Skip to content

bpy.utils submodule (bpy.utils.previews)

This module contains utility functions to handle custom previews.

It behaves as a high-level ‘cached’ previews manager.

This allows scripts to generate their own previews, and use them as icons in UI widgets (‘icon_value’ for UILayout functions).

Custom Icon Example

```
# This sample script demonstrates how to place a custom icon on a button or
# menu entry.
#
# IMPORTANT NOTE: if you run this sample, there will be no icon in the button
# You need to replace the image path with a real existing one.
# For distributable scripts, it is recommended to place the icons inside the
# addon folder and access it relative to the py script file for portability
#
#
# Other use cases for UI-previews:
# - provide a fixed list of previews to select from
# - provide a dynamic list of preview (eg. calculated from reading a directory)
#
# For the above use cases, see the template `ui_previews_dynamic_enum.py`.


import os
import bpy


class PreviewsExamplePanel(bpy.types.Panel):
    """Creates a Panel in the Object properties window"""
    bl_label = "Previews Example Panel"
    bl_idname = "OBJECT_PT_previews"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
    bl_context = "object"

    def draw(self, context):
        layout = self.layout
        pcoll = preview_collections["main"]

        row = layout.row()
        my_icon = pcoll["my_icon"]
        row.operator("render.render", icon_value=my_icon.icon_id)

        # my_icon.icon_id can be used in any UI function that accepts
        # icon_value # try also setting text=""
        # to get an icon only operator button


# We can store multiple preview collections here,
# however in this example we only store "main"
preview_collections = {}
```

```
def register():

    # Note that preview collections returned by bpy.utils.previews
    # are regular py objects - you can use them to store custom data.
    import bpy.utils.previews
    pcoll = bpy.utils.previews.new()

    # path to the folder where the icon is
    # the path is calculated relative to this py file inside the addon folder
    my_icons_dir = os.path.join(os.path.dirname(__file__), "icons")

    # load a preview thumbnail of a file and store in the previews collection
    pcoll.load("my_icon", os.path.join(my_icons_dir, "icon-image.png"), 'IMAGE')

    preview_collections["main"] = pcoll

    bpy.utils.register_class(PreviewsExamplePanel)


def unregister():

    for pcoll in preview_collections.values():
        bpy.utils.previews.remove(pcoll)
    preview_collections.clear()

    bpy.utils.unregister_class(PreviewsExamplePanel)


if __name__ == "__main__":
    register()
```

bpy.utils.previews.**new()**

RETURNS:

a new preview collection.

RETURN TYPE:

`ImagePreviewCollection`

bpy.utils.previews.**remove(pcoll)**

Remove the specified previews collection.

PARAMETERS:

pcoll (`ImagePreviewCollection`) – Preview collection to close.

class bpy.utils.previews.**ImagePreviewCollection**

Dictionary-like class of previews.

This is a subclass of Python’s built-in dict type, used to store multiple image previews.

Note

- instance with `bpy.utils.previews.new`
- keys must be `str` type.
- values will be `bpy.types.ImagePreview`

clear()

Clear all previews.

close()

Close the collection and clear all previews.

load(name, filepath, filetype, force_reload=False)

Generate a new preview from given file path.

PARAMETERS:

- **name** (*str*) – The name (unique id) identifying the preview.
- **filepath** (*str* | *bytes*) – The file path to generate the preview from.
- **filetype** (*str*) – The type of file, needed to generate the preview in [‘IMAGE’, ‘MOVIE’, ‘BLEND’, ‘FONT’].
- **force_reload** (*bool*) – If True, force running thumbnail manager even if preview already exists in cache.

RETURNS:

The Preview matching given name, or a new empty one.

RETURN TYPE:

`bpy.types.ImagePreview`

RAISES:

KeyError – if `name` already exists.

new(name)

Generate a new empty preview.

PARAMETERS:

name (*str*) – The name (unique id) identifying the preview.

RETURNS:

The Preview matching given name, or a new empty one.

RETURN TYPE:

`bpy.types.ImagePreview`

RAISES:

KeyError – if `name` already exists.