



FREE eBook

LEARNING blender

Free unaffiliated eBook created from
Stack Overflow contributors.

#blender

Table of Contents

About	1
Chapter 1: Getting started with blender	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Hello World! (Add-On).....	2
The viewport and its windows.....	4
Chapter 2: Getting Started with Programming in Blender	6
Introduction.....	6
Examples.....	6
Hello World! (Add-On).....	6
Credits	8

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [blender](#)

It is an unofficial and free blender ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official blender.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with blender

Remarks

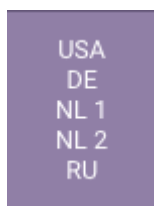
Blender is a free, open-source, 3-dimensional modeling, game building, and rendering software. Blender is written in C and C++. In addition, Blender can be extended with Python scripting. All of the source code of every single previous version since 2003 can be found here:

<http://download.blender.org/source/>

Examples

Installation or Setup

- Go to <https://www.blender.org/download/>
- Choose your operating system
- Click the proper mirror next to the version of blender for your operating system. You can usually just click the mirror closest to your current location. ([more info](#))



- Also, at the bottom of the page are also links to the daily experimental builds and the source code. This can allow you to get access to the latest features (at the cost of stability).
- You have successfully downloaded blender!

Once you have downloaded blender, to open it, simply unzip, and then open blender.exe or blender.app

Hello World! (Add-On)

```
# not all of this is required, but just here for reference
bl_info = {
    "name": "Hello World",          # name of the add-on
    "author": "Blender developer", # name of the author
    "version": (1, 0),             # version number for the add-on
    "blender": (2, 78, 0),         # version of Blender the add-on is
compatible with
    "location": "Space > Hello World", # where the user can find the add-on
    "description": "Greets something", # add-on description
    "warning": "Beta version",      # whatever the user needs to be warned
about
    "wiki_url": "",                # documentation link
    "category": "Development",     # add-on category
```

```

}

# the blender python module
import bpy
# this is just for convenience - could just use as bpy.props.StringProperty, but there are
normally lots of properties
from bpy.props import StringProperty

class HelloWorld(bpy.types.Operator):
    """Says hello to the world.""" # python docstring
    bl_idname = "greetings.hello_world" # this will be callable with
bpy.ops.greetings.hello_world()
    bl_label = "Hello World" # the user-friendly name for this operator (e.g., in
a button)
    bl_options = {'REGISTER', 'UNDO'} # 'UNDO' is only here for reference (you can't
actually take back what you say)

    name = StringProperty(
        name="name",
        default="world",
        description="Who to say hello to",
    )

    def execute(self, context):
        # Note: The execute method is called when the user clicks on the operator or calls it
from python.

        message = "Hello, " + self.name + "!"

        # print to console
        print(message)

        # show a popup that automatically goes away (in info area's header)
        self.report({'INFO'}, message)

        # show a popup under the cursor that doesn't go away automatically
        self.report({'ERROR'}, message)

        # generally return {'FINISHED'} or {'CANCELLED'} at the end of the execute method
        return {'FINISHED'}

# this is automatically called when the add-on is enabled
def register():
    # simply register the class
    bpy.utils.register_class(HelloWorld)

# this is automatically called when the add-on is disabled
def unregister():
    # simply unregister the class
    bpy.utils.unregister_class(HelloWorld)

# common "pythonic" approach to main()...call register() automatically
if __name__ == "__main__":
    register()

```

Save this in an python file (.py), then install as a regular add-on in Blender. Type "space" in just

about any area in Blender and type "Hello World" to find the operator you built.

The viewport and its windows

Blender's **viewport** is a dynamic, changeable interface composed of many different **windows**. With the program running by default, the **viewport** is composed of 5 different windows. Windows can be identified by looking for their small square indicator icons either in the top or bottom-left corner. They may look like these:



(the 3D view icon)



(the timeline icon)

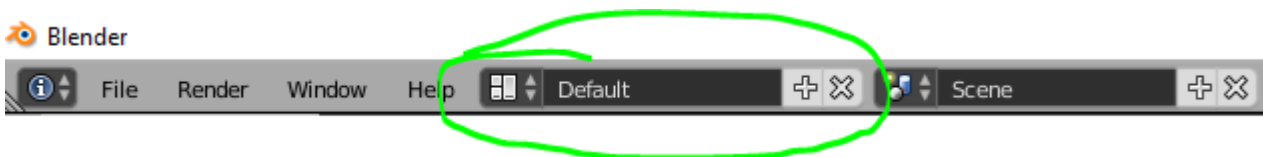


(the properties icon)

These small images denote the type of window they're attached to, and the window type can be changed by clicking on one of them and choosing another window.

All of the windows are **resizable** and **splittable**, meaning that they can each be split into two pieces, changed in size, or be combined together into one window.

To try this functionality, first take note of the location of the **screen layout selector** which appears at the very top of the screen just after the menu buttons:

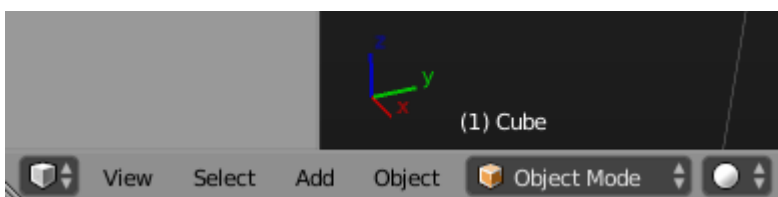


This selector will let you get back to the default window layout at any time, and acts (like many of Blender's selectors) as a dynamic list. This means that if you'd like to save this layout before you start experimenting, click the **+** button to copy the layout, and then make your changes. Return to the layout by clicking the layout name and selecting the default again.

Now that the layout can be returned to, drag one of the **grab handles** at the corner of the main window - it looks like this:



Your cursor will transform into a crosshair and the window will split in half.



Combining two windows together can be done with the **grab handle** from the opposite corner of

the window. In the image above, the grab handle used to split the window was in the *bottom-left* corner: thus the **grab handle** used to combine the window with another is the one in the *top-right*.

To combine the window with another, just drag this second **grab handle** in the direction you want to combine. You may have to pull it away from itself first, if you want to collapse it inwards, like so:



Read [Getting started with blender online](https://riptutorial.com/blender/topic/9241/getting-started-with-blender): <https://riptutorial.com/blender/topic/9241/getting-started-with-blender>

Chapter 2: Getting Started with Programming in Blender

Introduction

Whilst most of the Blender source code is written in C and C++, Extensions (Add-ons) are coded entirely in Python.

Blender comes with >90 extensions installed, but they are not all activated by default.

Blender extensions are installed and activated through the User Preferences window (accessible through the `File` menu or with the shortcut `Ctrl + Alt + u`).

Examples

Hello World! (Add-On)

```
# not all of this is required, but just here for reference
bl_info = {
    "name": "Hello World",           # name of the add-on
    "author": "Blender developer",  # name of the author
    "version": (1, 0),              # version number for the add-on
    "blender": (2, 78, 0),          # version of Blender the add-on is
    compatible with
    "location": "Space > Hello World", # where the user can find the add-on
    "description": "Greets something", # add-on description
    "warning": "Beta version",       # whatever the user needs to be warned
    about
    "wiki_url": "",                  # documentation link
    "category": "Development",      # add-on category
}

# the blender python module
import bpy
# this is just for convenience - could just use as bpy.props.StringProperty, but there are
# normally lots of properties
from bpy.props import StringProperty

class HelloWorld(bpy.types.Operator):
    """Says hello to the world.""" # python docstring
    bl_idname = "greetings.hello_world" # this will be callable with
    bpy.ops.greetings.hello_world()
    bl_label = "Hello World" # the user-friendly name for this operator (e.g., in
    a button)
    bl_options = {'REGISTER', 'UNDO'} # 'UNDO' is only here for reference (you can't
    actually take back what you say)

    name = StringProperty(
        name="name",
        default="world",
        description="Who to say hello to",
```



```

    )

    def execute(self, context):
        # Note: The execute method is called when the user clicks on the operator or calls it
        from python.

        message = "Hello, " + self.name + "!"

        # print to console
        print(message)

        # show a popup that automatically goes away (in info area's header)
        self.report({'INFO'}, message)

        # show a popup under the cursor that doesn't go away automatically
        self.report({'ERROR'}, message)

        # generally return {'FINISHED'} or {'CANCELLED'} at the end of the execute method
        return {'FINISHED'}

# this is automatically called when the add-on is enabled
def register():
    # simply register the class
    bpy.utils.register_class>HelloWorld)

# this is automatically called when the add-on is disabled
def unregister():
    # simply unregister the class
    bpy.utils.unregister_class>HelloWorld)

# common "pythonic" approach to main()...call register() automatically
if __name__ == "__main__":
    register()

```

Save this in an python file (.py), then install as a regular add-on in Blender. Type "space" in just about any area in Blender and type "Hello World" to find the operator you built.

Read [Getting Started with Programming in Blender online:](https://riptutorial.com/blender/topic/10157/getting-started-with-programming-in-blender)

<https://riptutorial.com/blender/topic/10157/getting-started-with-programming-in-blender>

Credits

S. No	Chapters	Contributors
1	Getting started with blender	10 Replies , Community , eunoia , GiantCowFilms , JakeD , Scott Milner
2	Getting Started with Programming in Blender	eunoia