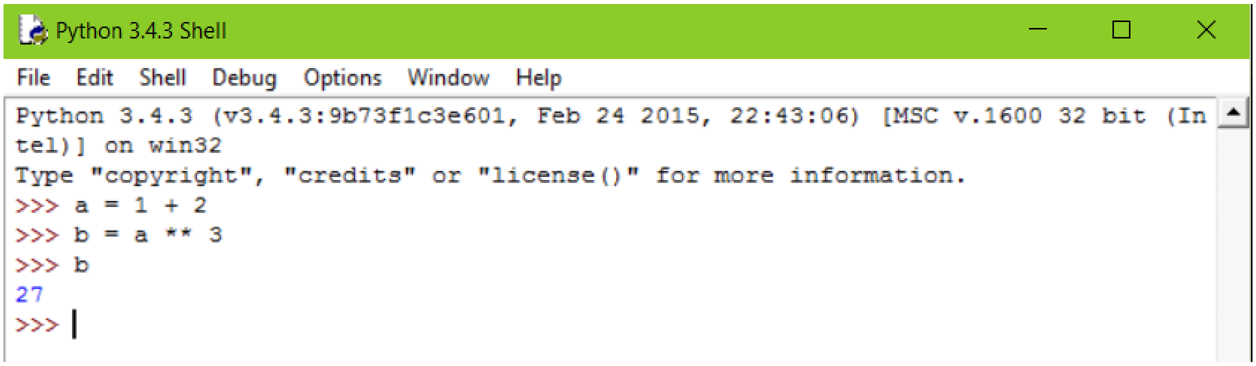
Installation of Python - Windows

If you already have Python installed on your computer, please check its version.
We will be teaching Python3. There are many major differences between Python2.x and Python3.x, so make sure you have at least one version of Python3 installed.
For the purpose of learning Python, feel free to use Python3.5 or another version. However, it is best to get **Python3.4** working with PyGame. This is the version we will be using later on.

1) Download the Python installer from http://programarcadegames.com/python-3.4.3.msi to install **Python3.4** on your machine.

2) Run the Python installer you downloaded by double-clicking it.
   a) Accept all the defaults.
   b) This should install Python3.4 in the directory (Just a fancy name for folder) "**C:\Python34**".
3) Python comes with a default Integrated Development Environment (IDLE); you can open it from the Start Menu. You can type any Python3 code in this interactive console and get a response right away. The Python Shell (IDLE) is going to be your friend. If you are not sure whether a certain command would work or not, you can always test it here first.



```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = 1 + 2
>>> b = a ** 3
>>> b
27
>>> 
```

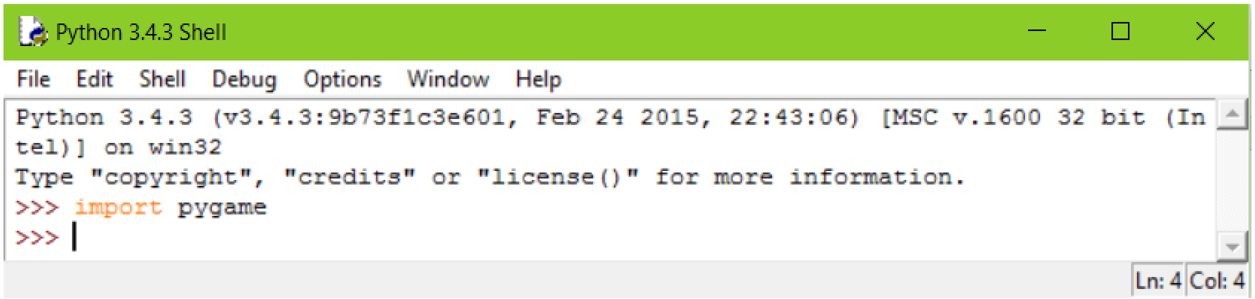
Installation of PyGame - Windows

If you are working with a computer that already has PyGame set up on it, feel free to skip this step, but if you want to set up (Python and) PyGame on your own Windows computer, don't worry. It is very easy.

We choose to use Python3.4 and PyGame1.9.2, even though the latest version of Python is 3.5.

It is possible to get PyGame working with Python3.5, but is more involved. If you really want to use Python3.5 with PyGame, you can ask one of the assistants for help.

4) Download the PyGame installer from http://programarcadegames.com/pygame-1.9.2a0.win32-py3.4.msi
5) Run the PyGame installer you downloaded by double-clicking it.
    a) When asked to “**Select Python Installations**”, select the first one “**Python 3.4 from Registry**” and PyGame will be installed in the default Python3.4 directory.
    b) If you installed Python3.4 in a different location, choose the second option “**Python from another location**” and provide the folder in which you installed Python3.4.
6) To verify that PyGame is installed correctly, you can run `import pygame` in a Python Shell (IDLE). You can open it up from the Start Menu. If it does not shout at you, congratulations, you have PyGame ready to go.



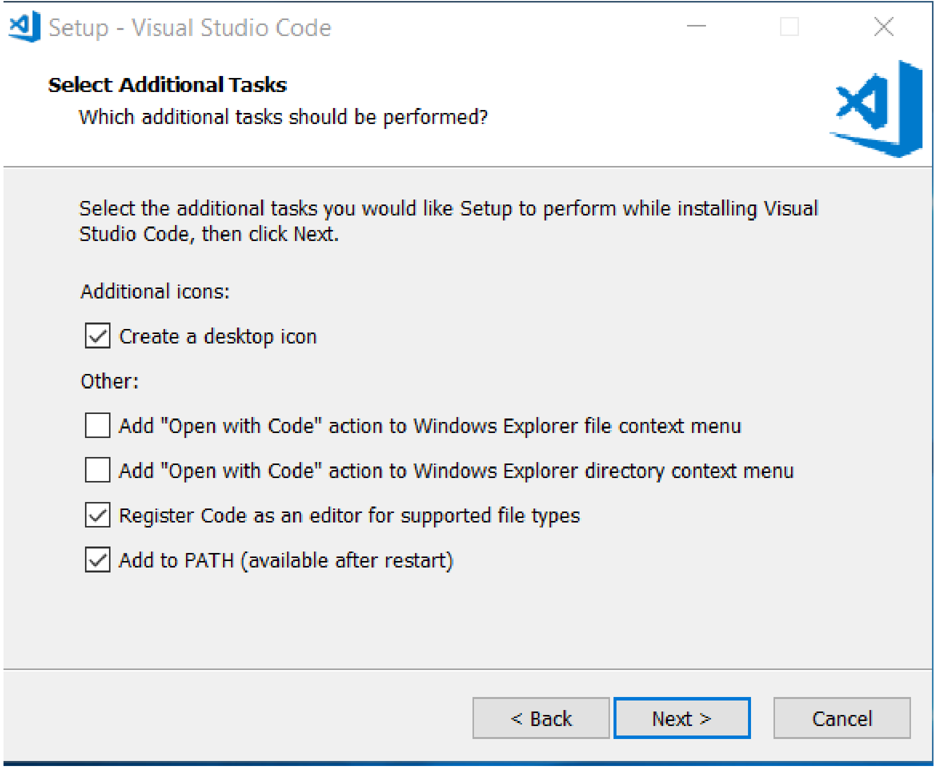
```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import pygame
>>> 
```

Installation of Visual Studio Code – Windows
(a development environment)

Visual Studio Code (a.k.a.VS Code) is a lightweight developer tool (or code editor).

Make sure you have already installed Python

1. Download and run Visual Code installer from http://code.visualstudio.com/Download.
2. In the following window, check "**Add to PATH**" and "**Register Code as an editor for supported file types**". You can decide on the rest of the options according to your preferences. If Visual Studio Code is added to the PATH, you can type "code + <folder_name>" in the **Command Prompt** to launch Visual Studio Code.



Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☑ Create a desktop icon

Other:

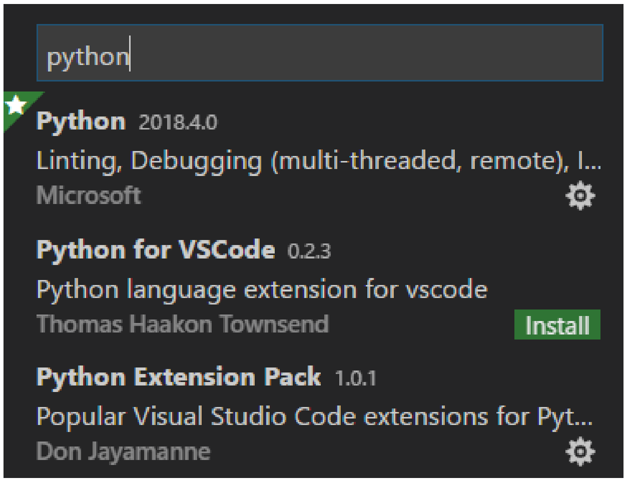
☐ Add "Open with Code" action to Windows Explorer file context menu

☐ Add "Open with Code" action to Windows Explorer directory context menu

☑ Register Code as an editor for supported file types

☑ Add to PATH (available after restart)

3. To maximize your Python productivity you should install Don Jayamanne's Python extension. You install the extension by launching VS Code, opening up the **Command Palette** (Control + Shift + P) and entering `ext install` in it, then pressing ENTER. Type `python` to see a list of Python extensions. Be sure to choose the correct one. This will give you access to features like Python-aware Intellisense,



python

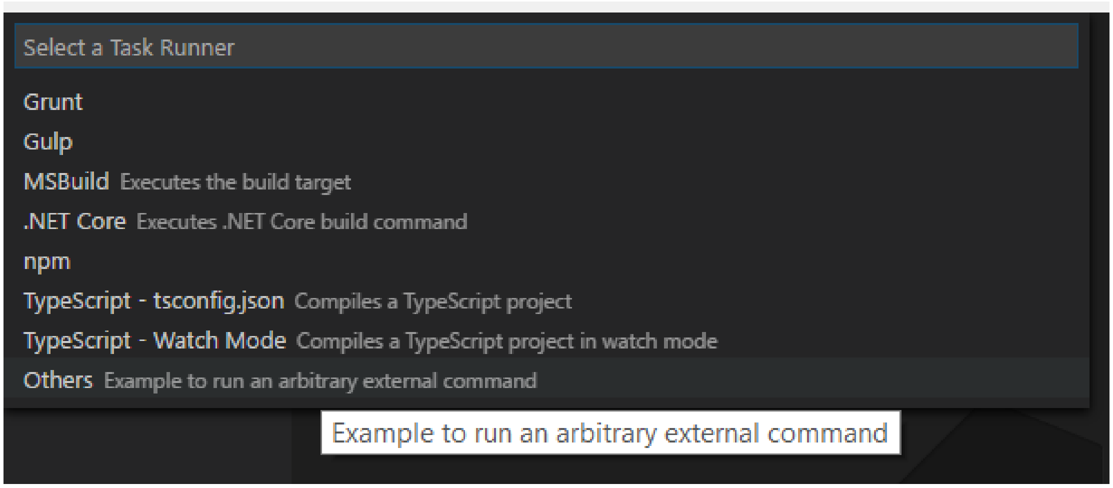
Python 2018.4.0
Linting, Debugging (multi-threaded, remote), I...
Microsoft

Python for VSCode 0.2.3
Python language extension for vscode
Thomas Haakon Townsend

Python Extension Pack 1.0.1
Popular Visual Studio Code extensions for Pyt...
Don Jayamanne

auto-completion, hover tooltips to view function and method signatures, error checking, code formatting, snippets, and more.

4. Now, let’s configure your workspace (namely, some folder in which you put all your code).
   a. Click File->Open Folder. Then create and select a folder you would like to use for all your code. This folder is now called the **workspace**.
   b. You can configure a build task in each workspace you have. A build task is a collection of instructions for your PC to run a program that you open in VS Code from your workspace.



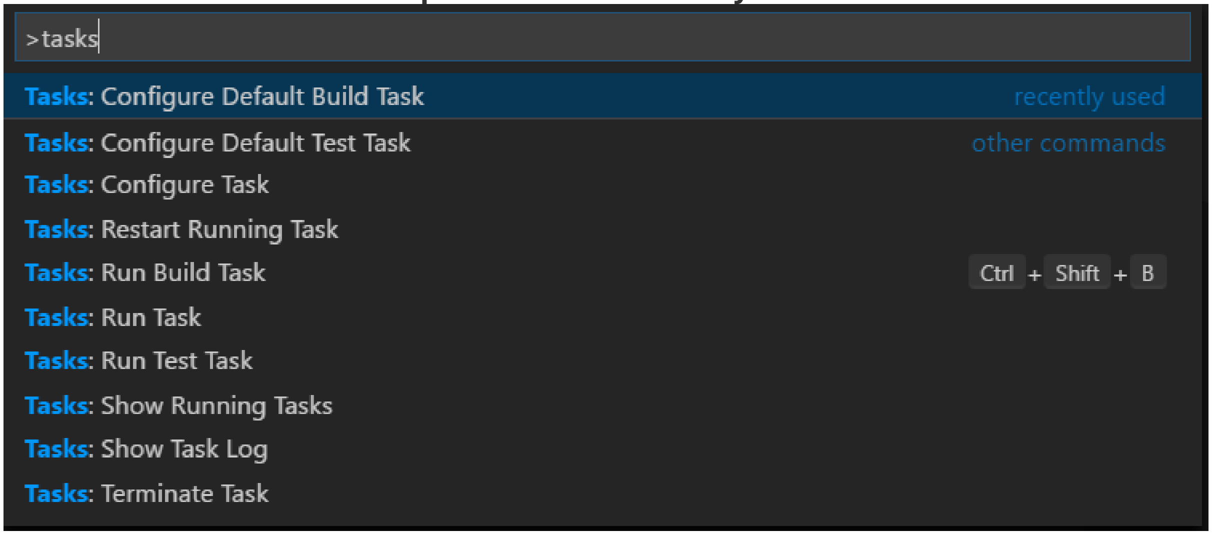
Select a Task Runner

Grunt
Gulp
MSBuild Executes the build target
.NET Core Executes .NET Core build command
npm
TypeScript - tsconfig.json Compiles a TypeScript project
TypeScript - Watch Mode Compiles a TypeScript project in watch mode
Others Example to run an arbitrary external command

Example to run an arbitrary external command

5. To configure the build task in the current workspace, press Control + Shift + P to reach the Command Palette, then type “task” and select the “Tasks: Configure Default Build Task” option. The first time you configure your workspace, the following

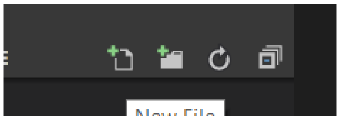
menu will show up. VS Code supports many build tools, but we aren’t going to use them. Choose “**Others Example to run an arbitrary external command**”.

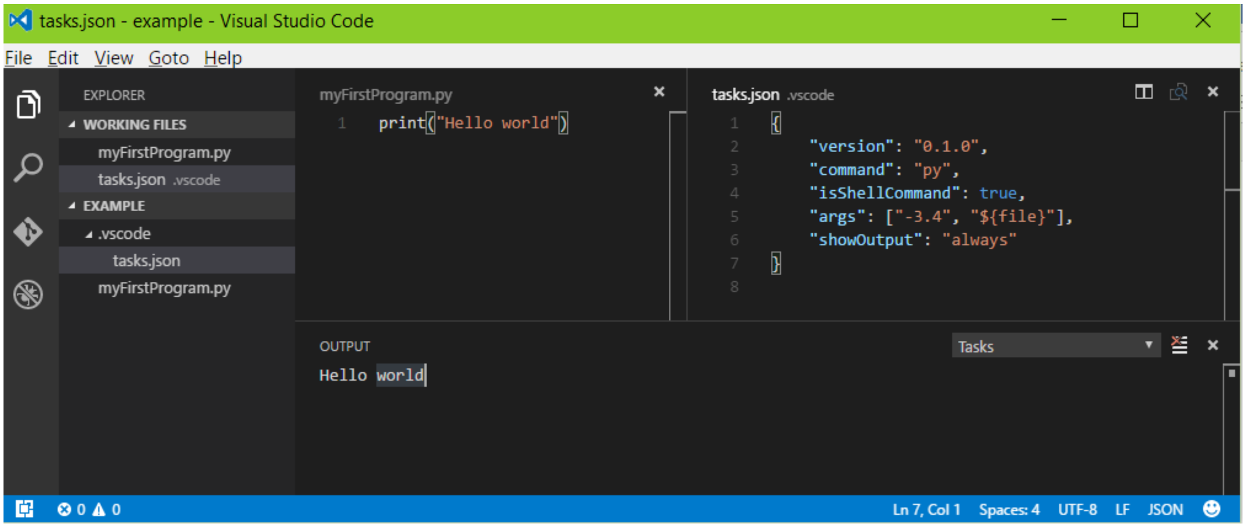


6. Now, let’s replace the contents of **tasks.json** with the snippet below and save the modified file. To run the Python program you are editing, you press **Control + Shift + B**. Once you have this, you probably won’t need to modify “tasks.json” again because the task will work for the whole workspace (anywhere within the folder)

```
{
    "version": "0.1.0",
    "command": "py",
    "isShellCommand": true,
    "args": ["-3.4", "${file}"],
    "showOutput": "always"
}
```

7. Let’s go ahead and write a simple “Hello World” program in your workspace, and use the shortcut to run your program. To create your “Hello World” program, click the New File button that shows up when you hover over the workspace title. If you see the correct output, you have successfully configured your working environment. Enjoy your programming journey!!





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

```
{
    "version": "0.1.0",
    "command": "py",
    "isShellCommand": true,
    "args": ["-3.4", "${file}"],
    "showOutput": "always"
}
```

```
Hello world
```