

Data Mining com Python

Victor Machado da Silva, MSc
victor.silva@professores.ibmec.edu.br

Índice

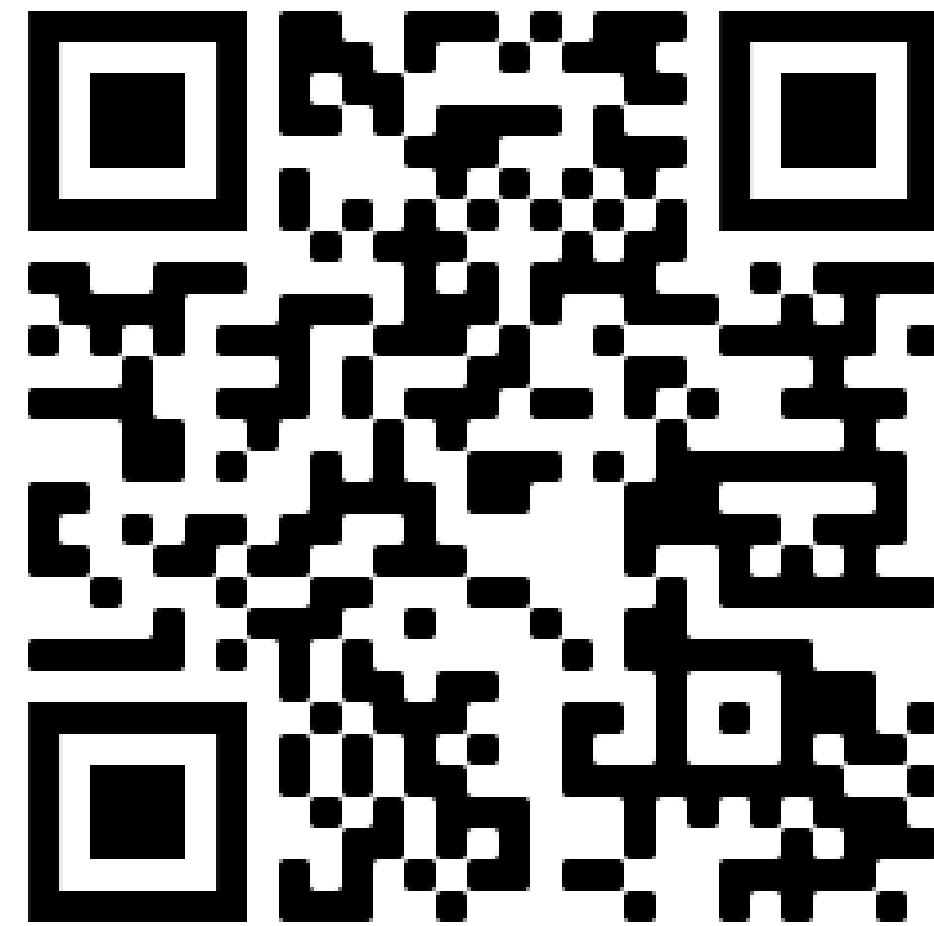
- [Apresentação do curso](#)
- [Configurando o ambiente Python](#)
- [Instalando e configurando IDEs](#)
- [Utilizando o PyCharm](#)
- [Utilizando o VSCode](#)
- [Configurando o pylint](#)
- [Instalando os pacotes via PyPI](#)
- [Por que Python?](#)



Apresentação do curso

Apresentação do curso

- Contato: victor.silva@professores.ibmec.edu.br
- Aulas às sextas-feiras, de 18:30 às 22:50
- Grupo no Whatsapp: <https://chat.whatsapp.com/C4Rx0AUhIfJ1WF45fwjF0P>
- Material no GitHub: <https://github.com/victor0machado/2020.2-datamining>



Apresentação do curso

Sem.	Data	Tópico
01	14/08	Introdução à disciplina / Python: configurando o ambiente e primeiras aplicações
02	21/08	Python: tipos de dados, operadores e funções
03	28/08	Python: estruturas condicionais e de repetição
04	04/09	Python: listas e dicionários, pacotes externos e ambientes virtuais
05	11/09	Mineração de Dados: objetivos, organização, modelagem e ferramentas
06	18/09	Extração de dados via API e web scraping com requests e json
07	25/09	Uso do Jupyter/Google Colab e manipulação de dados com Pandas
08	02/10	P1
09	09/10	Visualização de dados com Matplotlib e Seaborn
10	16/10	Análise de dados – k-vizinhos próximos e Naive Bayes
11	23/10	Análise de dados – Regressão linear simples, múltipla e logística
12	30/10	Análise de dados – Árvores de decisão
13	06/11	Análise de dados – Agrupamento
14	13/11	Redes Neurais
15	20/11	SEM AULA (CONSCIÊNCIA NEGRA)
16	27/11	P2
17	04/12	SEM AULA
18	11/12	PS
19	18/12	SEM AULA

Apresentação do curso

Avaliação

- Proporção:
 - Exercícios periódicos (AC): 20%
 - Projeto (AP1): 40%
 - Projeto (AP2): 40%
- Detalhes das entregas:
 - Exercícios da AC são individuais
 - Projetos de AP1 e AP2 em grupos de no mínimo 2 e no máximo 3 pessoas
 - Entrega via Integrees
- AS será uma prova com consulta, que substituirá a menor nota entre AP1 e AP2.

Sugestões de materiais para estudo



- Python é uma linguagem intuitiva para o aprendizado, porém é importante termos à mão livros, apostilas e outros materiais para auxiliar os estudos. Abaixo encontram-se algumas sugestões:
 - Documentação oficial em Python: <https://docs.python.org/pt-br/3/index.html>
 - Joel Grus - *Data Science do Zero: Primeiras regras com o Python* (Alta Books)
 - Raul S. Wazlawick - *Introdução a Algoritmos e Programação com Python* (Elsevier)
 - Sérgio Luiz Banin - *Python 3 - Conceitos e Aplicações - Uma abordagem didática* (disponível no Minha Biblioteca!)
 - Stack Overflow: <https://stackoverflow.com/questions/tagged/python>
 - Artigos no Medium.com: <https://medium.com/search?q=python>
 - Leandro A. da Silva et al – *Introdução à Mineração de Dados – Com aplicações em R* (Elsevier)



Sugestões de materiais para estudo

- Canais interessantes no Youtube sobre Python e programação:
 - Programação Dinâmica: <https://www.youtube.com/c/ProgramacaoDinamica/>
 - Curso em Vídeo: <https://www.youtube.com/c/CursoemVideo/>
 - Sentdex (em inglês): <https://www.youtube.com/c/sentdex>
 - Filipe Deschamps: <https://www.youtube.com/c/FilipeDeschamps>
 - DevMedia: <https://www.youtube.com/c/DevmediaBrasil>
 - FreeCodeCamp.org: <https://www.youtube.com/c/freeCodeCamp>



Algumas dicas



- Uma disciplina de programação exige alguns cuidados para facilitar o aprendizado...
 - Tente acompanhar simultaneamente o que está sendo implementado
 - Avise se não estiver conseguindo acompanhar!
 - Resolva exercício! Programação não se aprende só assistindo ou lendo



 Victor Silva (RJ) 



 Ausente  Sair da sessão

Comentários

 Satisfeito  Insatisfeito

 Surpreso  Confuso

 Mais rápido  Mais devagar

 Concordar  Discordar

Conhecendo um pouco melhor...





Por que Python?

Por que Python?

- Python foi concebido no final da década de 1980, por Guido van Rossum, na Holanda
- A linguagem foi batizada em homenagem ao programa de TV britânico *Monty Python's Flying Circus*, que fazia muito sucesso na época em boa parte do mundo
- A linguagem foi desenvolvida com o objetivo de ser simples de se desenvolver, e com uma estrutura semântica e sintática intuitiva e natural

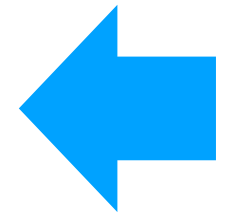


Por que Python?

- Características do Python:
 - Linguagem de propósito geral
 - Fácil e intuitiva
 - Multiplataforma
 - É possível começar a programar de forma simples, sem instalar inúmeros pacotes
 - Livre
 - Organizada
 - Orientada a objetos
 - Inúmeras bibliotecas à disposição
 - Extensa comunidade, com vasta documentação online

Por que Python?

- Principais áreas de atuação:
 - Inteligência Artificial
 - Biotecnologia
 - Computação 3D
 - Ciência de Dados
 - Internet das Coisas



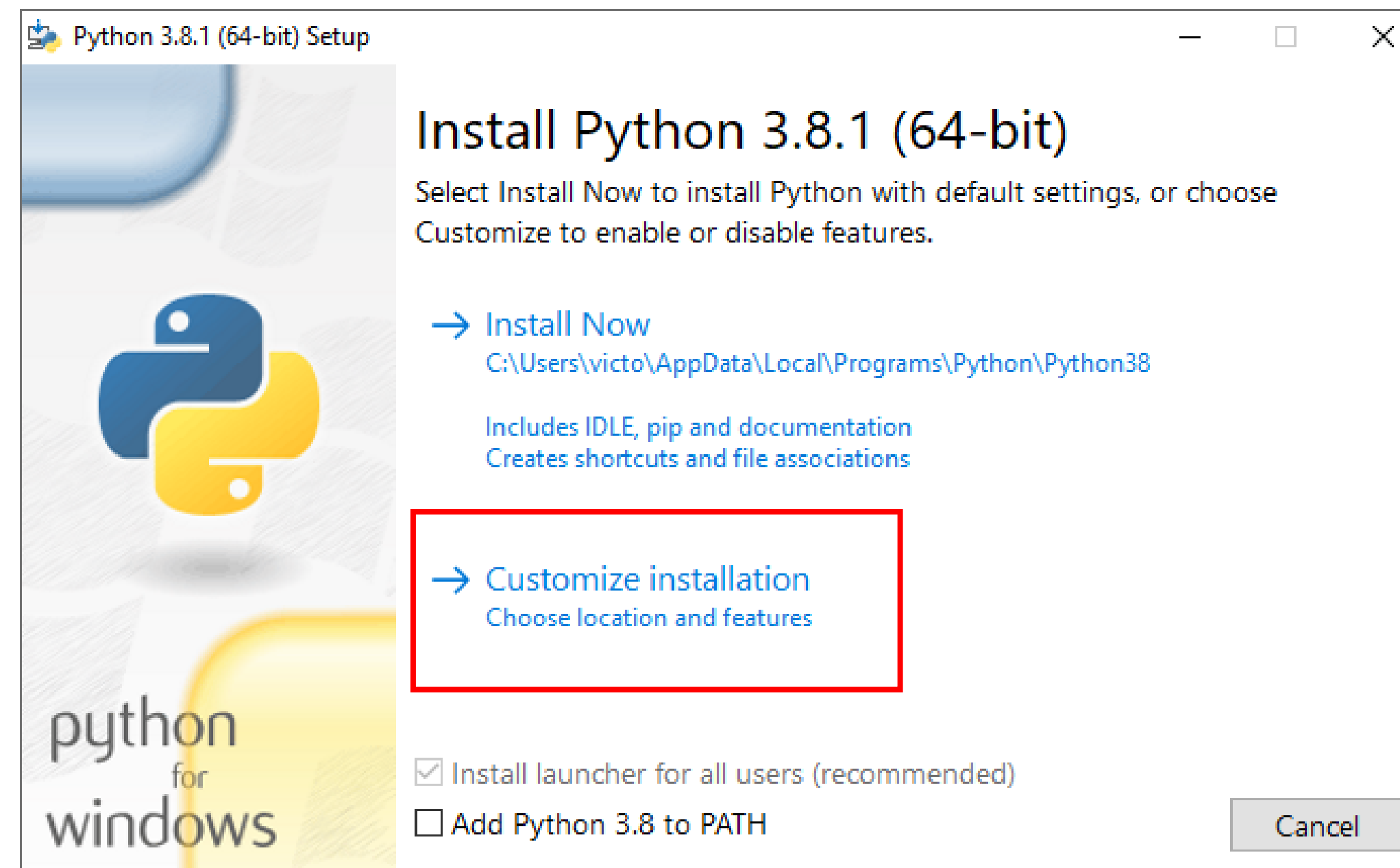
Configurando o ambiente Python

O que é Python?

- Python é uma linguagem de programação de alto nível, lançada por Guido van Rossum em 1991. Atualmente é uma das linguagens de uso mais abrangentes no mundo todo, principalmente nas áreas de Data Science e em aplicações de *back-end*, ou seja, de processamento de dados que não interagem diretamente com o usuário final.
- Diversas organizações utilizam Python atualmente:
 - Google
 - Yahoo!
 - NASA
 - AirCanada

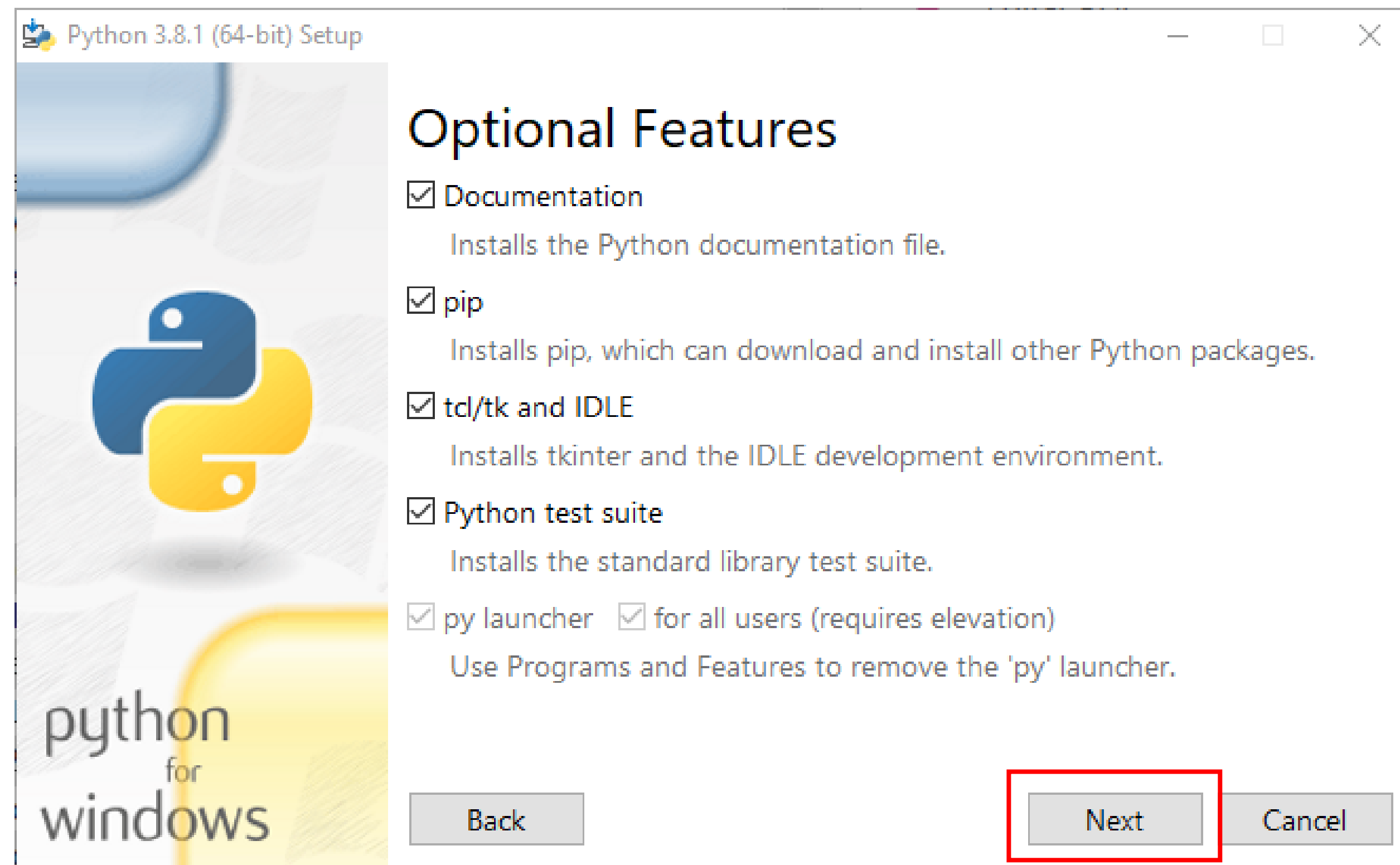
Como instalar Python?

- Neste curso podemos trabalhar com qualquer versão recente do Python, 3.7 ou superior. Para baixar o instalador para Windows, clique [neste link](#). O download do instalador para macOS se encontra [neste link](#).
- Este curso focará no uso do Python para Windows. Para o uso no macOS, veja no [site oficial da linguagem](#) informações particulares.
- Ao clicar no instalador, selecione a opção “Customize installation”.



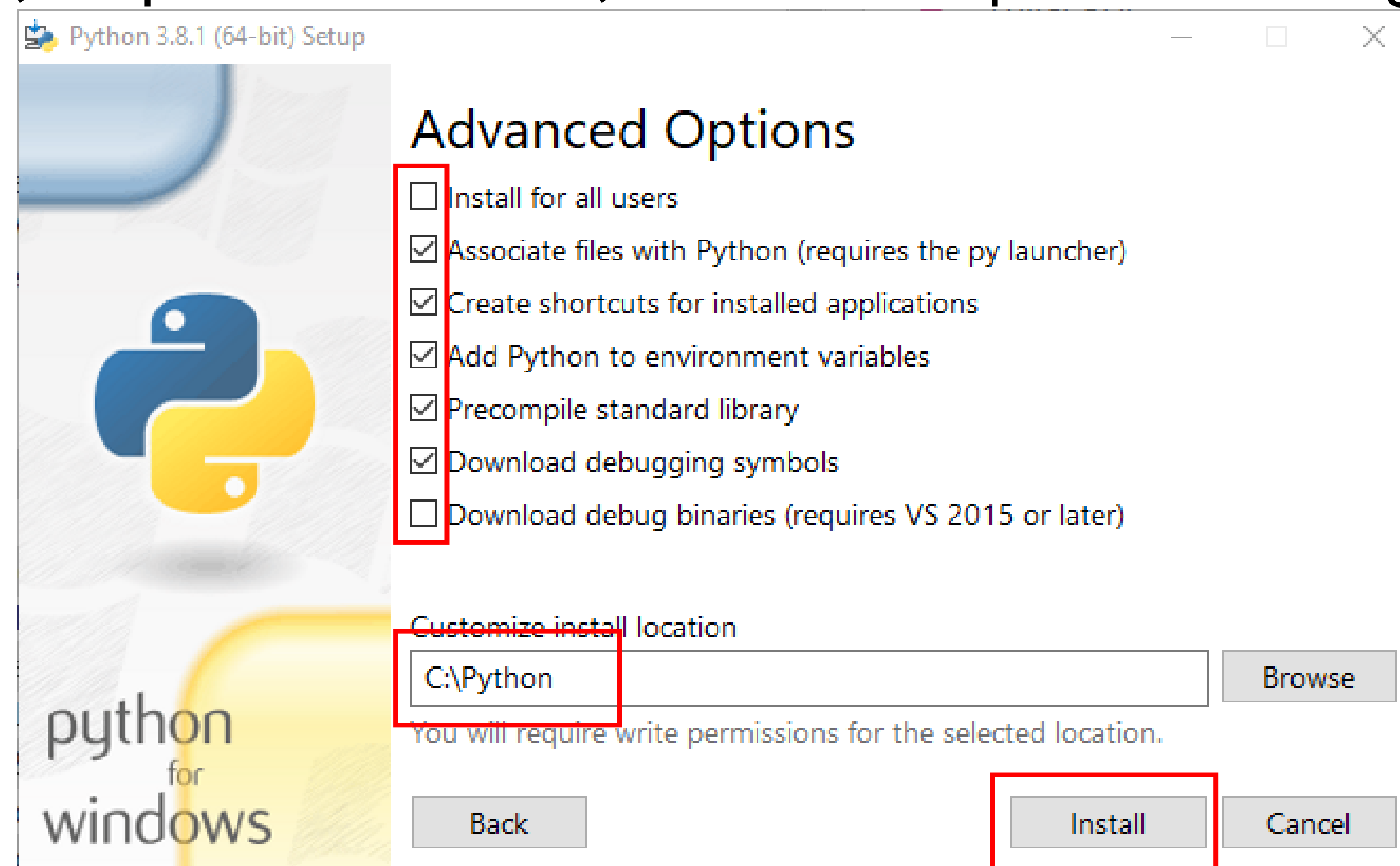
Como instalar Python?

- Na tela “Optional Features”, clique em “Next”.




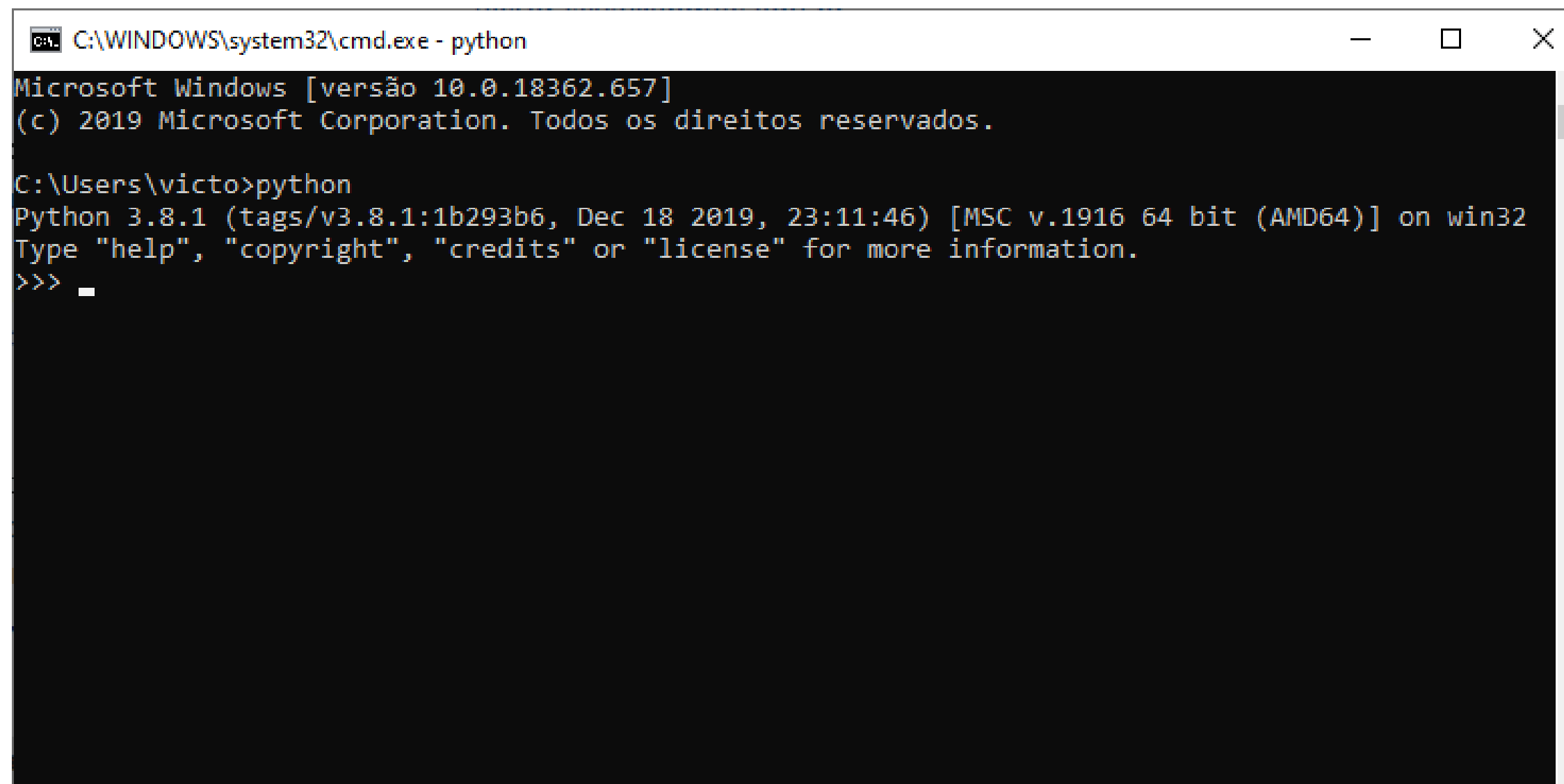
Como instalar Python?

- Na tela “Advanced Options”, deixe as caixas de opções marcadas conforme a imagem abaixo.
- No campo “Customize install location”, escolha um caminho de fácil acesso. O caminho sugerido é “C:\Python”.
- Com tudo pronto, clique em “Install”, e conclua após a mensagem de sucesso.



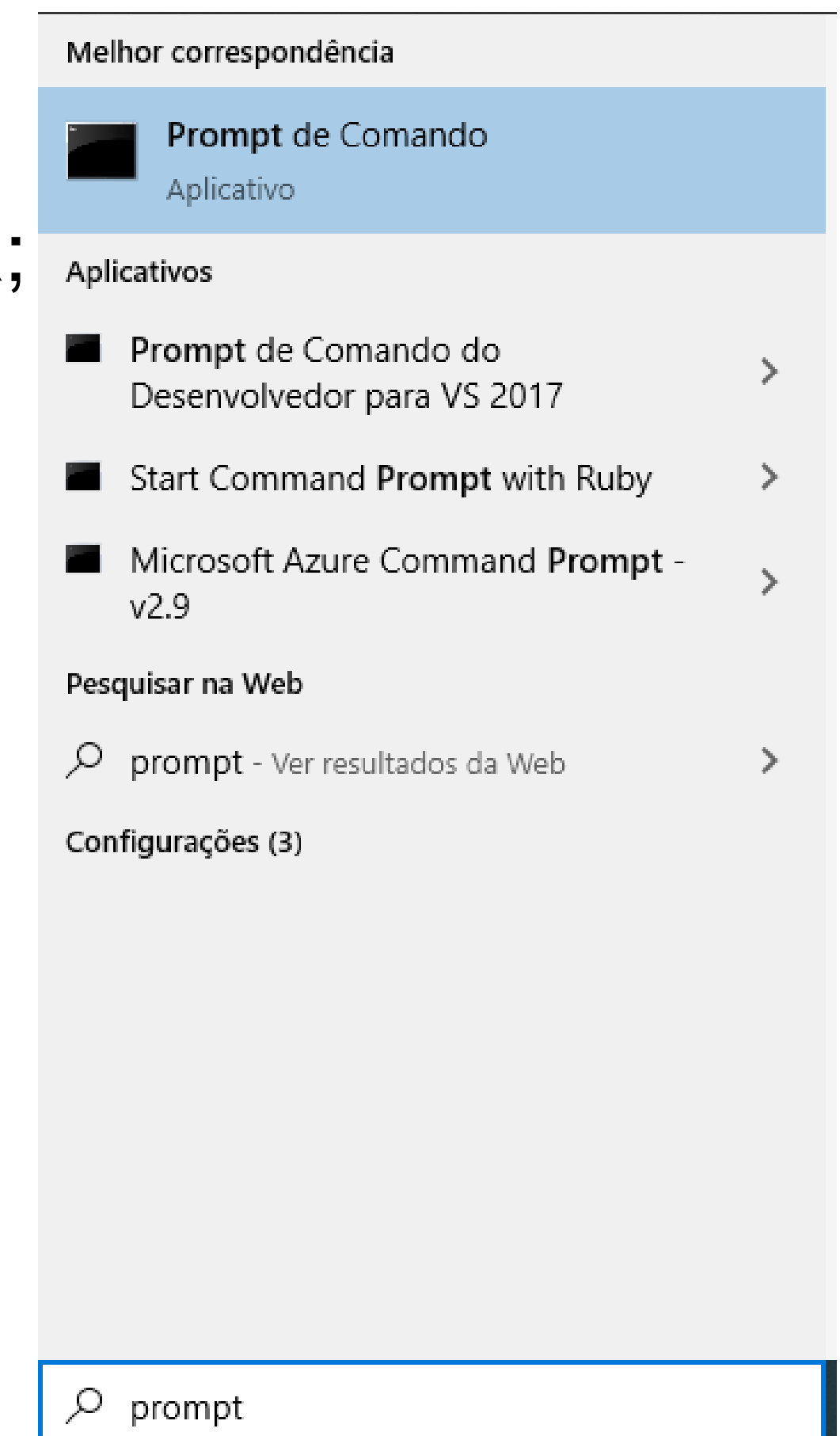
Como instalar Python?

- Para conferir se a instalação foi bem sucedida, faça os seguintes passos:
 - Clique no botão do Windows ;
 - Digite a caixa de pesquisa **prompt de comando**, e abra o programa;
 - Na janela que abrir, digite o comando **python** e pressione Enter;
 - O Windows deve inicializar um editor de Python na mesma janela, como mostrado abaixo.



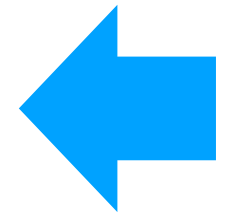
```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [versão 10.0.18362.657]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\victo>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```



Algumas dicas iniciais após instalar o Python

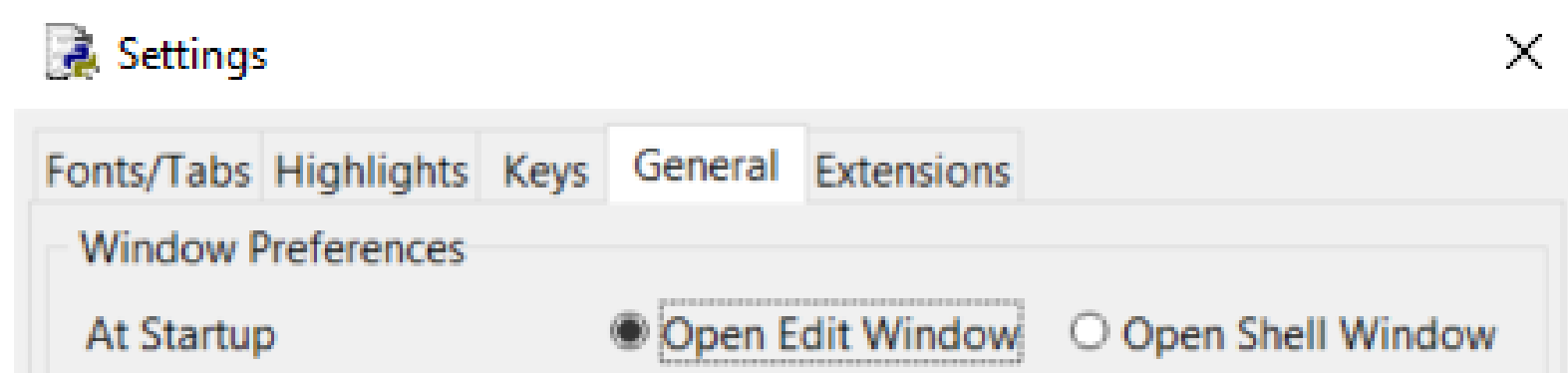
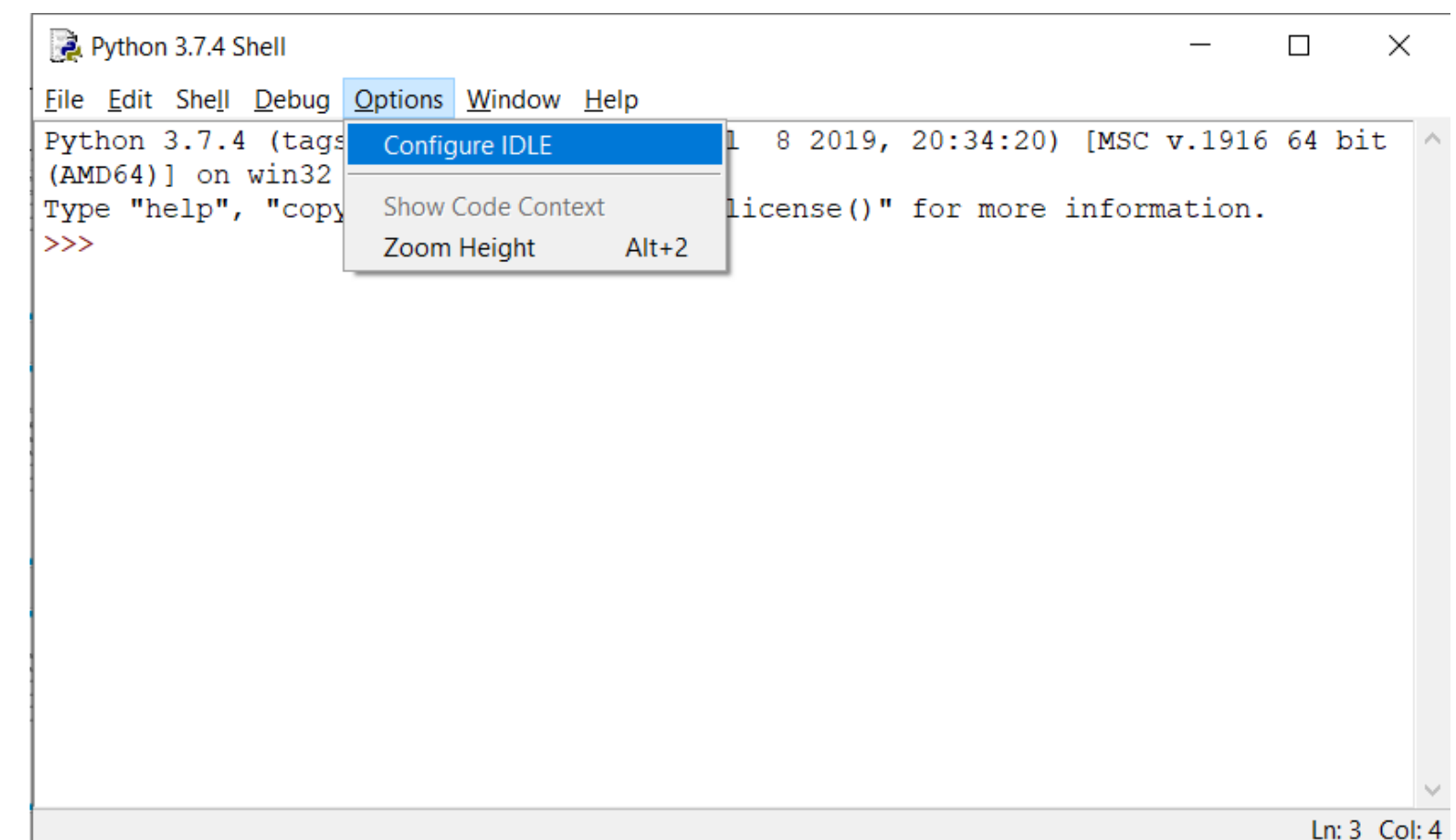
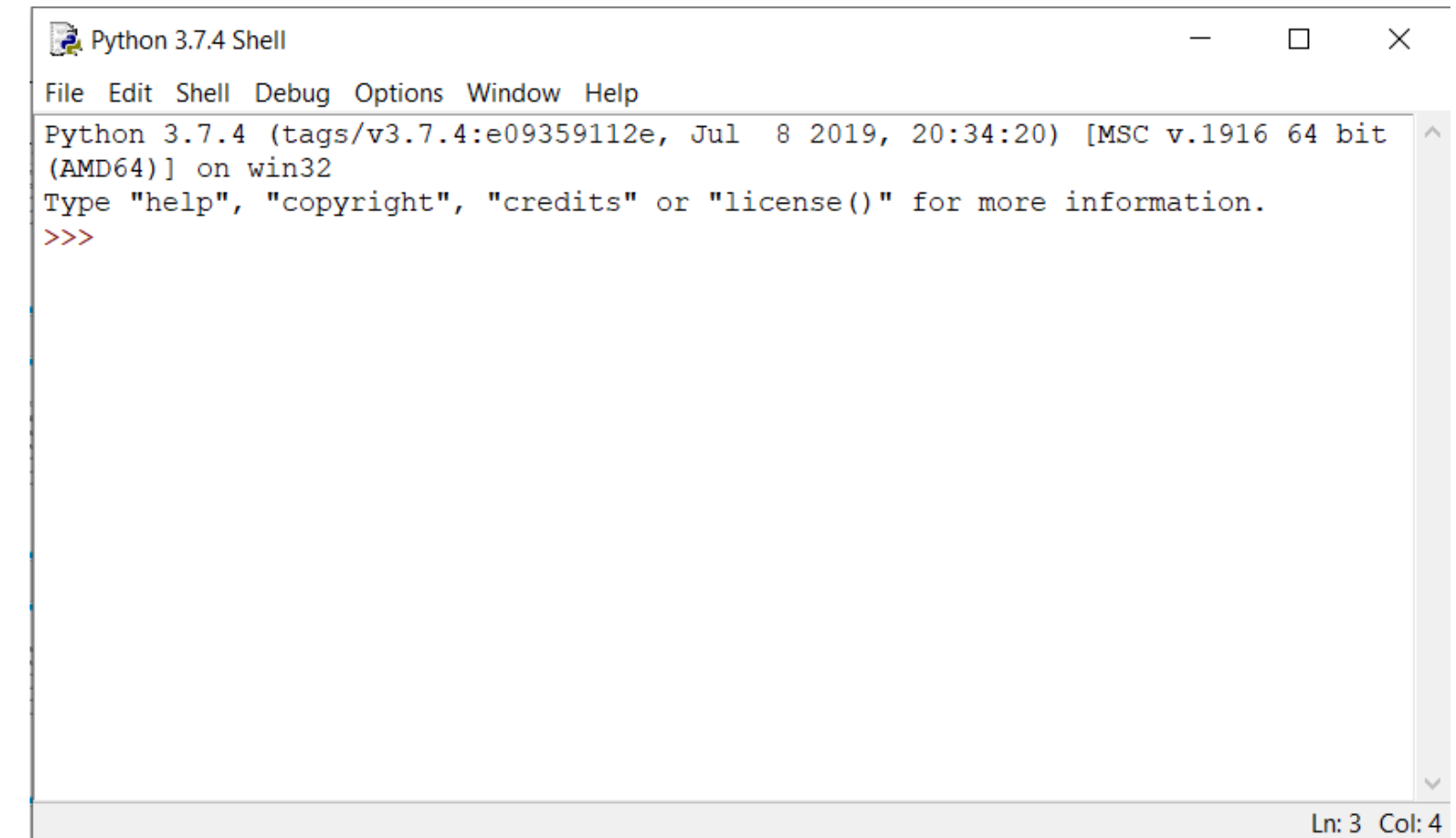
1. Antes de abrir o programa, abra o Windows Explorer, clique em **Este Computador** e, em seguida, no drive do seu computador (C: ou D:, dependendo da sua máquina);
2. Neste diretório, crie uma pasta chamada **Projetos**. Esta pasta será usada para armazenar todos os seus projetos de software;
3. Dentro da pasta de projetos, crie a pasta da disciplina (p.ex., **algoritmos**);
4. Evite utilizar caminhos muito longos (p.ex., C:\Users\12304010\Projetos\Nome-da-pessoa\Documentos\etc...) ou incluir espaços no caminhos (p.ex., C:\Victor Machado). O primeiro é muito trabalhoso para utiliza-lo recorrentemente, e o segundo pode causar alguns problemas na execução do código;
5. Sempre que criar arquivos Python, comece o nome do arquivo com uma letra (p.ex., **main.py**, **app.py**, **aula.py**). Evite usar números, espaços ou acentos nos nomes dos arquivos.



Instalando e configurando IDEs

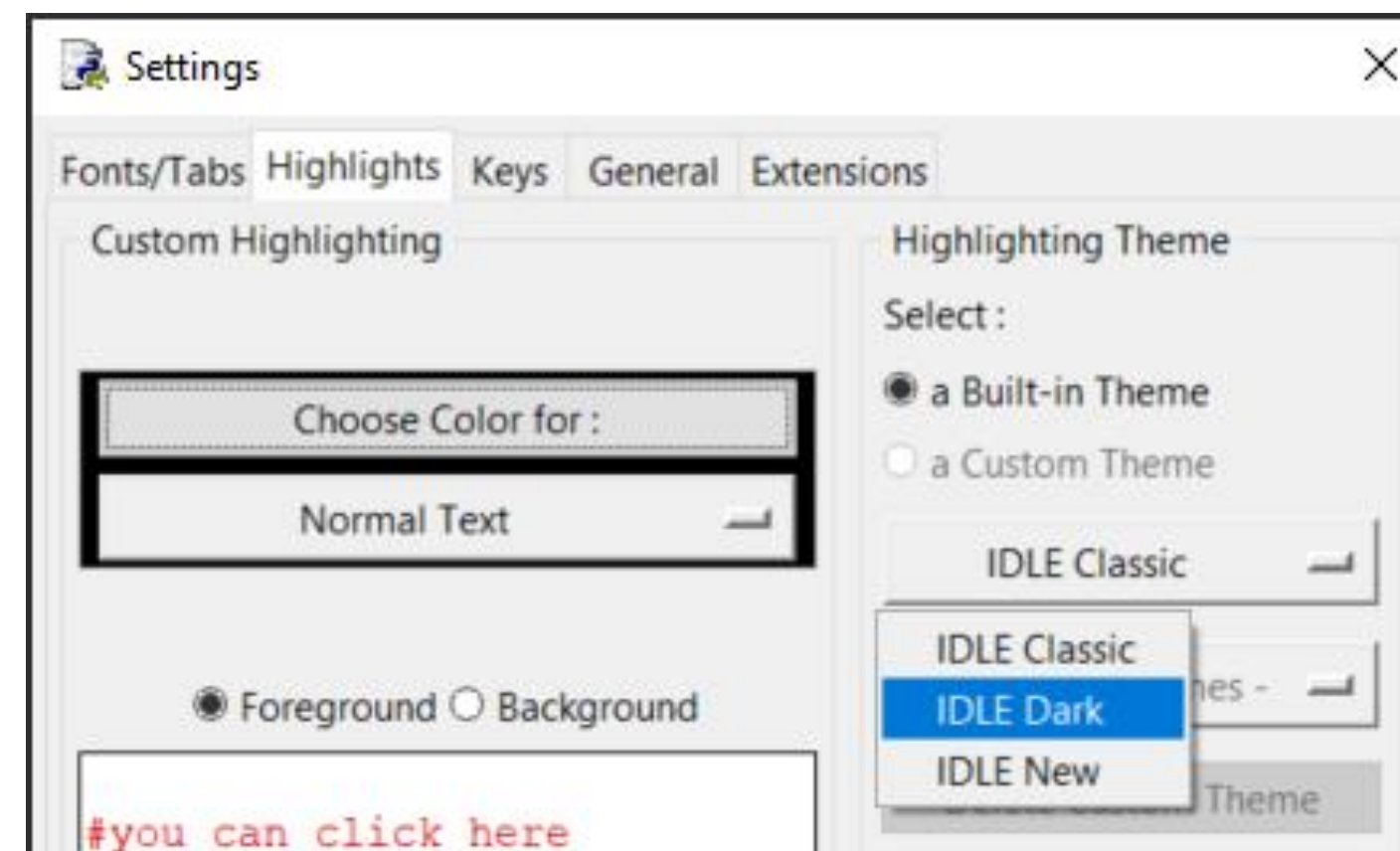
Configurando o IDLE

- Abra IDLE utilizando o ícone do Windows e procurando pelo programa
- O programa abre no modo **Shell**, que é a janela de execução do código. Usamos essa tela apenas para checar os resultados ou quando queremos executar códigos de uma linha (testar uma função, por exemplo)
- Para abrir o editor automaticamente, vá no menu **Options > Configure IDLE**
- Na aba **General**, em **Windows Preferences**, marque a opção **Open Edit Window**. Clique em Ok e reinicie o IDLE



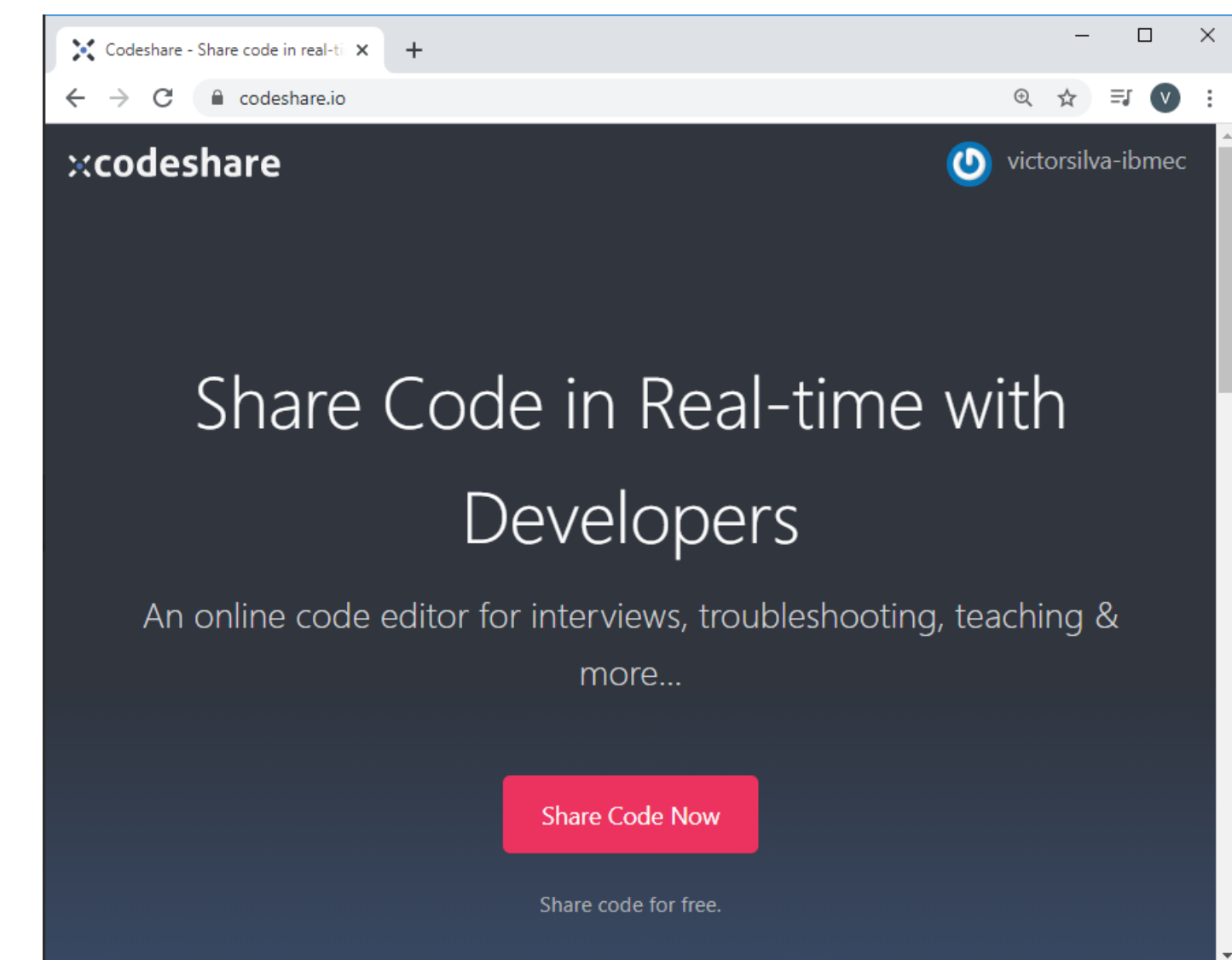
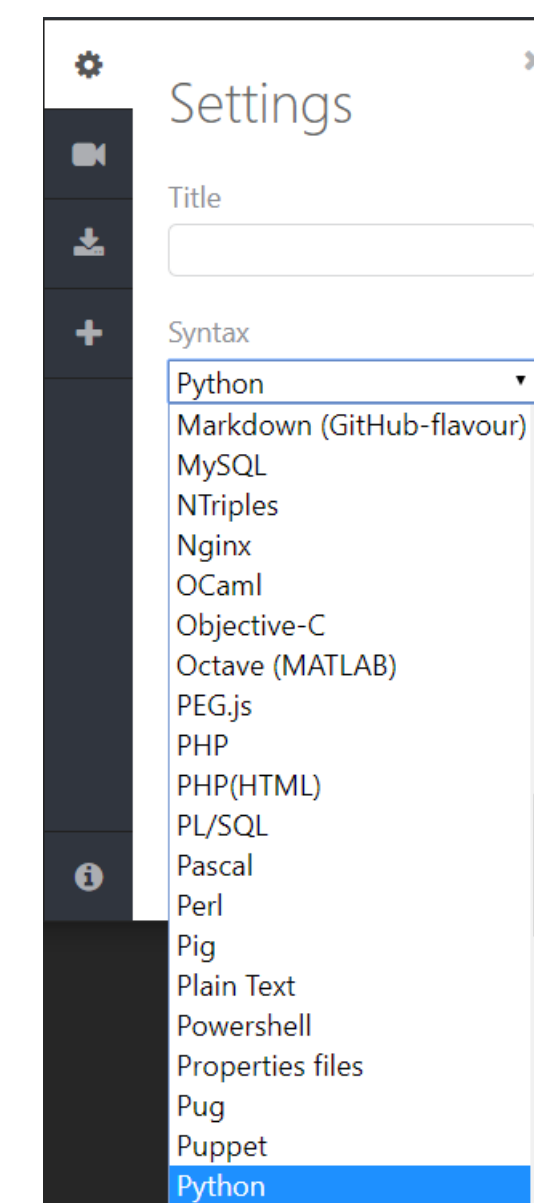
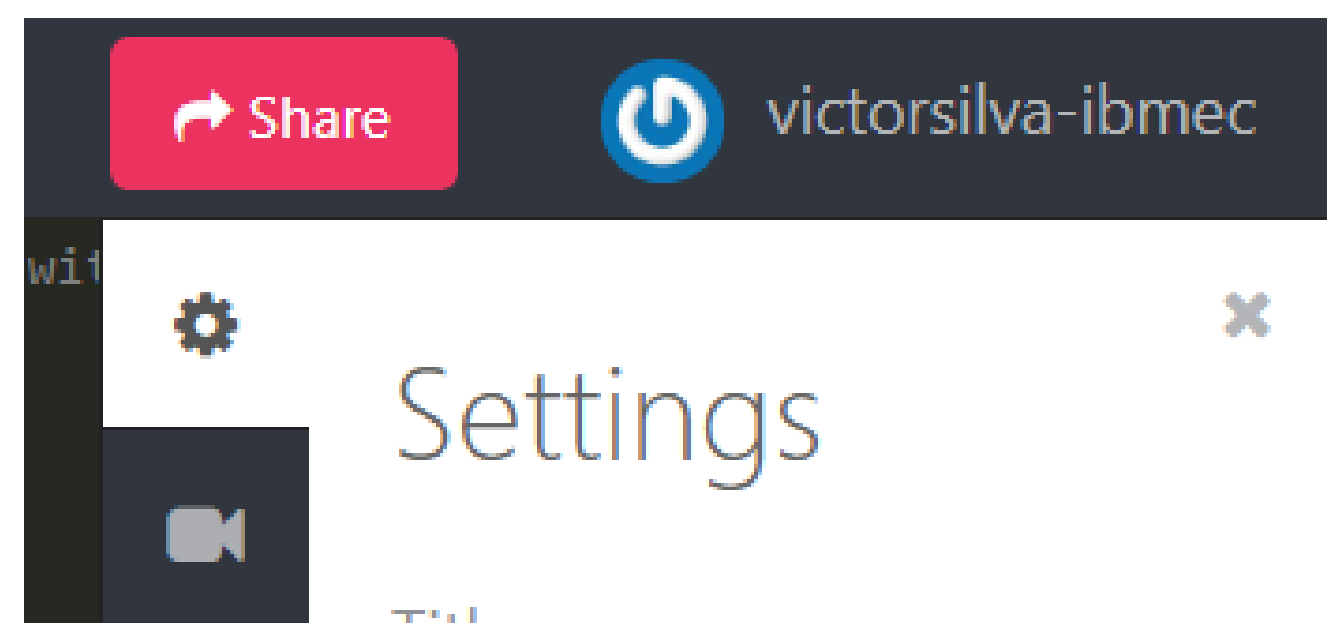
Configurando o IDLE

- Normalmente é usual programadores trabalharem com editores de texto que tenham um fundo escuro, o que prejudica menos a visão
- Na mesma tela de **Configurações**, vá para a aba **Highlights**, e na coluna da direita, em **Highlight Theme**, clique em **IDLE Classic** e selecione a opção **IDLE Dark**
- Clique em OK para mudar o tema para escuro



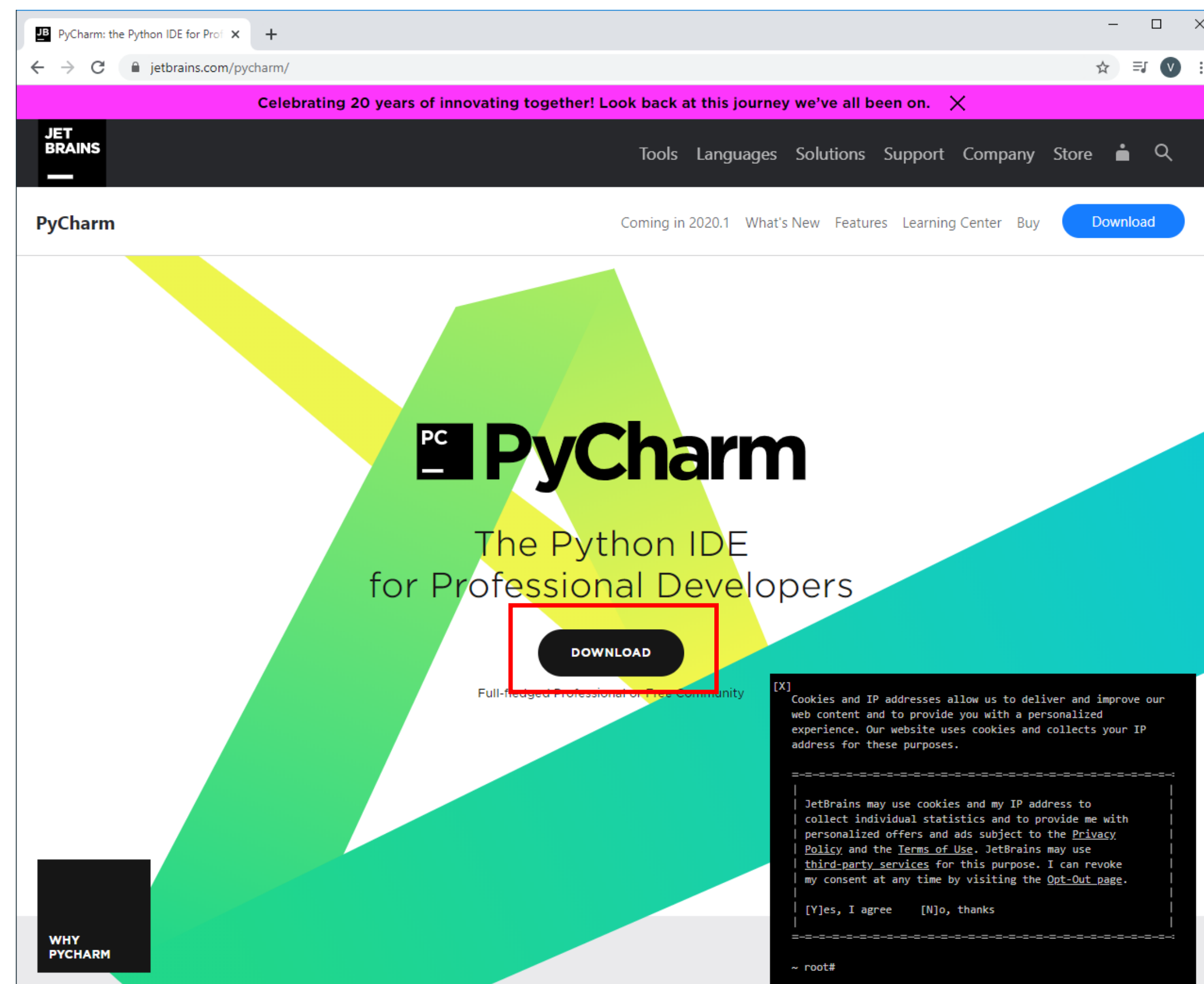
Usando o CodeShare

- Acesse <http://codeshare.io> e faça o cadastro
- Tendo cadastrado, na tela inicial clique em “Share Code Now”
- Um editor de texto vai aparecer. Na coluna da direita, clique na engrenagem, e em **Syntax** marque a opção **Python**
- Para compartilhar, clique em **Share**, no canto superior da janela
- O CodeShare vai liberar um link para ser usado por outras pessoas



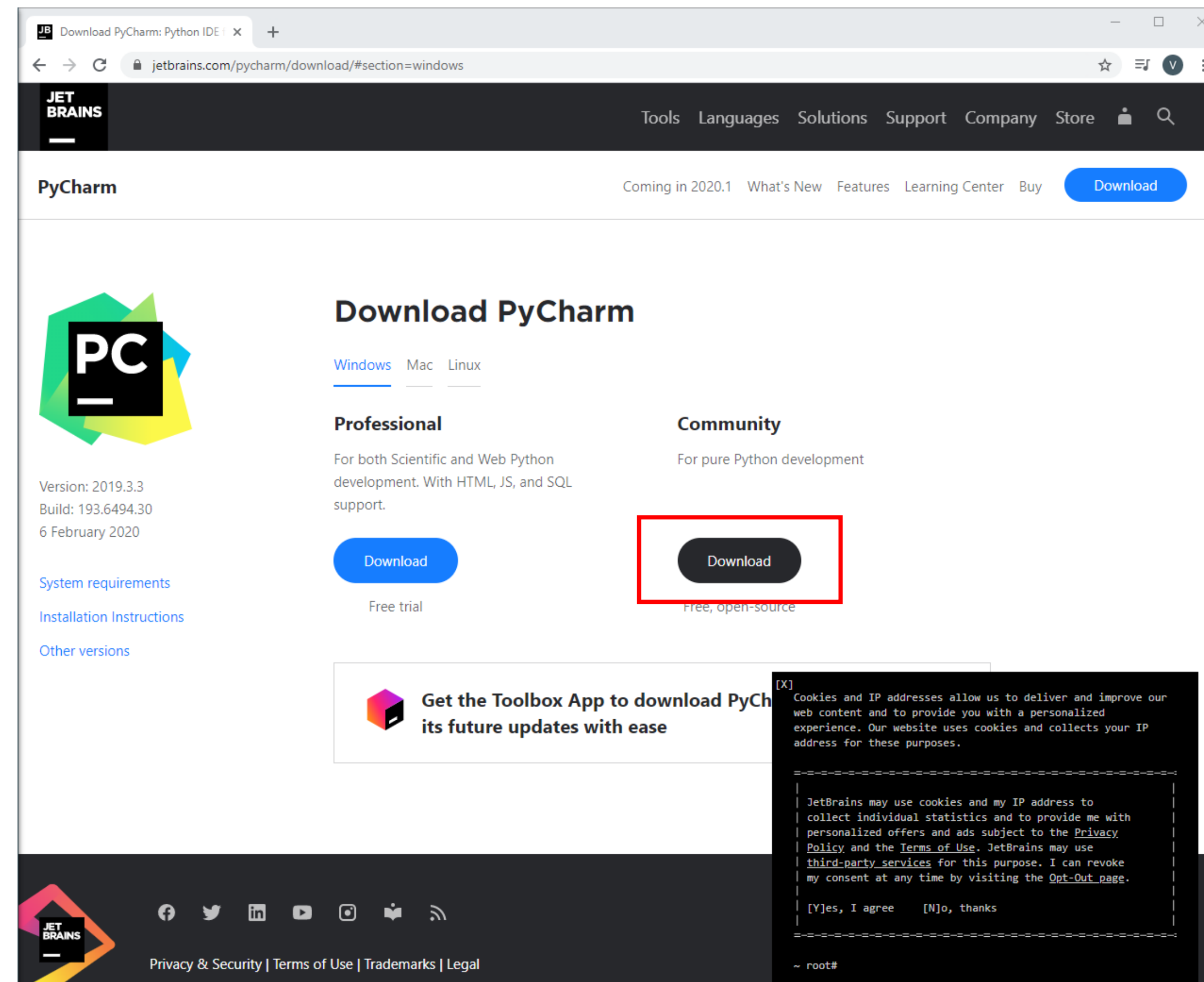
Instalando e configurando o PyCharm

- O **PyCharm** é um dos editores (ou IDEs) mais usados para a programação de aplicações em Python. Para baixar e instalar, primeiro acesse a página <https://www.jetbrains.com/pycharm/> e clique em “Download”.



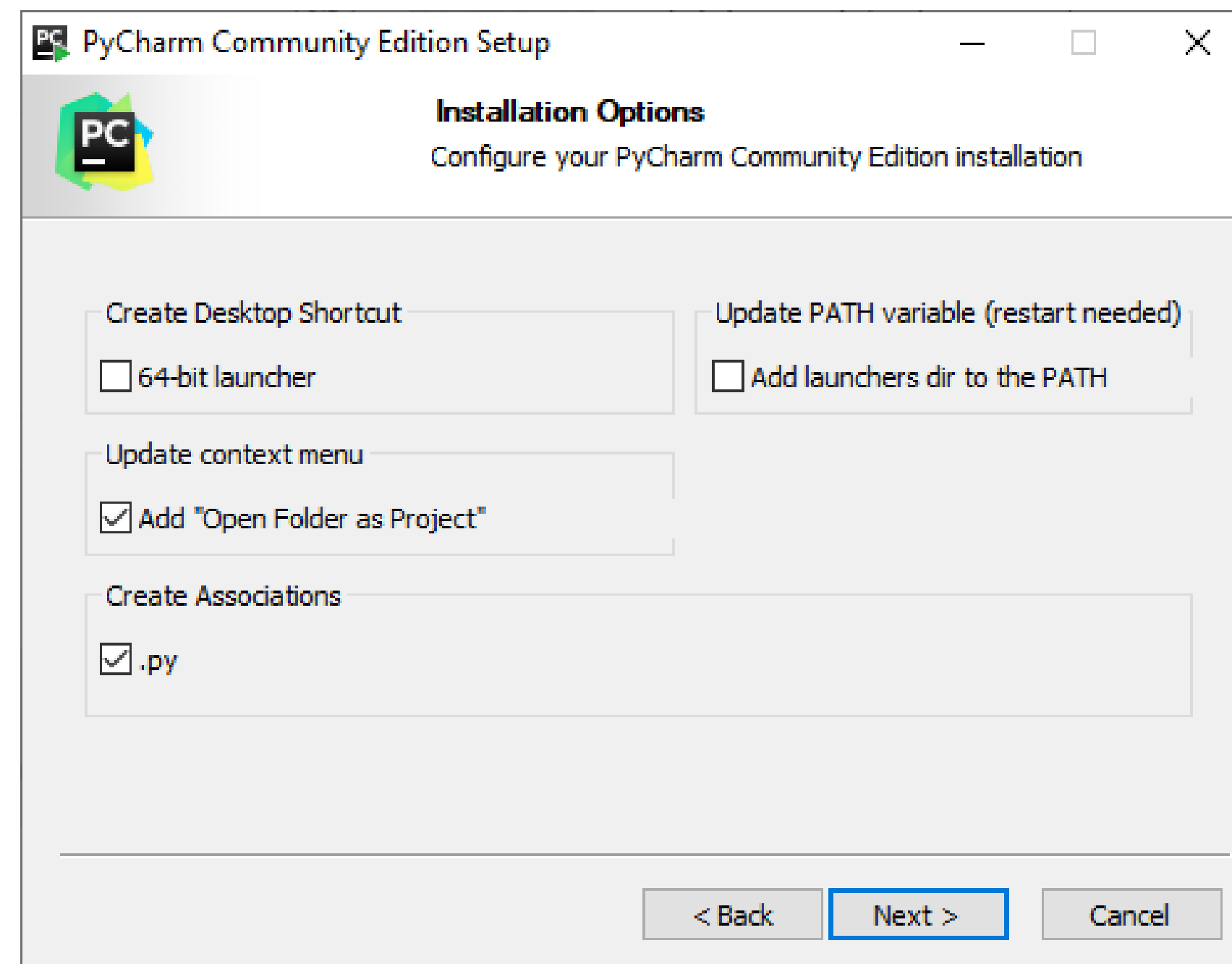
Instalando e configurando o PyCharm

- Escolha o seu sistema operacional (Windows, Mac ou Linux - vamos trabalhar com Windows) e baixe a versão “Community”. O download deve começar automaticamente.



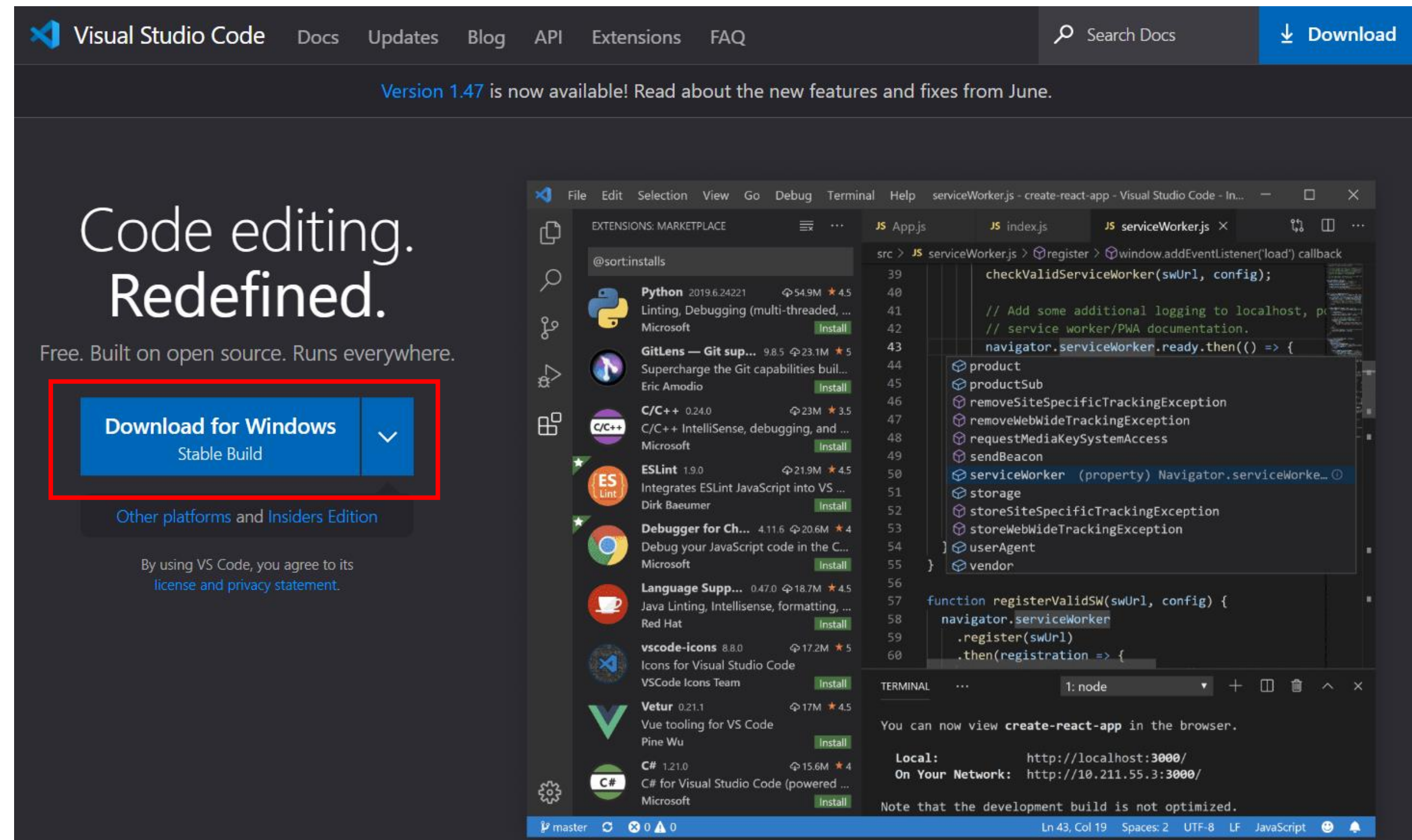
Instalando e configurando o PyCharm

- Abra o instalador do PyCharm e clique em “Next”;
- Na tela de escolha do caminho de instalação, clique em “Next”;
- Na tela seguinte, marque as opções indicadas na imagem abaixo e clique em “Next”;
- Na tela seguinte, clique em “Install” para começar a instalação.



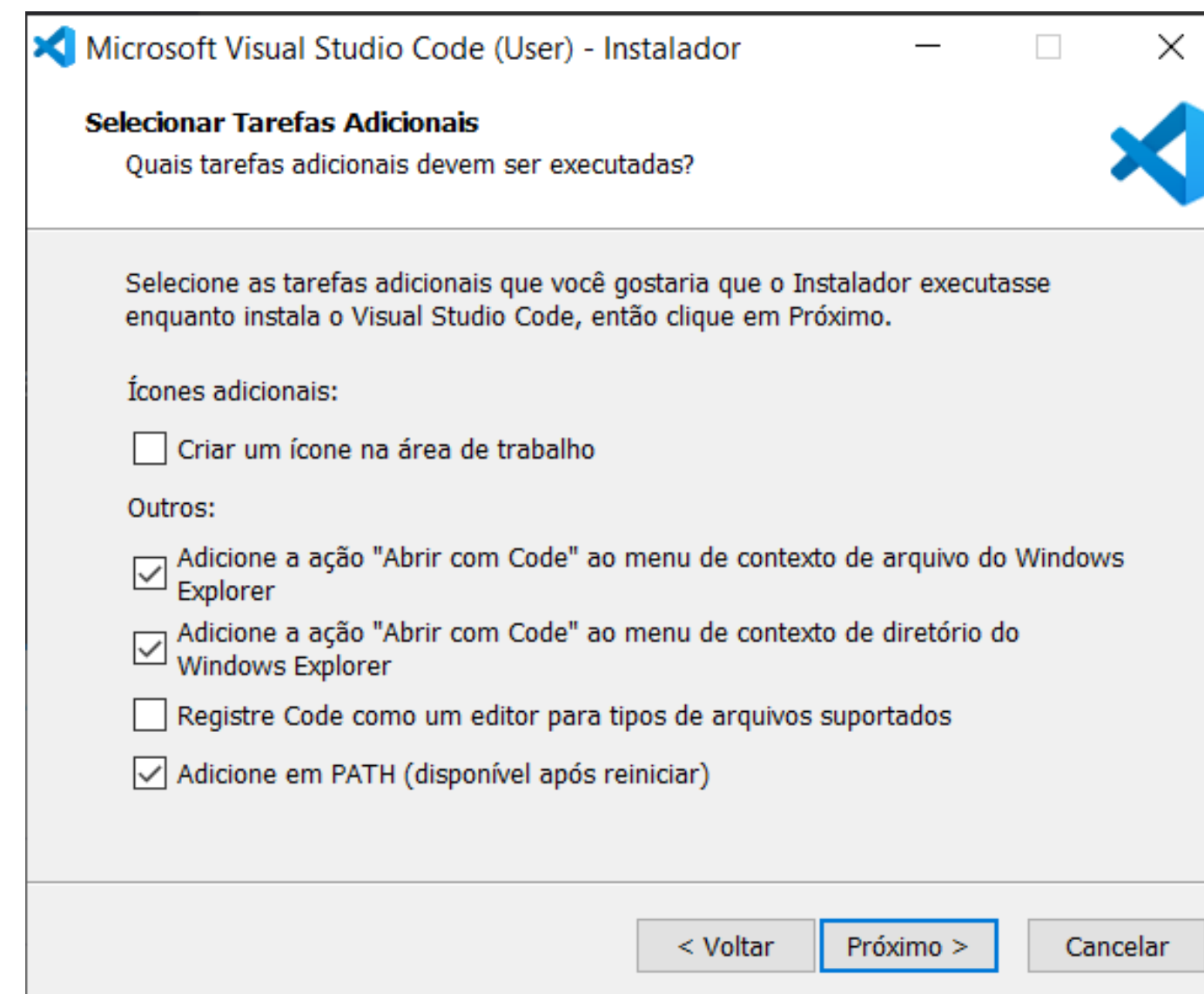
Instalando e configurando o VSCode

- O **VSCode** é um dos IDEs recentes mais famosos para basicamente qualquer linguagem de programação. Possui inúmeras extensões que facilitam e customizam o editor para a necessidade de cada programador. Para baixar e instalar, primeiro acesse a página <https://code.visualstudio.com/> e clique em “Download for Windows”. Clicando na seta à direita existem opções para MAC e Linux.



Instalando e configurando o VSCode

- Abra o instalador, e após aceitar o acordo de licença e clicar em “Próximo”, clique em “Próximo” novamente até chegar na tela “Selecionar Tarefas Adicionais”
- Marque as opções abaixo e clique em “Próximo” e depois em “Instalar”

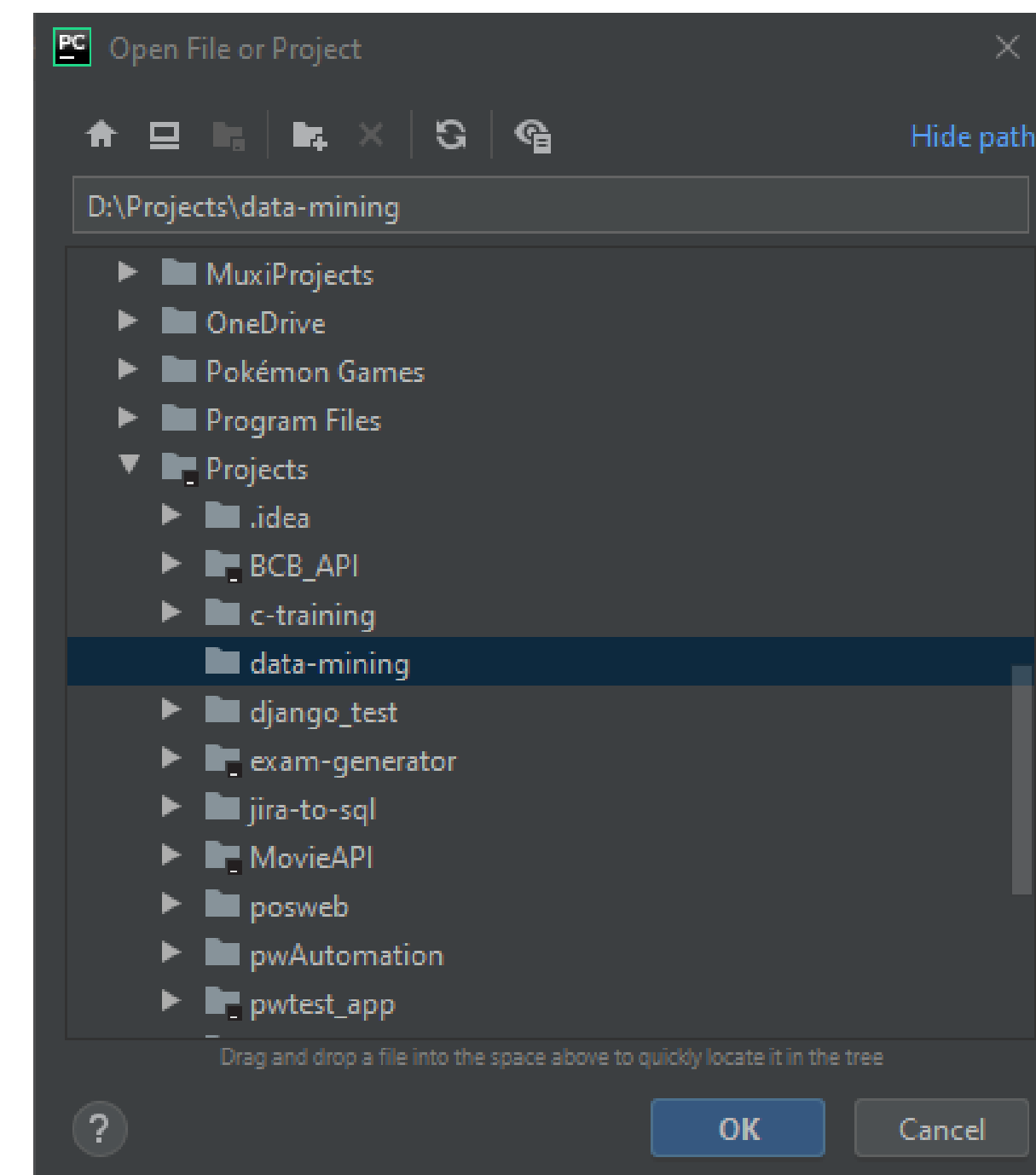
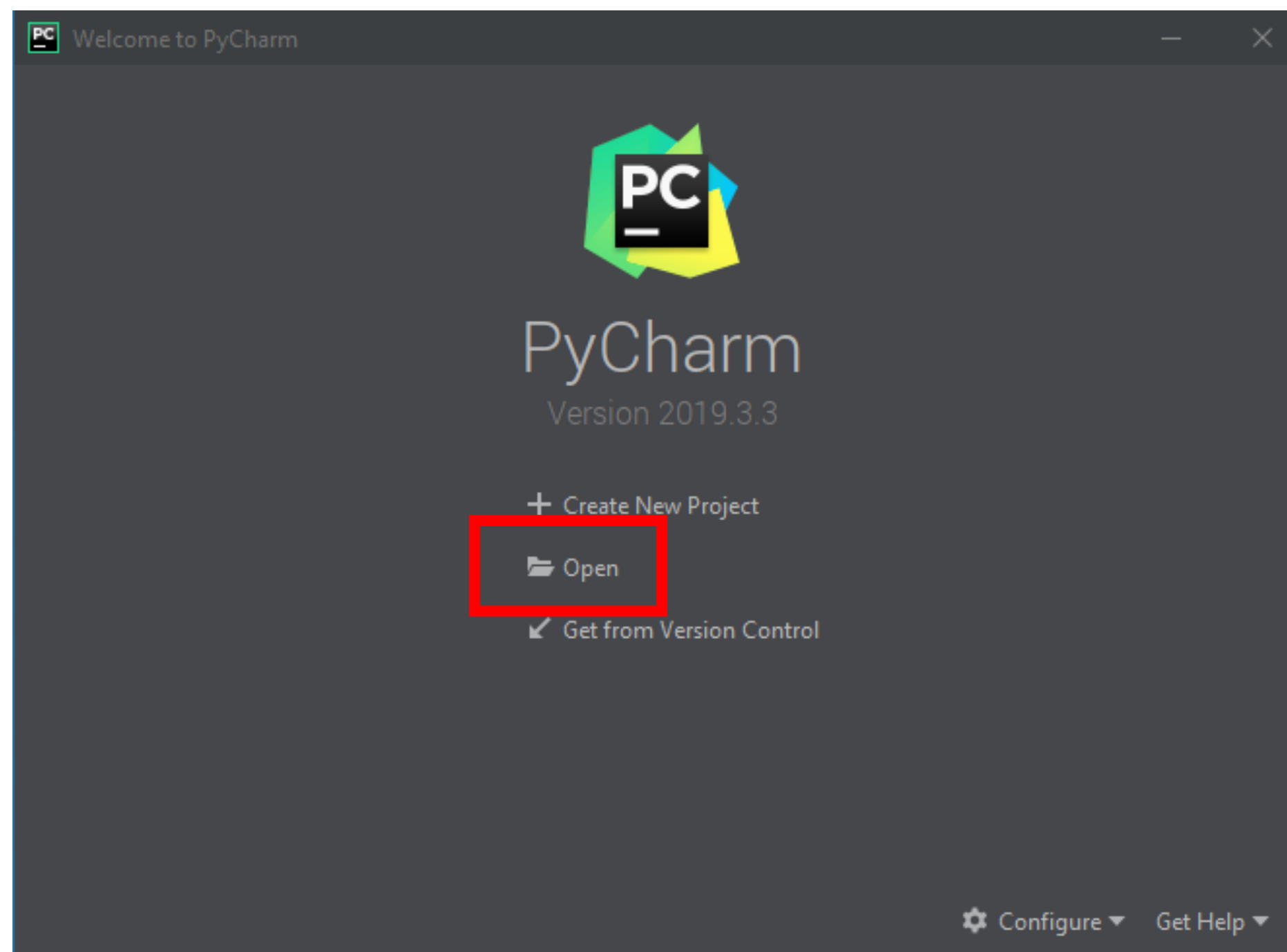




Utilizando o PyCharm

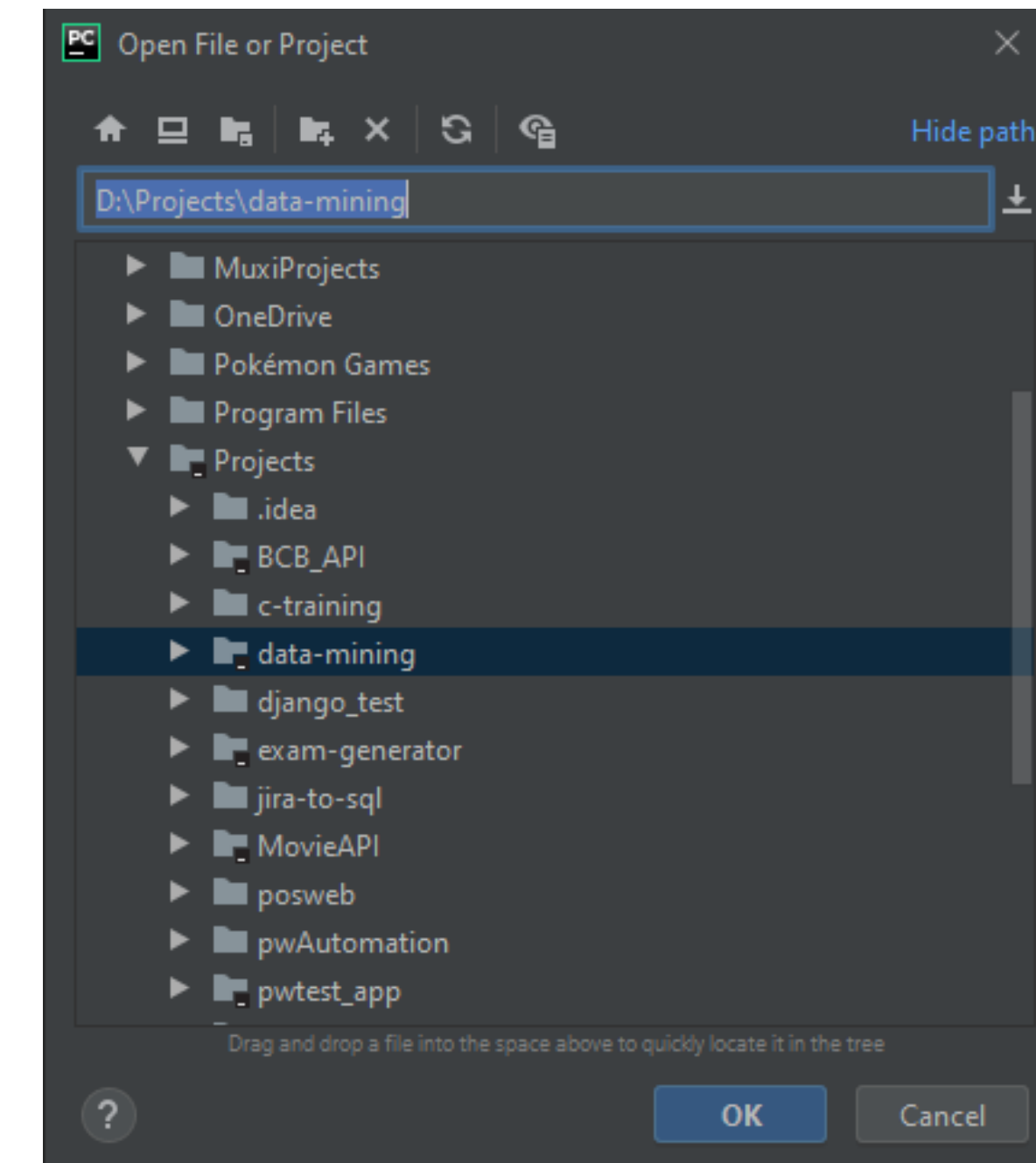
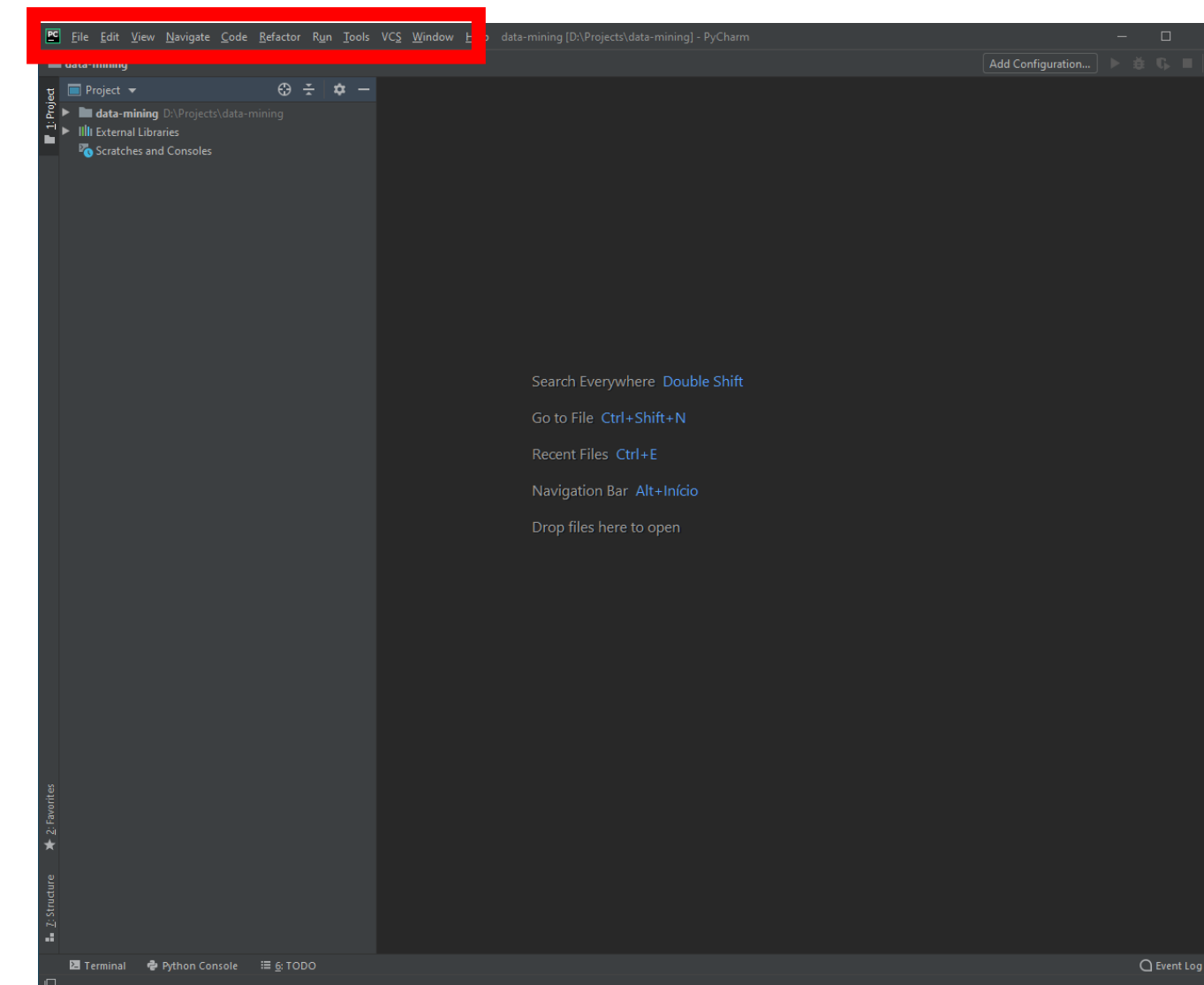
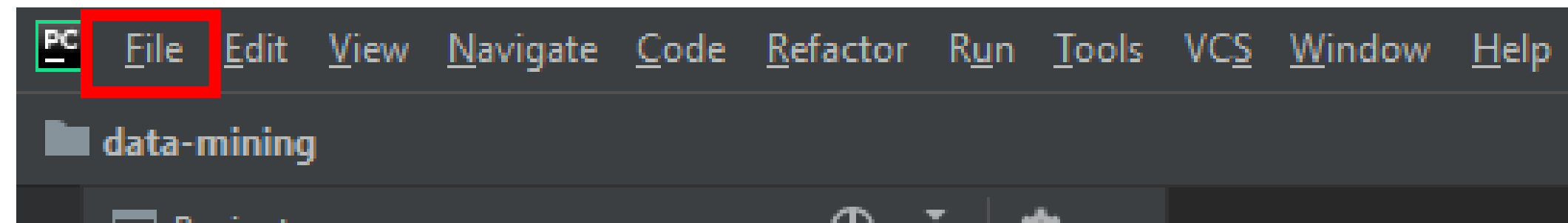
Iniciando o PyCharm

- **Se essa é a primeira vez que abre o PyCharm:**
 - Garanta que você possui uma pasta com o projeto desejado (p.ex., **D:\Projetos\data-mining**);
 - Abra o PyCharm, e na tela inicial clique em **Open**. Selecione a pasta que você criou e clique em **Ok**;



Iniciando o PyCharm

- **Se você já abriu o PyCharm antes:**
 - Garanta que você possui uma pasta com o projeto desejado (p.ex., **C:\Projetos\data-mining**);
 - Abra o PyCharm, e na tela que abrir clique em **File > Open**. Selecione a pasta que você criou e clique em **Ok**;

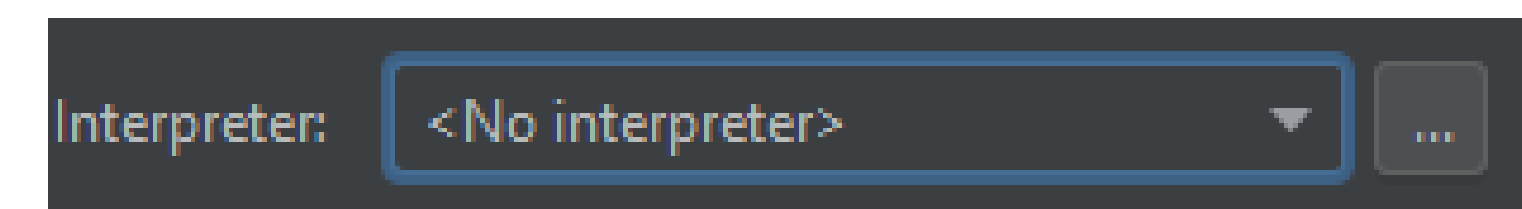
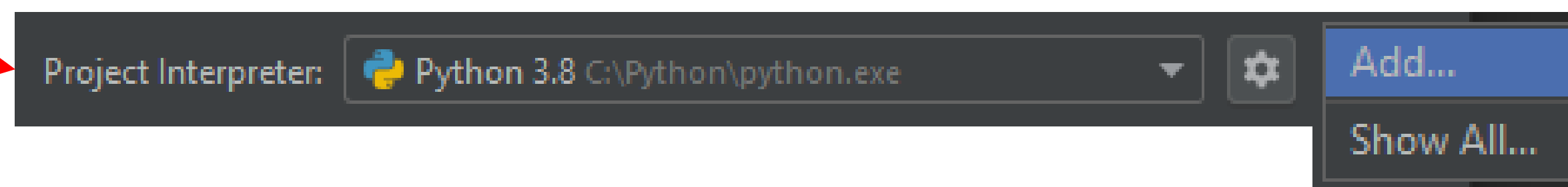


Algumas configurações do PyCharm

- **Mudar o tema para escuro:**
 - Clique em **File > Settings**;
 - Na janela que aparecer, procure por **Appearance & Behavior > Appearance**. No campo **Theme**, selecione o tema desejado (minha sugestão é o **Darcula**);
 - Clique em **Ok** para fechar a tela de configurações.

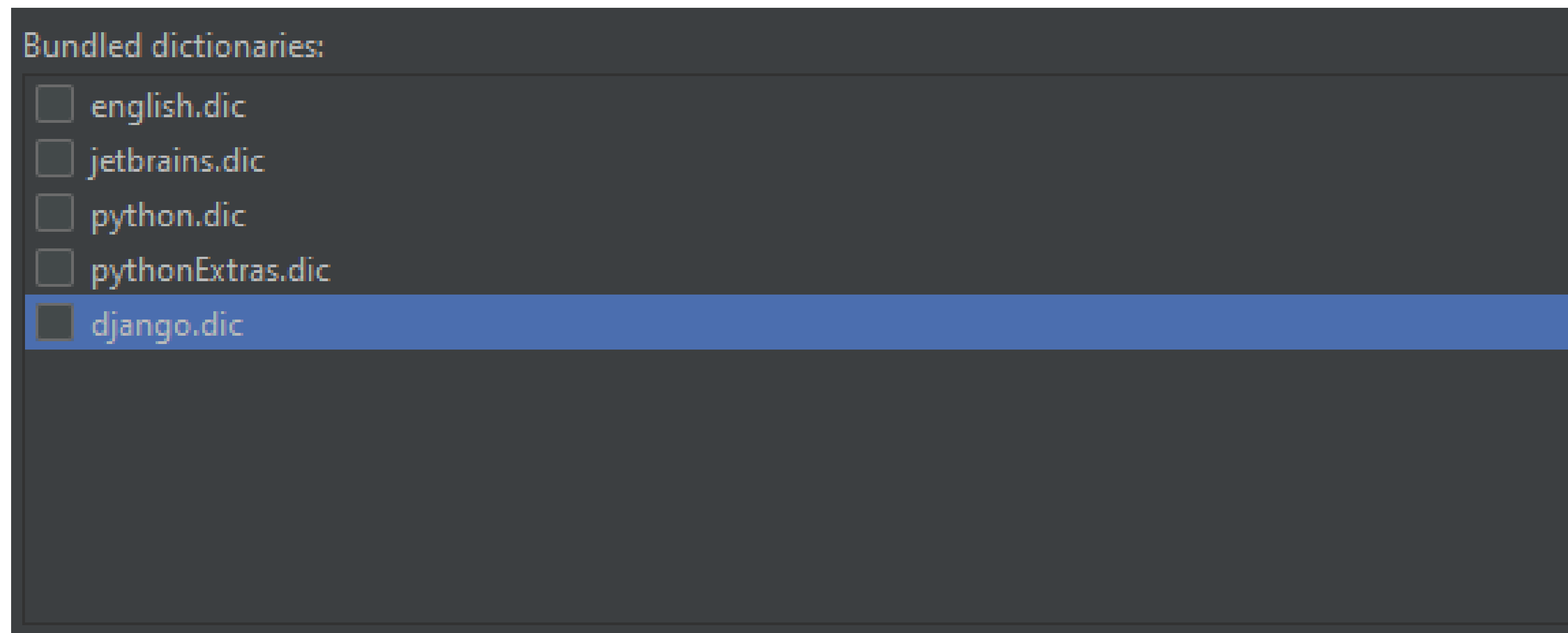
Algumas configurações do PyCharm

- **Configurar o interpretador de Python:**
 - Clique em **File > Settings**;
 - Na janela que aparecer, procure por **Project: <nome-do-projeto> > Project Interpreter**. No campo **Project Interpreter**, verifique se o Python informado é o instalado (no meu caso, **C:\Python\python.exe**). Caso não seja, clique na engrenagem e em **Add...**;
 - Na nova janela, clique em **System Interpreter**, e no campo **Interpreter** clique nos três pontos à direita para indicar o local que o seu Python está instalado. Aperte **Ok** até voltar à tela de **Settings**;
 - Novamente no campo **Project Interpreter**, altere o Python para refletir o que está instalado. Em seguida clique em **Ok**.



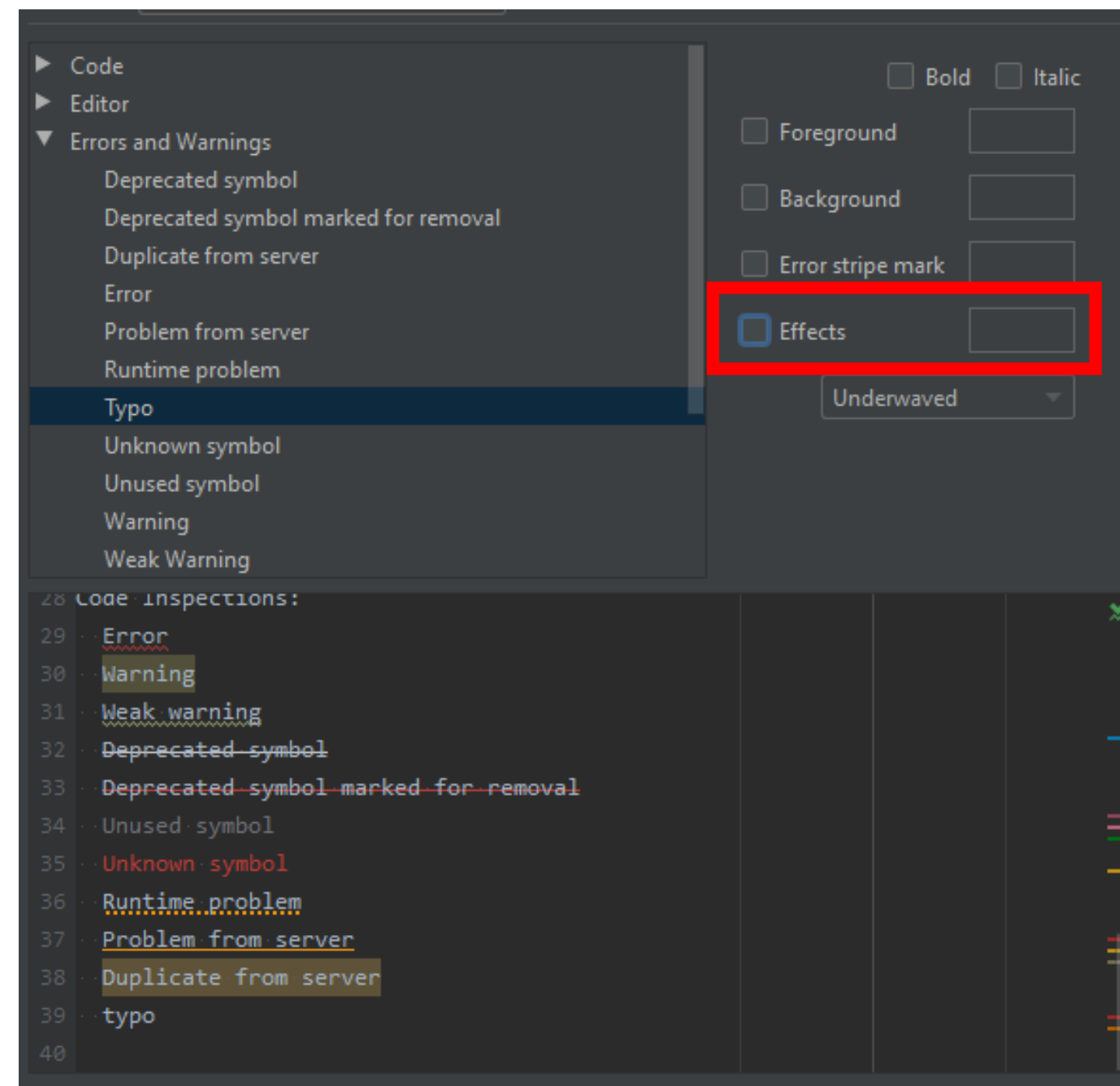
Algumas configurações do PyCharm

- **Desativar inspeção ortográfica:**
 - Clique em **File > Settings**;
 - Na janela que aparecer, procure por **Editor > Spelling**. No campo **Bundled dictionaries**, desmarque todas as opções;
 - Clique em **Ok** para sair das configurações.



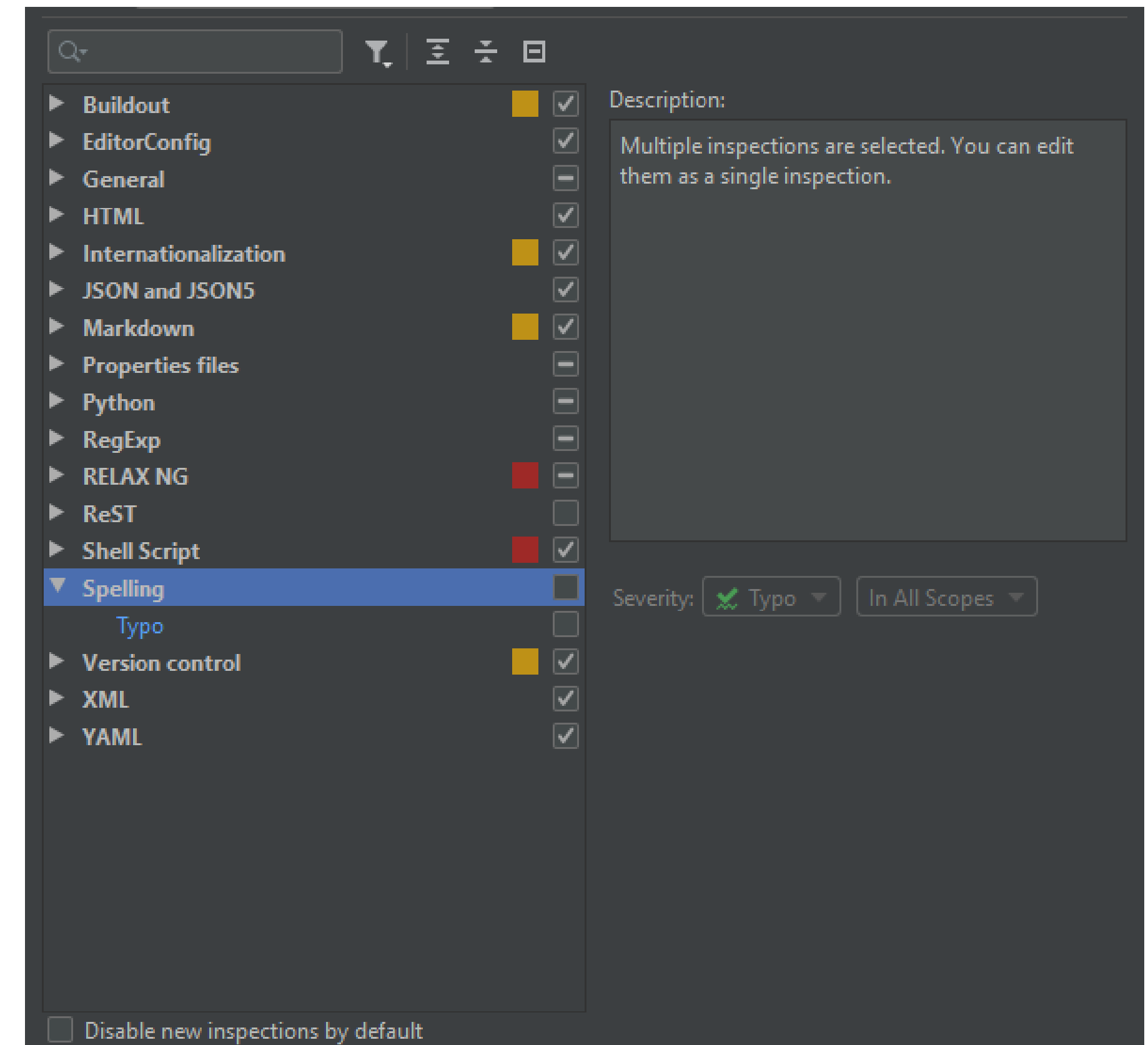
Algumas configurações do PyCharm

- **Desmarcar esquema de cores para *tipos*:**
 - Clique em **File > Settings**;
 - Na janela que aparecer, procure por **Editor > Color Scheme > General**. No campo **Errors and Warnings > Typo**, desmarque a opção **Effects**;
 - Clique em **Ok** para sair das configurações.



Algumas configurações do PyCharm

- **Ajustar os destaques de inspeções:**
 - Clique em **File > Settings**;
 - Na janela que aparecer, procure por **Editor > Inspections**. Desmarque o campo **Spelling**;
 - Na seção **Python**, recomendo *desmarcar* algumas opções para simplificar o aprendizado:
 - *Boolean variable check can be simplified*;
 - *Chained comparisons can be simplified*;
 - *Comparison with None performed with equality operators*.
 - Clique em **Ok** para sair das configurações.



Algumas configurações do PyCharm

- **Configurar uma determinada execução de código:**
 - Garanta que as configurações do slide **Configurar o interpretador de Python** foram executadas;
 - Aperte **Alt + Shift + F10 > Edit Configurations**, ou na barra de tarefas clique em **Run > Edit Configurations**;
 - Na janela que aparecer, no canto superior esquerdo clique no botão de **+ > Python**;
 - Dê um nome para a execução (p.ex., **aula**) e em **Script path**, informe o caminho do arquivo que você deseja executar;
 - Certifique-se que o **Python interpreter** é o configurado anteriormente, clique em **Apply** e em seguida em **Close**;
 - Para rodar o arquivo, é só usar o atalho **Shift + F10**.





Utilizando o VSCode

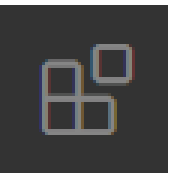
Iniciando o VSCode

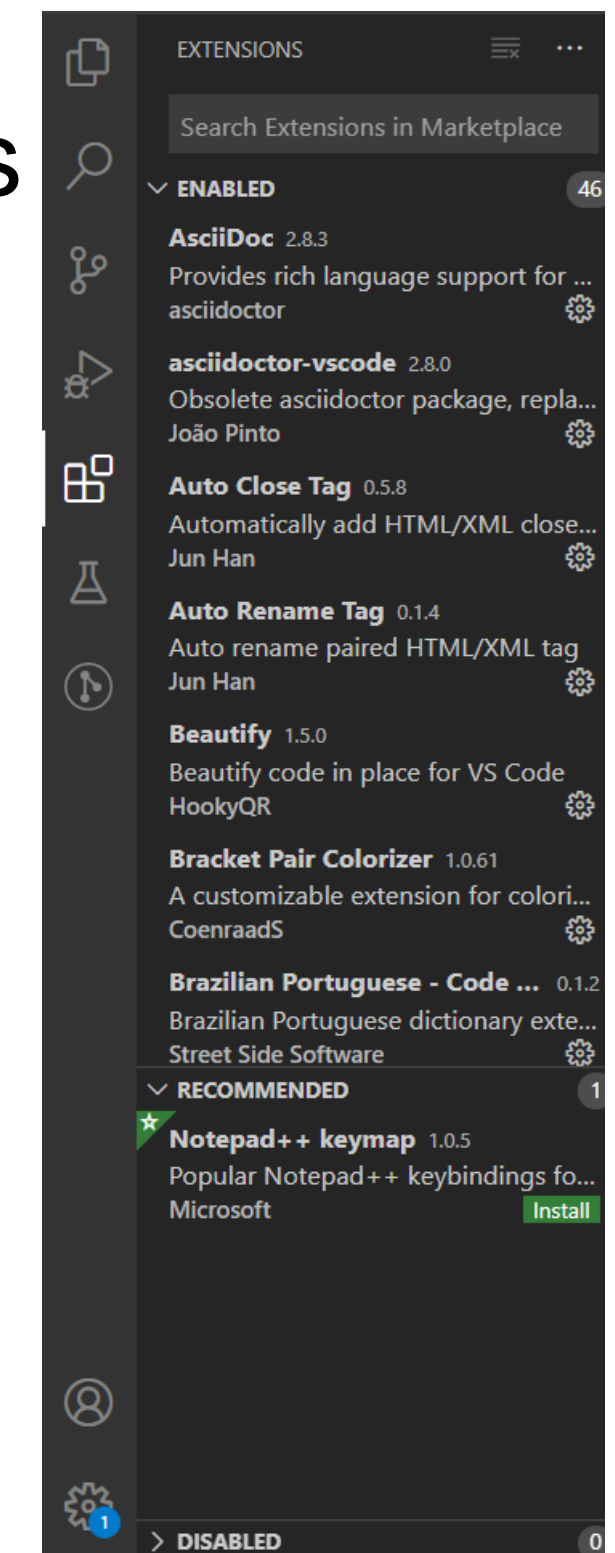
- **Se essa é a primeira vez que abre o VSCode:**
 - Garanta que você possui uma pasta com o projeto desejado (p.ex., **D:\Projetos\algoritmos**);
 - Abra o VSCode, e na tela inicial clique em **File > Open Folder**. Selecione a pasta que você criou e clique em **Selecionar Pasta**.



Iniciando o VSCode

- **Instalando extensões:**

- Boa parte do atrativo no VSCode é a possibilidade de instalar extensões e customizar a experiência de uso;
- Para instalar extensões, vá na coluna à esquerda e clique no ícone  ;
- Uma barra de extensões vai aparecer, e você poderá buscar as extensões desejadas e instalá-las;
- Abaixo seguem algumas sugestões de extensões para o curso:
 - Beautify
 - Bracket Pair Colorizer
 - Brazilian Portuguese - Code Spell Checker
 - C/C++
 - Code Runner
 - GitLens - Git supercharged
 - Git History Diff
 - Git History
 - Gitconfig Syntax
 - Markdownlint
 - Partial Diff
 - Python
 - Visual Studio IntelliCode
 - vscode-python-docstring
 - AsciiDoc



Iniciando o VSCode

- **Atalhos interessantes no VSCode:**
 - Ctrl + K, Ctrl + O: Abre uma pasta
 - Ctrl + D: Quando uma palavra estiver selecionada, seleciona todas as palavras no arquivo
 - Ctrl + F: Procura por uma palavra ou sentença no arquivo
 - Ctrl + Shift + F: Procura por uma palavra ou sentença em todos os arquivos da pasta
 - Ctrl + H: Substitui uma palavra ou sentença por outra em todo o arquivo
 - Alt + ↓ ou Alt + ↑: Move a linha inteira para baixo ou para cima
 - Shift + Alt + ↓ ou Shift + Alt + ↑: Copia a linha inteira para baixo ou para cima
 - Ctrl + Alt + ↓ ou Ctrl + Alt + ↑: Inclui um ou mais cursores nas linhas abaixo ou acima

Iniciando o VSCode

- **Atalhos interessantes no VSCode:**
 - F12: Vai para a declaração de uma variável ou função
 - F12 F12: Mostra todos os usos de uma variável ou função
 - Alt + → ou Alt + ←: Retrocede ou avança na última posição do cursor
 - Ctrl + S: Salva um arquivo
 - Ctrl + P: Abre um arquivo diretamente
 - Ctrl + ,: Abre o menu de configurações
 - Ctrl + Shift + P: Abre uma dropdown com opções de configuração
 - Ctrl + `: Abre a janela do terminal

Iniciando o VSCode

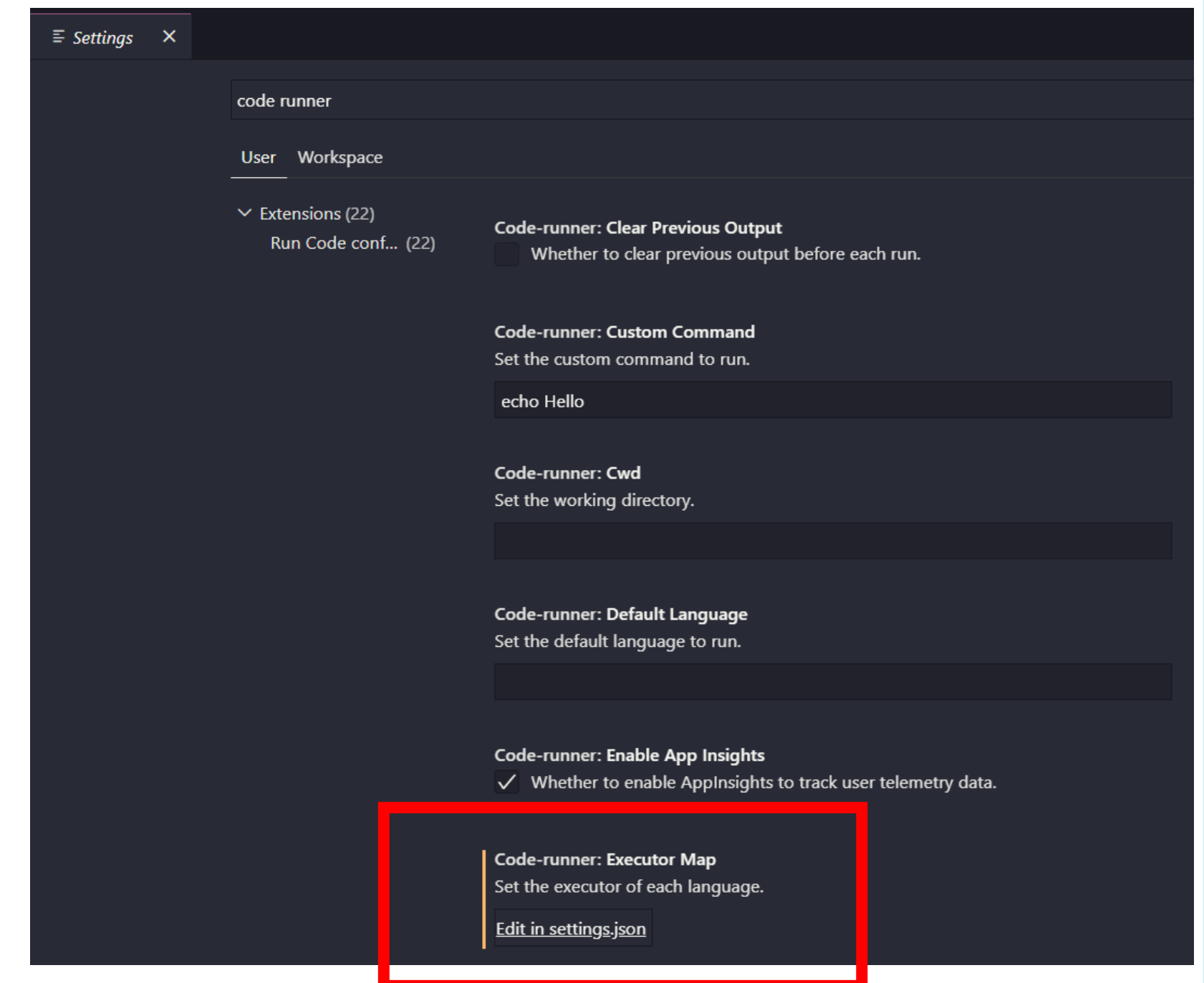
- **Atalhos interessantes no VSCode:**
 - Ctrl + G: Vai para uma linha específica do arquivo
 - Ctrl +]: Divide a tela em duas
 - Ctrl + 1 (ou 2, 3, 4): Seleciona um painel específico
 - Alt + Shift + 0: Alterna entre divisão na horizontal ou na vertical
 - Ctrl + F4 ou Ctrl + W: Fecha o arquivo selecionado (se for um painel selecionado e ele estiver vazio, fecha o painel)
 - Ctrl + ;: Comenta a(s) linha(s) selecionada(s)

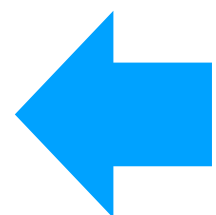
Iniciando o VSCode

- **Executando um código Python no VSCode com a extensão *Code Runner*:**
 - Essa extensão permite a execução do código através de um atalho
 - Com a extensão instalada, use o atalho **Ctrl + ,** ou vá em **File > Preferences > Settings**;
 - Na barra de pesquisa, digite “Code Runner”. Procure pela caixa abaixo e clique em “Edit in settings.json”;
 - Inclua as seguintes linhas (edite o endereço para o caminho em que o Python está instalado):

```
"python.pythonPath": "C:\\Python\\python.exe",  
"code-runner.executorMap": {  
  "python": "\"C:\\Python\\python.exe\""  
},  
"code-runner.runInTerminal": true,
```

- Para rodar, na janela do código use o atalho **Ctrl + Alt + N**.





Configurando o pylint

Configurando o pylint

- O pylint é um pacote do Python que auxilia na adequação do código aos padrões de programação da comunidade Python
- Para instalar o pylint...
 - ...para MacOS veja [aqui](#)
 - ...para Windows veja [aqui](#)
- Para executar o pylint basta entrar com o comando **pylint <caminho>**, com o caminho do arquivo que deseja rodar o linter.

```
C:\Users\vmachado>pylint D:\Victor\Pessoal\IBMEC\2020.2\ALG\Aulas\teste.py
***** Module teste
D:\Victor\Pessoal\IBMEC\2020.2\ALG\Aulas\teste.py:1:0: C0114: Missing module docstring (missing-module-docstring)

-----
Your code has been rated at 0.00/10
```


Configurando o pylint

- O pylint ainda pode ser rodado direto no terminal do VSCode ou do PyCharm, basta abrir o terminal e seguir os mesmos passos mencionados anteriormente. No entanto, os ambos os IDEs fornecem meios de utilizar o pylint diretamente durante a implementação do código.
- **Usando o pylint no PyCharm:**
 - O PyCharm já possui a inspeção automática do editor, após configurar o interpretador. Para identificar as inspeções vá em **File > Settings** e na janela clique em **Editor** e depois em **Inspections**;
 - O PyCharm também possui um plugin. Em **File > Settings**, clique em **Plugins** e busque por **Pylint**. Instale e reinicie o IDE. Sempre que abrir um arquivo Python o IDE vai incluir a opção “Pylint” no canto inferior esquerdo. Para executar basta clicar na opção e depois em “Run”.



Configurando o pylint

- **Usando o pylint no VSCode:**

- Com a extensão “Python” instalada no VSCode e o pacote pylint instalado, abra o menu de configurações (use o atalho **Ctrl + ,** ou vá em **File > Preferences > Settings**) e procure por python linting;
- Marque as seguintes opções:
 - Python > Linting: Enabled
 - Python > Linting: Lint On Save
 - Python > Linting: Pylint Enabled
- Para rodar o pylint sem ser pelo terminal, use o atalho **Ctrl + Shift + P** e em seguida digite “Run Linting” e aperte Enter. O IDE vai marcar no código os problemas.

Python > Linting: Enabled

☒ Whether to lint Python files.

Python > Linting: Lint On Save

☒ Whether to lint Python files when saved.

Python > Linting: Pylint Enabled

☒ Whether to lint Python files using pylint.



Instalando pacotes pelo PyPI

Instalando pacotes pelo PyPI

- Uma das grandes vantagens ao se programar em Python é ter à disposição uma gama de pacotes e bibliotecas disponíveis pela própria comunidade, que desenvolve novas funcionalidades e distribui online, na maioria das vezes de forma gratuita.
- Um dos locais mais confiáveis e mais simples de se obter um novo pacote é através do PyPI, um repositório oficial de pacotes da linguagem, mantido pela própria organização que mantém o Python.
- O PyPI é acessado utilizando uma ferramenta chamada **pip**, que é instalada automaticamente ao se instalar o interpretador de Python. Portanto, a instalação de novos pacotes é muito fácil de se realizar, com apenas uma linha de comando.

Instalando pacotes pelo PyPI

- Instalando pacotes via pip no Windows:
 - Clique no botão do Windows ;
 - Digite a caixa de pesquisa **prompt de comando**, e abra o programa;
 - No terminal entre com o seguinte comando:

```
C:\Users\vmachado>C:\Python\python.exe -m pip install pylint
```
- Lembre-se de substituir o caminho do Python para o caminho instalado no seu computador, e altere **pylint** para o pacote escolhido.

Instalando pacotes pelo PyPI

- A instalação do Python no MacOS não costuma vir com o pip. Caso não tenha vindo, tente fazer os passos abaixo primeiro:

- Abra o terminal (pasta **Applications > Utilities > Terminal**);
- Entre com os seguintes comandos:

```
curl https://bootstrap.pypa.io/get-pip.py > get-pip.py  
sudo python get-pip.py
```

- Para instalar um pacote via pip no MacOS:
 - Usando o mesmo terminal usado para instalar o pip, entre com o seguinte comando:

```
sudo pip install <nome_do_pacote>
```

- Veja o vídeo abaixo para mais detalhes:
 - <https://www.youtube.com/watch?v=yBdZZGPpYxg>

OBRIGADO!



www.ibmec.br

 /ibmec

 ibmec

 @ibmec_oficial

 ibmec

