. Точка поточного виконання (наступний рядок, який буде виконаний) відзначений синьою лінією.

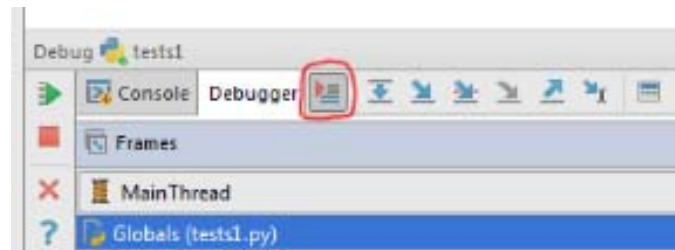
Ви можете відвідати інші файли, а потім повернутися до поточної точки виконання за допомогою дій, описаних у даному підрозділі.

Існує кілька варіантів пошуку поточної точки виконання:

1. У головному меню виберіть **Run | Show Execution Point**.

2. Натисніть **Alt + F10**.

3. Натисніть  на «[stepping](#)» панелі вікна інструментів налагодження.



## Покрокове виконання програми

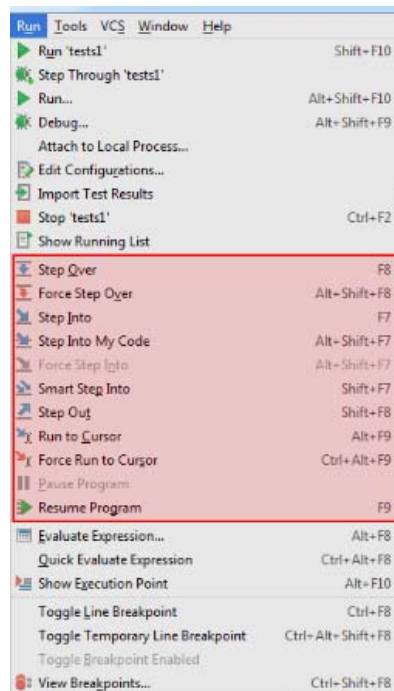
Коли досягається точка останову або ваша програма припинена, вікно інструментів налагодження стає активним і дозволяє одержати контроль над виконанням програми. Для цього можна використовувати команди меню **Run**, або іконки на панелі інструментів степпинга у вікні інструмента налагодження ([Debug tool window](#)).

Кожна дія [степпинга](#) просуває точку виконання до наступного місця виконання, залежно від дії, яку ви вибираєте.

# СТЕПИНГ

Можна виконати одну з наступних дій:

1. У головному меню вибрати **Run | <stepping command>**



2. Скористатися комбінаціями клавіш на клавіатурі, які зазначені праворуч від пунктів меню, розглянутого вище.
3. Використовувати кнопки «[stepping](#)» панелі вікна інструментів налагодження.

